



# **SZARP HOWTO**

## **SZARP + Debian/Red Hat**

**Michał Blajerski**

**Krzysztof Gałązka**

**Dariusz Marcinkiewicz**

**Maciej Mochol**

**Jarosław Nowisz**

**Paweł Pałucha**

**Lucjan Przykorski**

**Michał Rój**

**Stanisław Sawa**

Dokument znany pod nazwą „SZARP Howto” powstał jako zbiór informacji dotyczących Systemu Zbierania, Archiwizacji i Prezentacji danych SZARP, a w szczególności zagadnień związanych z instalacją, konfiguracją i używaniem systemu operacyjnego RedHat Linux, pod kontrolą którego działał SZARP. Obecnie dokumentacja techniczna systemu SZARP została przeniesiona do oddzielnego dokumentu, natomiast jeżeli chodzi o Linuksa nacisk został przeniesiony na dystrybucję Debian i to do niej odnosi się większość aktualnych informacji, choć dokument może być także przydatny dla użytkowników innych dystrybucji.

HOWTO przeznaczone jest dla ludzi z ogólną wiedzą dotyczącą komputerów i podstawowych komend Linuksa/Uniksa (w szczególności nie jest wymagana biegła znajomość Linuksa).

Właścicielem praw autorskich do dokumentu jest firma Praterm S.A. (<http://www.praterm.pl>) z siedzibą w Warszawie. Ten utwór objęty jest licencją *Creative Commons Uznanie autorstwa-Na*

*tych samych warunkach.* Aby zobaczyć kopię niniejszej licencji przejdź na stronę <http://creativecommons.org/licenses/by-sa/2.0/pl/> lub napisz do Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA. Na podstawie niniejszej licencji dozwolone jest kopiowanie, rozpowszechnienie, modyfikowanie i komercyjne wykorzystanie tego dokumentu, pod warunkiem zamieszczenia informacji o autorach/właścicielu praw autorskich oraz udostępniania tego dokumentu (w tym jego zmodyfikowanych wersji) na warunkach tej samej licencji.

Firma Praterm S.A. oraz autorzy nie ponoszą żadnej odpowiedzialności za wykorzystanie informacji zawartych w tym dokumencie.

\$Id\$

# 1. Wstęp

## 1.1. Historia systemu SZARP

Wszystko zaczęło się w roku 1991. Inicjatorem był Witold Kowalewski, właściciel firmy Praterm. Praterm od momentu powstania w 1983 roku zajmował się automatyzacją ciepłownictwa. Początkowo wykorzystywał urządzenia elektro-mechaniczne, od 1989 roku zaczął wykorzystywać mikroprocesorowe sterowniki przemysłowe firmy Z-Elektronik należącej do Krystiana Zasadzińskiego. Wkrótce okazało się, że dla przeprowadzenia analiz prowadzenia przez automatykę procesów technologicznych w ciepłownictwie potrzebny jest system do rejestrowania danych charakteryzujących te procesy: temperatur, ciśnień itp. Zbiegło się to z początkiem ery komputerów PC. W 1991 roku Piotr Branecki stworzył pierwszy program w Turbo Pascalu pracujący pod DOSem, który zbierał dane ze sterowników podłączonych za pomocą terminala wieloportowego firmy Specialix przez RS-y 232. Dane zapisane były w formacie binarnym specyficznym dla tego programu. W roku 1992 do prac nad rozwojem SZARPa dołączył Lucjan Przykorski. Oprogramowanie zostało podzielone na dwie części:

- Serwer zbierający dane i prezentujący je w formie raportów oraz wykresów z ostatniej doby.
- Terminal z programem przeglądającym do prezentacji w formie wykresów danych zarejestrowanych od początku pracy systemu.

Wszystko pracowało w jednowątkowym DOSie, stąd potrzeba dwóch komputerów, aby można było uruchomić w systemie więcej niż jeden program. Sieć pracowała na Novell NWLite 1.0, potem 1.1. Baza zapisywana była w systemie opartym na plikach dBaseIV: DBF + MDX, który to format zachował się aż do SZARP v2.1. W roku 1993 do zespołu programistów dołączył na chwilę Stanisław Kłosiński, autor innego programu klienckiego dedykowanego do pracy na terminalu, służącego do specjalistycznej analizy zarejestrowanych danych.

W tym samym roku 1993 Branecki i Przykorski przenieśli SZARP na platformę Unix - był to USL Unix SVR4.2 firmy Consensus, którego ilość egzemplarzy sprzedana w Polsce dała się policzyć na palcach. SZARP jako system pracujący w środowisku wielozadaniowym został podzielony na szereg programów. Część z nich działała jako demony pracujące w tle, interfejs użytkownika stanowiły zaś aplikacje

klienckie wykorzystujące Motif 1.1.1. W roku 1994 do zespołu programistów dołączył Marcin Anderson, twórca kilku programów zarówno w kategorii demonów jak i aplikacji użytkowych.

We wrześniu 1994 Branecki i Przykorski obronili na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej pracę dyplomową: "System zbierania danych i sterowania wolnozmiennych procesów technologicznych o rozproszonych punktach kontrolno-pomiarowych". Praca to dotyczyła SZARPa i współpracujących z nim sterowników przemysłowych. Pomimo innowacyjnego charakteru pracy, kilku działających w rzeczywistości instalacji i ewenementu w postaci filmu, który dyplomanci nakręcili w ramach pracy, spotkała się z pełną ignorancją ze strony uczelni.

Jeszcze w roku 1994 SZARP doczekał się nowego systemu operacyjnego: Novell UnixWare 1.1, który został przez Novella zakupiony od AT&T. Następną platformą był Novell UnixWare 2.03, po której to wersji Novell sprzedał ze stratą UnixWare do SCO. Ta ostatnia wypuściła w 1995 roku SCO UnixWare 2.1, który nie licząc kilku patchy był ostatnim z komercyjnych Unixów, na którym pracował SZARP.

W latach 1993-1995 jako terminale wykorzystywane były komputery pracujące w DOSie z programem XAppeal firmy Xtreme charakteryzujący się bardzo małymi wymaganiami sprzętowymi i jeszcze mniejszą listą obsługiwanego sprzętu nie różniąc się specjalnie w tym ostatnim od linii UnixWare. Od roku 1996 w swoich instalacjach Praterm zaczął wykorzystywać emulatory Xserwera pracujące pod MS-Windows: Starnet X-Win, SCO XVision Eclipse 7.1, Labf.com X-Win Pro / WinaXe. Najwięcej terminali zostało wykorzystanych w oparciu o ten ostatni, najtańszy, najlżejszy i najbardziej rozwijający się produkt fińskich programistów.

W 1996 roku pozostający do tego momentu bezimiennym system zbierania, archiwizacji i prezentacji danych doczekał się oficjalnej nazwy SZARP v2.0. Nazwa SZARP v1.0 miała dotyczyć instalacji pod DOSem. SZARP v2.0 składał się z 12 modułów, jego struktura pozwalała na dowolną rozbudowę o kolejne. W lecie roku 1996 od Pratermu odszedł Piotr Branecki, na początku roku 1997 firmę opuścił także Marcin Anderson. W roku 1997 w Legionowie dokonana została ostatnia instalacja SZARPa na SCO UnixWare. Pod koniec roku 1997 do zespołu programistów dołączyli Maciej Mochol i Jarosław Nowisz. To oni dokonali portu SZARP z UnixWare do Linuksa tworząc w ten sposób SZARP v2.1. Oprócz samego przeniesienia wykonali szereg usprawnień i usunęli niezliczoną masę błędów. W roku 1998 SZARP doczekał się repozytorium CVS, na którym rezyduje do dnia dzisiejszego. W tym samym roku Maciej Mochol zaadoptował SZARPa do autoconfa i stworzył z niego dystrybucję (wcześniej była to luźno rozrzucona grupa programów). Od roku 1998 SZARP był instalowany już tylko na Linuxie, początkowo na dystrybucji Red Hat poczynając od wersji 5.2.

W roku 1999 do zespołu programistów dołączył Michał Rój. Na początku roku 2000 Maciej Mochol stworzył narzędzie parconf, które później wielokrotnie udoskonalane znakomicie usprawniło zmiany w konfiguracji poszczególnych serwerów, które to zmiany wcześniej wymagały dużej ilości ręcznych operacji. W połowie 2000 roku zespół rozszerzył się o Pawła Pałuchę. Pod koniec roku 2000 zrezygnował z pracy nad projektem Jarosław Nowisz. W połowie kwietnia 2001 uruchomiona została pierwsza instalacja BODASa - Bazy Ogólnopolskiej Danych Archiwalnych SZARP. Część systemów z SZARPem podłączonych do internetu przez stałe łącze zrzuca aktualne dane co 10 minut, nieliczne już hosty z modemami łączą się raz na dobę zgrywając dane na serwer praterm.com.pl. Do centralnego serwera Pratermu podłączane są sukcesywnie kolejne hosty z SZARPem. W kwietniu 2001 do zespołu developerów SZARPa dołączył Stanisław Sawa. Pod koniec roku 2001 Paweł Pałucha dodał do SZARPa pierwszą od dłuższego czasu aplikację z graficznym interfejsem użytkownika - Dyspozytor.

Na początku 2002 zrezygnował z pracy w zespole Maciej Mochol. We wrześniu 2002 roku Paweł Pałucha obronił na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego kolejną pracę magisterską dotyczącą SZARPa - *Zdalne sterowanie i kontrola procesu przemysłowego z wykorzystaniem technologii XML/SVG*, tworząc tym samym nową wersję wizualizacyjnej części systemu. Pod koniec roku 2002 ten sam developer rozpoczął przenoszenie części aplikacji użytkowych SZARPa na platformę MS-Windows, z wykorzystaniem biblioteki wxWindows/wxWidgets.

W roku 2003 SZARP był instalowany na pochodnych dystrybucji Red Hat - Aurox 9.x. W tym samym roku, w związku z wycofaniem się Red Hat z rozwijania dystrybucji niekomercyjnych, SZARP zaczął być instalowany na Debianie, którego specjalizowaną dystrybucję przygotował Stanisław Sawa. W tym samym roku terminale windowsowe zaczęły być wyposażane w Xserwer Cygwin - w odróżnieniu od WinaXe darmowy i wyposażony w wiele funkcji jak np. serwer sshd. Także w 2003 roku Paweł Pałucha rozpoczął prace nad przeniesieniem konfiguracji do IPK - Idealnego Pliku Konfiguracyjnego - w formacie XML.

W roku 2004 do zespołu "sterownikowego" firmy Praterm S.A. dołączył Paweł Kolega, także uczestniczący w rozwoju oprogramowania. Paweł Pałucha zaprojektował i zaimplementował nowy, prostszy format bazy danych, co pozwoliło na stopniowe wycofywanie się z ostatniego komercyjnego elementu systemu - biblioteki Codebase do obsługi bazy dBase IV. Jako emulator X-terminala dla terminali z Windows, ze względu na prostotę instalacji, zaczął być stosowany freeware'owy Pexus X/Deep32.

06.12.2004 akcje Praterm SA zadebiutowały na GPW, co w praktyce oznaczało szereg poważnych zmian dla SZARPa. Skończył się etap pilnowania każdej złotówki przy pracach rozwojowych (rozpatrywanie każdego tematu pod kątem zwrotu w jak najkrótszym czasie), zamiast tego należało położyć nacisk na bezpieczeństwo. 01.01.2005 Lucjan Przykorski oficjalnie przestał być aktywnym członkiem zespołu programistów (co stało się faktem jakiś czas wcześniej), zajmując się w Pratermie sprawami technicznymi, informatyka zaś pozostała jego hobby. Szefem działu IT został Paweł Pałucha, który od tego momentu decydował o przyjmowaniu nowych informatyków i szczegółowych pracach nad SZARPem.

Na przełomie roku 2004/2005 współpracę przy tworzeniu SZARP'a zakończył Michał Rój, natomiast dołączyli trzej nowi programiści - Michał Blajerski, Dariusz Marcinkiewicz i Krzysztof Gałązka. Do działu sterownikowego dołączył Jarek Janik. Dokończono przenoszenie całości konfiguracji do formatu XML. Powstało wiele sterowników do nowych urządzeń, oprogramowanie do importu danych z innych systemów telemetrycznych, zrealizowano instalacje wykorzystujące łączność bezprzewodową WLAN i komunikację GSM/GPRS.

Pod koniec III-go kwartału 2005 z pracy w zespole programistów zrezygnował Stanisław Sawa. Na wszystkich ciepłowniach Pratermu dokonano aktualizacji systemów operacyjnych do Debiana, przeprowadzono konwersję istniejących baz na nowy format. Działania te udało się także podjąć u większości klientów zewnętrznych - mimo kilku nowo zakupionych przedsiębiorstw nadal około połowy instalacji to spółki spoza grupy Praterm.

Rok 2006 to okres dość intensywnego rozwoju aplikacji klienckich, w szczególności pierwszym użytkownikom zaoferowano wersje programu przeglądającego nowej generacji, pracującego także pod kontrolą Windows. W połowie roku do zespołu programistów dołączył Krzysztof Ołowski, choć w dużej mierze z myślą o pracy przy oprogramowaniu do sprawozdawczości finansowej w holdingu.

## 1.2. Wdrożone instalacje

Poniższe zestawienie wymienia w kolejności chronologicznej (pierwsza instalacja, nie upgrade) wszystkie przedsiębiorstwa ciepłownicze z SZARP v2.1 - 3.1. Nie zawiera ono instalacji na węzłach ciepłych.

**Tabela 1. Wdrożenia**

Ciepłownia	Rok instalacji	Konfiguracja
------------	----------------	--------------

<b>Ciepłownia</b>	<b>Rok instalacji</b>	<b>Konfiguracja</b>
Pyskowice	1992	1 serwer + 1 terminal
Suwałki Ciepłownia Główna	1992	1 serwer + 6 terminali
Gliwice	1993	3 serwery + 8 terminali
Skierniewice	1994	1 serwer + 3 terminale
Zagórz	1997	1 serwer
Legionowo Ciepłownia Łajski	1997	1 serwer + 6 terminali
Sztum	1998	1 serwer + 1 terminal
Starachowice Północ	1998	1 serwer + 1 terminal
Żory	1998	1 serwer + 4 terminale
Rawicz	1998	1 serwer + 1 terminal
Ełk	1998	1 serwer + 1 terminal
Gizycko	1998	1 serwer + 3 terminale
Racibórz	1999	1 serwer + 1 terminal
Włocławek	1999	1 serwer + 1 terminal
Poręba	2000	1 serwer + 2 terminale
Pasłęk	2000	1 serwer + 1 terminal
Starachowice Południe	2000	1 serwer + 1 terminal
Legionowo Budynek administracji	2000	1 serwer + 1 terminal
Świdnik	2000	1 serwer + 3 terminale
Radomsko	2000	1 serwer + 3 terminale
Suwałki Ciepłownia Centrum	2000	1 serwer + 2 terminale
Bytów	2001	1 serwer + 2 terminale
Katowice	2001	1 serwer + 4 terminale
Przasnysz	2002	1 serwer + 1 terminal
Świecie	2002	1 serwer + 1 terminal
Zamość	2003	1 serwer + 2 terminale
Międzyrzec Podlaski	2003	1 serwer + 1 terminal
Kraśnik	2003	1 serwer + 4 terminale
Tarnowskie Góry	2004	1 serwer + 1 terminal
Kępno	2004	1 serwer + 1 terminal
Chrzanów	2004	1 serwer + 1 terminal
Wodzisław Śląski	2004	1 serwer + 2 terminale
Lidzbark Warmiński	2005	1 serwer + 1 terminal
Dobre Miasto	2005	1 serwer + 1 terminal
Libiąż	2005	1 serwer + 1 terminal
Atex Zamość	2005	1 serwer + 2 terminale
Modlin	2005	1 serwer + 1 terminal
Gniew	2006	1 serwer + 1 terminal
Radomsko	2007	1 serwer + 1 terminal

Ciepłownia	Rok instalacji	Konfiguracja
Komech Sosnowiec	2007	1 serwer + 1 terminal
Jarocin ciepłownia Paged	2008	1 serwer + 1 terminal
Frombork	2008	1 serwer
PEC Jarocin	2008	1 serwer + 1 terminal
Dalkia Poznań	2008	1 serwer
KPEC Kraśnik	2008	1 serwer + 1 terminal
EC Szombierki ZEC Bytom	2009	1 serwer + 1 terminal
RCEkoenergia Czechowice	2009	1 serwer
Ruda Śląska (ZEC Katowice wydział XII)	2009	1 serwer + 2 terminale
"Mysłowice" - Mysłowice (ZEC Katowice wydział III)	2009	1 serwer + 2 terminale
"Wujek" Katowice (ZEC Katowice wydział VI)	2009	1 serwer + 2 terminale
ZEC Biłgoraj	2009	1 serwer
"Wesoła" Mysłowice (ZEC Katowice wydział IX)	2009	1 serwer + 2 terminale
Ciepłownia Miejska Małogoszcz	2009	1 serwer
EC Szombierki Bytom	2009	1 serwer + 1 terminal
"Kazimierz-Juliusz" Sosnowiec (ZEC Katowice wydział I)	2010	1 serwer + 2 terminale
"Murcki" Katowice (ZEC Katowice wydział IV)	2010	1 serwer + 2 terminale
"Kleofas" Katowice (ZEC Katowice wydział VIII)	2010	1 serwer + 2 terminale
Dalkia Szczytno	2010	1 serwer + 2 terminale
Ergis Wąbrzeźno	2010	1 serwer + 1 terminal
PEC Jelenia Góra	2010	1 serwer + 1 terminal

## 2. Wymagania sprzętowe

Ogólnie SZARP dobrze się czuje na co najmniej dwóch maszynach: jednej - serwerze najlepiej schowanym w kącie i drugiej terminalu z Linuksem lub Windows z jakimś emulatorem XServera. W przypadku wybitnie ograniczonych środków można oczywiście poprzestać tylko na jednym komputerze. Trudno określić ściśle modele poszczególnych komponentów komputerów pracujących z Linuksem i SZARP'em z uwagi na ciągle zmiany w ofercie producentów. Zbiór wymagań podany jest przy poszczególnych elementach. Dokładniejsze informacje można uzyskać w HOWTO dotyczących instalacji Linuksa (Hardware-HOWTO, Ethernet-HOWTO). Dobrym źródłem informacji są też newsy, np. przeglądane z <http://www.deja.com>. Należy zaznaczyć, że istnienie tego rozdziału, o ile było bardzo potrzebne w początkach rozwoju Linuksa, to w tej chwili traci znaczenie, gdyż praktycznie większość

dostępnego sprzętu w tej chwili bez problemu pracuje z Linuksem.

## 2.1. Serwer

Chodzi 24 godziny na dobę, musi być niezawodny, odporny na przepięcia i zaniki zasilania, z bieżącą archiwizacją danych.

### 2.1.1. Płyta główna

Wymagane są oczywiście sloty PCI do instalacji kart wieloportowych, przyda się także port szeregowy na płycie. Grafika może być wbudowana, podobnie jak karta sieciowa.

BIOS musi mieć następujące opcje:

- podnoszenie systemu samodzielnie w przypadku powrotu zasilania po jego zaniku,
- wyłączenie opcji jakiegokolwiek oszczędzania energii,
- po zaniku napięcia / wyłączeniu w dowolnym momencie (szczególnie w trakcie bootowania przy POST), żeby nie wchodził automatycznie do BIOSa.

W ostatnich latach praktycznie jedyne płyty, z którymi zdarzały się problemy, oparte były na chipsetach SiS (i chodziło głównie o grafikę). Chipsety innych producentów (Intel, AMD, Via, NVidia) nie powinny sprawiać problemów. Należy zwrócić jedynie uwagę, żeby nie kupować super-nowości, np. nie wszystkie kontrolery S-ATA są w tej chwili poprawnie obsługiwane przez Linuksa.

### 2.1.2. Procesor

Żadnych szczególnych wymagań - większość instalacji jest na Intelach, chociaż stosowane też były AMD. SZARP kompiluje się też i działa na platformie AMD Athlon 64, ale obecnie należy to uznać za platformę eksperymentalną. Wydajność procesora powinna być dyktowana wymaganiami używanego środowiska graficznego, sam SZARP zadowolony się naprawdę czymkolwiek.

### 2.1.3. Pamięć

Serwer teoretycznie zadowolony się nawet 32 MB (oczywiście bez X-ów), ale do sprawnej pracy w środowisku graficznym zalecane jest obecnie 512 MB.

### 2.1.4. Pamięć masowa

Do niedawna zalecane były dyski SCSI - mają tę przewagę nad IDE, że kolejukują odwołania do dysku, więc teoretycznie w środowisku wielozadaniowym powinny działać szybciej mniej machając głowicami. Obecnie jednak dyski IDE ATA zapewniają zupełnie wystarczającą prędkość. Serial ATA zapewnia większą wydajność niż ATA, należy tylko zwrócić uwagę, żeby kontroler S-ATA na płycie był poprawnie obsługiwany przez Linuksa.

W przypadku SCSI należy pamiętać o zakończeniu łańcucha urządzeń terminatorem - albo dodatkowym, albo przez przełączenie odpowiedniego jumpera na urządzeniu. Kompatybilność z Linuksem można

sprawdzić na stronie producenta, w większości przypadków nie powinno być problemów (kontrolery firm Adaptec czy Tekram).

Jeżeli chodzi o sprzętowy RAID, to dobre wsparcie dla Linuksa posiadają chipsety firmy HighPoint. Należy zwrócić uwagę, że do instalacji Linuksa na RAID'ie sprzętowym konieczne jest wsparcie jądra dla danego układu już na etapie instalacji (być może konieczne będzie ręczne załadowanie odpowiedniego modułu - instalatory zarówno RedHata jak i Debiana dają taką możliwość).

### 2.1.5. Grafika

Karta graficzna - znów dowolna, byle współpracowała z Linuksem. Karty obu najpopularniejszych producentów - Nvidii i ATI działają z Linuksem (choć nie zawsze dostępna jest np. akceleracja 3D). Nie powinno być także problemów z większością kart zintegrowanych - powinny działać zarówno S3, Intel Extreme Graphics I/II czy VIA.

Listę driverów i obsługiwanych przez nie kart dla XFree86 można obejrzeć na stronie [www.xfree86.org](http://www.xfree86.org) (<http://www.xfree86.org/current/RELNOTES4.html#43>), dla X-serwera X.org odpowiedni link to <http://www.x.org/X11R6.8.2/doc/manindex4.html>.

### 2.1.6. Sieć

Spora część nowych płyt głównych wyposażona jest w zintegrowaną kartę sieciową Ethernet 100Mb/s lub 1Gb/s. Praktycznie wszystkie karty, zarówno te zintegrowane, jak i w postaci kart PCI czy PCMCIA (dla notebooków), są bez problemu obsługiwane przez Linuksa. Najpopularniejsze są karty oparte na układach firmy Realtek, inne spotykane to między innymi Intel, Via. Rozbudowaną listę kart wraz z informacją o współpracy z Linuksem można znaleźć w sieci jako Ethernet-HOWTO (<http://www.tldp.org/HOWTO/Ethernet-HOWTO.html>).

W przypadku RJ45 (praktycznie jedyne obecnie stosowane złącza) można połączyć dwa komputery bez użycia HUB-a. Używa się do tego tzw. skrętki przekrosowanej (zasada działania podobna do NULL-modemu), którą łączy się bezpośrednio dwa gniazda RJ.

Przy większej ilości komputerów trzeba zastosować HUB'a (obecnie praktycznie już niespotykane) lub switcha. Starsze urządzenia posiadały oznaczony (często aktywowany przełącznikiem) port Up-Link, pozwalający na połączenie kaskadowo dwóch lub więcej urządzeń. W przypadku braku takiego portu, do łączenia trzeba było używać skrosowanego kabla. Obecnie większość nowych switchy automatycznie wykrywa sposób podłączenia, więc praktycznie kabli zwykłych i skrosowanych można używać wymiennie.

Switche (w przeciwieństwie do HUB'ów) starają się filtrować ruch, ucząc się topologii sieci. Jeżeli dokonamy zmian w topologii sieci (pozmiemy porty, w które powpinane są konkretne komputery), może być konieczne zmuszenie switcha do zapomnienia dotychczasowej konfiguracji, najprościej przez reset - wystarczy na kilka sekund odłączyć zasilanie switcha.

Żeby sieć działała z prędkością 100Mbps lub więcej, przewody muszą być połączone 1-do-1, ale musi być także przestrzegana norma EIA/TIA 568. Specyfikacja dostępna jest na

<http://www.faqs.org/faqs/LANs/cabling-faq/preamble.html>  
(<http://www.faqs.org/faqs/LANs/cabling-faq/preamble.html>). W skład normy wchodzi następujące rodzaje okablowania:

- *Standard EIA/TIA T568A (zwany także ISDN, uprzednio zwany EIA)*

Pin Wire Color

```

==== =====
      /--T3  1  White/Green
pair3 \--R3  2  Green
    /-----T2  3  White/Orange
  /           /-R1  4  Blue
pair2 \ pair1 \-T1  5  White/Blue
    \-----R2  6  Orange
      /--T4  7  White/Brown
pair4 \--R4  8  Brown

```

- *Standard EIA/TIA T568B (zwany także AT&T, uprzednio zwany 258A; częściej stosowany)*

```

==== =====
      /--T2  1  White/Orange
pair2 \--R2  2  Orange
    /-----T3  3  White/Green
  /           /-R1  4  Blue
pair3 \ pair1 \-T1  5  White/Blue
    \-----R3  6  Green
      /--T4  7  White/Brown
pair4 \--R4  8  Brown

```

- *USOC (Universal Service Order Code - praktycznie nieużywany)*

```

      8-pins    6-pins
      |         |
    /-----T4  1  White/Brown
  /           /-----T3  2  1  White/Green
 /           /           /-----T2  3  2  White/Orange
 /           /           /-R1  4  3  Blue
pr4\ pr3\ pr2\ pr1\ -T1  5  4  White/Blue
  \           \           \-----R2  6  5  Orange
  \           \           \-----R3  7  6  Green
  \-----R4  8  Brown

```

- *Przekrosowany kabel EIA/TIA T568B ma jedną wtyczkę jak powyżej, a drugą:*

```

==== =====
      /--T2  1  White/Green
pair3 \--R2  2  Green
    /-----T3  3  White/Orange
  /           /-R1  4  Blue
pair2 \ pair1 \-T1  5  White/Blue
    \-----R3  6  Orange
      /--T4  7  White/Brown
pair4 \--R4  8  Brown

```

Numeracja pinów na wtyku jest taka: patrząc od dołu (od strony złącz) z kablem do sieci od lewej 1,2,3,4,5,6,7,8.

## 2.1.7. CD-ROM

Dowolny CD-ROM IDE - wykorzystywany głównie przy instalacji systemu.

### 2.1.8. Napęd dyskietek

Praktycznie niewykorzystywany, choć dyskietka może być sposobem na uruchomienie komputera gdy wszystkie inne zawiodą (popsuty CD-ROM, uszkodzony kanał IDE...).

### 2.1.9. Monitor

W przypadku instalacji z dwóch komputerów przy serwerze można postawić byle 15" o rozdzielczości maksymalnej 800x600 - monitor ten włączany jest sporadycznie, tylko przy pracach serwisowych i to też często praca na nim przebiega w trybie tekstowym. Jeżeli instalacja jest "uboga" (tylko serwer), paradoksalnie monitor musi być porządny: minimum 17", zalecany 19" o przyzwoitej częstotliwości odświeżania lub panel LCD. Rozdzielczość najczęściej wykorzystywana to 1024x768, praktyka pokazuje, że większa rozdzielczość znajduje zastosowanie dopiero w monitorach 20", 21", które skądinąd nie są zalecane z uwagi na dużą pracę szyją, którą musi wykonać siedzący przed takim ekranem użytkownik.

Dla monitorów CRT minimalna zalecana częstotliwość odświeżania pionowego to 85 Hz. Aktualną częstotliwość można sprawdzić zwykle gdzieś w menu ekranowym monitora. Parametr ten nie ma znaczenia dla paneli LCD. Dla paneli LCD natomiast istotne jest ustawienie rozdzielczości na odpowiadającą rozdzielczości matrycy (zwykle 1024x768 dla 15", 1280x1024 dla 17").

### 2.1.10. Obudowa, klawiatura, myszka, kable i inne

Dla komputerów instalowanych w ciepłowni konieczne jest zabezpieczenie przed nadmiernym zapyleniem. Na rynku występują obudowy przemysłowe, są one kilkukrotnie droższe od zwykłych, ale wyposażone w dodatkowy wiatrak z wymiennym filtrem i bardzo porządnie wykonane. Zwykle wykorzystywaną pyłoszczelną obudową przemysłową jest Advantech IPC-610MB lub IPC-510MB (tańszy wariant) do zabudowy w szafie 19", jednak doskonale nadający się jako desktop. W przypadku płyt do Pentium IV należy zwrócić uwagę, czy zasilacz wspiera takie płyty (wszystkie nowe zasilacze wspierają). Dla procesorów Pentium IV z rdzeniem Prescott wymagany jest specjalny typ obudowy, z dodatkową rurą odprowadzającą powietrze wprost znad procesora. Zalecana moc zasilacza to nie mniej niż 300W (istotne zwłaszcza przy karcie wieloportowej Moxa).

Dystrybutorem Advantecha jest firma Elmark [www.elmark.com.pl](http://www.elmark.com.pl) (<http://www.elmark.com.pl>).

Jeżeli nie instalujemy obudowy przemysłowej, można zastosować dowolną obudowę dostosowaną do współpracy z wykorzystywaną płytą główną. Zwykle wygodniejsza będzie niezbyt mała obudowa typu "Tower" (stojąca).

Klawiatura PS/2 (lub USB) powinna być niezawodna, klawisze do Windows nie są wymagane (częściej przeszkadzają). Przy obecnych cenach niezawodność nie jest tak istotna - w razie czego zawsze klawiaturę można wyłączyć. Myszka w przypadku serwera "schowanego" może być dowolna. W przypadku "ubogiej" instalacji, gdzie praca odbywa się przy serwerze oraz przy terminalu, konieczne jest zastosowanie myszy optycznej (PS/2 lub USB), która jest dość odporna na zabrudzenia. Jeżeli używamy klawiatury USB, trzeba zwrócić uwagę, żeby klawiatura była wspierana przez BIOS - w przeciwnym razie nie będziemy mogli za jej pomocą wejść do BIOS'u i zmieniać opcji.

### 2.1.11. Moduły wieloportowe

Do podłączenia sterowników konieczna jest zwykle większa ilość portów RS-232. Zwykle 2 standardowe porty w komputerze nie wystarczają, nowe płyty mają zresztą zwykle 1 port lub w ogóle są

ich pozbawione.

W starszych instalacjach wykorzystywane były moduły wieloportowe firmy Specialix. Obecnie w nowych systemach instalowane są moduły Moxya z uwagi na znacznie niższą cenę tudzież dobre wsparcie dla Linuksa.

Specialix <http://specialix.co.uk>, firma wykupiona w 1999 roku przez Perle, produkuje system wieloportowy wykorzystywany w SZARPIe do komunikacji ze sterownikami. System składa się z karty i szeregu modułów 8-mio portowych. Do każdej karty można dołączyć maksymalnie 4 moduły, do komputera można podłączyć do 4 kart, co łącznie daje maksymalnie 128 portów w jednym serwerze. W chwili obecnej Specialix produkuje tylko karty PCI: SX PCI Adapter. Dołączalne moduły to SXDC8/DX (złącza żeńskie DB25), SXDC8/MX (złącza męskie DB25) lub SXDC8/RJX (złącza RJ45). W praktyce używane są głównie SXDC8/DX. Łączenie tych modułów ze sobą odbywa się poprzez specjalne hebelki umieszczone na każdym rogu modułu - w celu dostania się do nich trzeba zdemontować zasłaniające wciskane klapki, co wymaga użycia siły. Ilość wykorzystywanych modułów Specialixa zależy od ilości sterowników, zwykle są to 2-3 sztuki. Dystrybutorami Specialixa w Polsce są Talex <http://www.talex.com.pl> i Veracomp <http://www.veracomp.com.pl>. Podłączenie sterowników do SXDC8/DX wykonuje się przewodem null-modem DB9-DB25 męsko-męski. Podłączenie do SXDC8/MX wykonuje się przewodem 1-do-1 DB9-DB25 męsko-żeński. Podłączenie do SXDC8/RJX wiąże się z koniecznością zastosowania przejściówki DB9 męski z gniazdkiem na RJ45. Przewód połączeniowy RJ45 ma być 1-do-1. W przejściówce należy połączyć przewody (na wtyczce RJ45 pin 1 jest pierwszym z lewej patrząc na wtyczkę od strony przewodu przy zatrasku skierowanym do góry):

- GND - czerwony (pin 4 na RJ45) - pin 5 na DB9 do sterownika,
- TxD - zielony (pin 5 na RJ45) - pin 3 na DB9 do sterownika,
- RxD - żółty (pin 6 na RJ45) - pin 2 na DB9 do sterownika.

Istnieją dwa urządzenia Moxya, które dobrze współdziałają z Linuksem. Karty Moxa C320Turbo i C320Turbo/PCI wraz z modułami rozszerzającymi C32030T i C32045T (są też inne moduły) są - podobnie jak Specialix - kartami wieloportowymi. Występują w różnych odmianach, na stronie <http://www.moxa.com> można znaleźć więcej informacji na ten temat. Charakteryzują się doskonałym wsparciem dla Linuksa, ze strony firmy Moxa można ściągnąć drivery do Linuksa, ale dostępne są one domyślnie w jądrach systemu. Wersją ograniczoną do 8 portów jest karta C168H/PCI z modułem OPT-8A. Stosuje się ją dla instalacji ograniczonych w sensie ilości sterowników, gdzie ważne jest maksymalne ograniczenie nakładów inwestycyjnych. Instalacja jest równie łatwa i dokładnie opisana w helpach na dołączonym do karty CD-ROMie, jak w przypadku C320Turbo.

Dystrybutorem firmy Moxa jest firma Elmark [www.elmark.com.pl](http://www.elmark.com.pl) (<http://www.elmark.com.pl>).

Dystrybutorem wtyczek i obudówek do wtyczek DB25, DB9 i RJ45 jest np. Meditronik <http://www.meditronik.com.pl>.

Jeżeli potrzebujemy mniejszej ilości portów, możemy wykorzystać kartę dwuportową PCI. Dobrze współpracują z Linuksem karty na układzie firmy NetMos (wystarczy ustawienie portu i przerwanie komendą **setserial**).

Jeszcze innym rozwiązaniem jest użycie przejściówki USB/RS232. Testowane były najpopularniejsze przejściówki na układzie PL-2303 formy Prolific.

### 2.1.12. UPS

Na polskim rynku występuje wiele modelei UPSów dobrze współpracujących z Linuksem, między innymi: APC <http://www.apcc.com>, który zresztą dostał certyfikat RedHat Ready, Powerware <http://www.powerware.com>, Ever <http://www.ever.com.pl>, oraz Fideltronik <http://www.fideltronik.com.pl> produkujący linię UPSów line interactive Ares z darmowym oprogramowaniem UPS Monitor 1.0 działającym poprawnie w RedHat v6.1. Stosowane z powodzeniem UPSy Fideltronika to: Ares 300, 500, 700, 800, 1000, 1600, 3000. Producentem przejściówek do komunikacji po RS-ie między UPS Ares a komputerem jest Sterkom (zalecane) lub Fideltronik. Jeżeli używamy nowego kabla Fideltronika, to przejściówka nie jest potrzebna. Obecnie najczęściej stosowane w instalacjach SZARP-a są UPS-y APC, stosowane były także UPS'y Ever, ale wycofujemy się z nich ze względu na słabą jakość.

Rozpowszechnienie systemów plików z journalingiem powoduje, że poprawna współpraca z oprogramowaniem zamykającym system przestaje być także warunkiem krytycznym. W większości przypadków, jeżeli dłuższe zaniki napięcia nie zdarzają się zbyt często, można darować sobie komunikację między UPS'em i komputerem, więc wybór konkretnego urządzenia jest bardzo ułatwiony.

### 2.1.13. Modem

Polecany jest zewnętrzny, pełny modem z portem RS232. Każdy taki modem będzie dobrze współpracował z Linuksem.

Czasami lepszym rozwiązaniem będzie jednak modem wewnętrzny (np. gdy potrzebujemy dwóch portów szeregowych do podłączenia sterowników). Poza tym modemy wewnętrzne są zwykle sporo tańsze od zewnętrznych. Niestety, zdecydowana większość z nich to tzw. soft-modemy, wymagające współpracy procesora komputera i odpowiedniego oprogramowania. W związku z tym nie należy ich stosować przy słabych procesorach, a poza tym należy się upewnić, że modem ma sterowniki pod Linuksa (co zdarza się nieczęsto). Wsparcie pod Linuksa posiadają m.in. modemy oparte na chipach Conexant, np. modemy Pentagram (modele Diablo 56I, Fanatic 56I, Omen2 56I, Hex 56I, Hex2 56I). Modele Diablo i Hex2 były z powodzeniem używane w instalacjach SZARP'a.

### 2.1.14. Terminale ISDN

Wewnętrzne, karty rozszerzeń PCI. Dotychczas zastosowano z powodzeniem:

- Karta bazująca na chipsecie HFC 2BDS0 PCI,
- Karta bazująca na NETjet PCI,
- Karta bazująca na układzie Winbond 6692.

## 2.2. Terminal

Maszyna z wymaganiami co do niezawodności mniejszymi niż serwer. Jeżeli ma pracować pod kontrolą Linuksa, należy zwrócić uwagę na dobór sprzętu podobnie jak dla serwera.

## 2.3. Ustawienia BIOS'u serwera

W BIOSie serwera - lub jumperami (obecnie niespotykane) - należy ustawić następujące opcje:

- Ustawić zegar na UCT: w zimie jest to czas Polski - 1 godzina, w lecie czas Polski - 2 godziny. Czas letni zaczyna się w nocy z ostatniej soboty na niedzielę w marcu, a czas zimowy zaczyna się w nocy z ostatniej soboty na niedzielę w październiku.
- Urządzenia IDE, których nie ma ustawić na None, chociaż ustawienie na Auto nie powinno zaszkodzić, a może być przydatne przy częstej zmianie konfiguracji.
- Ustawić **Halt on all but keyboard**.
- Wyłączyć hasło na BIOS-a: **Password** ustawić na Disabled.
- Można wyłączyć support dla USB (także przydział IRQ), jeżeli nie będzie używane.
- OS/2 > 64MB Disabled.
- Wyłączyć oszczędzanie energii.
- Włączyć wstawianie po zaniku zasilania: **System After AC Back** ustawić na Full-On albo **State After Power Failure** na On. Uwaga: niektóre stare płyty główne (np. Soltek SL-65DV2, SL-65KV2-T) wymagają dodatkowo uaktywnienia odpowiedniej zworki na płycie głównej.
- W maszynie pełniącej funkcję serwera można wyłączyć zintegrowaną kartę muzyczną (o ile płyta posiada taką).
- Kolejność bootowania (jeśli zostały już zainstalowane wszystkie OS-y): dysk twardy, CD-ROM.
- **Plug&Play OS** ustawić na No.
- Wyłączyć detekcję obecności wirusów w sektorze startowym: **Boot virus detection** lub **Anti-Virus Protection** lub **Virus Warning** ustawić na Disabled.
- **Quick boot test** Enabled.
- **Boot floppy seek** Disabled.
- W nowych BIOSach istnieje monitoring parametrów fizycznych pracy procesora i płyty głównej; należy w takim przypadku ustawić:
  - alarm w przypadku wyłączenia wiatraczka procesora,
  - alarm w przypadku przekroczenia temperatury procesora powyżej zadanej 65-75st.C,
  - alarm w przypadku przekroczenia temperatury płyty głównej powyżej zadanej.
- Włączyć kontroler SCSI, jeśli jest zintegrowany z płytą.
- Uaktywnić terminator na kontrolerze SCSI, jeśli jest zintegrowany z płytą. Jeżeli w serwerze jest kontroler SCSI Adaptec, w jego BIOSie należy ustawić opcje (aby dostać się do BIOSu SCSI podczas startowania systemu należy wcisnąć **Ctrl-A**, zaś w przypadku kontrolera SCSI Tekram należy wcisnąć **F2** bądź **F6**):
  - **Parity checking** Enabled.
  - **Termination** Automatic.
  - **Boot Device Configuration** - SCSI ID musi się zgadzać z ustawionym jumperami na dysku, najlepiej jeśli jest 0:
    - **SCSI ID** 0,
    - **Lun** (zawsze) 0.
- W sekcji **SCSI Device Configuration**:
  - **Initiate Sync** Yes.

- **MaxSync Transfer** 40 dla Wide SCSI, 20 dla Narrow SCSI.
- **Enable Discon** Yes.
- **Initiate Wide Neg.** Yes (tylko dla Wide SCSI).
- **Send Start** No.
- **BIOS Multiple LUN Supp.** No.
- **Include BIOS Scan** Yes.
- **Sekcja Advanced Configuration Options:**
  - **P&P SCAM** Enabled (urządzenia SCAM póki co są dość rzadkie).
  - **Reset SCSI at Init** Enabled.
  - **Extended Translation for DOS Drives > 1GB** Enabled.
  - **Host BIOS** Enabled.
  - **Support Removable Disks Boot Only** Disabled.
  - **Display Ctrl-A** Disabled.
  - **BIOS Support for boot CD-ROM** Enabled.
  - **BIOS Support for Int 13** Enabled.

W celu skrócenia czasu uruchamiania komputera warto jest wyłączyć funkcję Boot PROM na karcie sieciowej (jeśli karta to wspiera).

Jeżeli instalujemy kartę dwuportową PCI, konieczne może być właśnie wyłączenie nieużywanych urządzeń (USB, modem, karta dźwiękowa, port drukarki) w celu zwolnienia przerwań dla karty. Wyłączenie zbędnych urządzeń przyspieszy także start systemu.

## 3. Konfiguracja i instalacja oprogramowania

### 3.1. DOS

Niegdyś zalecane było utworzenie na dysku jednej partycji z systemem DOS i instalacja na niej np. programu CID do edycji plików dBase IV. Obecnie nie ma już żadnego powodu aby instalować DOS'a. Dwa kolejne rozdziały opisują więc uruchamianie Linuksa z poziomu DOS'a (też zaszłość historyczna - lepiej wykorzystać menedżera ładowania takiego jak GRUB czy LILO), oraz instalację wolnego kłona DOS'a - systemu FreeDOS.

#### 3.1.1. Uruchamianie Linuksa spod DOS-a

- Aby móc odpalić Linuksa spod DOS-a, w config.sys trzeba wstawić linijkę:

```
INSTALL=loadlin.exe c:\linux\autoboot\mlinuz root=/dev/hda4
```

i przy bootowaniu wybrać odpowiednią sekcję (należy podzielić `autoexec.bat` i `config.sys` na sekcje do uruchamiania DOS/Windows lub Linuksa). Można też uruchomić `loadlin.exe` z linii komend.

- Graficznym interfejsem do `loadlin.exe` działającym pod Windows9x jest `ux-1.0`. Podobnie jak `loadlin.exe` nie wymaga zainstalowanego w Master Boot Sector `lilo`. Po ściągnięciu `ux-1.0.zip` jego instalacja i konfiguracja sprowadza się do:
  - rozpakowania plików w katalogu np. `C:\UX`
  - przekopiowania jądra z Linuksa do `C:\` pod nazwą `vmlinuz`
  - uruchomienia `ux.exe` i ustawienia:
 

```
Linux kernel image: C:\vmlinuz
Root device: /dev/hda3
```

 (w polu `/dev/hda` trzeba podać partycję linuxową - jak znaleźć jej nazwę - opisuje to Sekcja 3.2.7), reszta ustawień niezmienniona
  - za każdym następnym razem po uruchomieniu `ux.exe` wystarczy nacisnąć GO!
  - to GPL i pełne źródła są dołączone - można spolszczyć plik `resources`

Wadą `ux-1.0` jest, że uruchomiony w ten sposób Linux może nie widzieć niektórych urządzeń.

### 3.1.2. FreeDOS

W sytuacji gdy potrzebujemy aplikacji działających pod DOS'em i z jakiegoś powodu nie możemy wykorzystać `dosemu` (emulatora DOS'a pod Linuksa), warto rozważyć skorzystanie z systemu FreeDOS. Zwłaszcza, że oryginalny system MS-DOS lub PC-DOS jest już w tej chwili właściwie niedostępny. FreeDOS zapewnia praktycznie 100% emulację oryginalnego DOS'a w wersji 3.3/6.22 (jądro / programy użytkowe). Poza tym zapewnia lepszą obsługę sprzętu niż oryginał, w tym nie posiada większości DOS-owych ograniczeń dotyczących układu partycji na dysku. Ma wbudowaną zaawansowaną obsługę wiersza poleceń (także z dopełnianiem poleceń), często dystrybuowany jest np. razem z edytorem Vim czy przeglądarką Lynx. FreeDOS jest tworzony na licencji GPL, w związku z tym jest kompletnie darmowy.

Strona [www.freedos.org](http://www.freedos.org) (<http://www.freedos.org>) zawiera między innymi dokumentację dotyczącą systemu, można z niej ściągnąć także obrazy dyskietek lub CD-ROM'u wykorzystywane do instalacji systemu.

Ze względu na szybkość i wygodę polecana jest raczej instalacja z CD-ROM'u. W katalogu <http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/distributions/> znajdują się podkatalogi z dystrybucjami FreeDOSa, skąd należy ściągnąć obraz dysku instalacyjnego `fdboot.iso`, np. <http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/distributions/beta9rc3/fdbootcd.iso>. Należy wypalić CD-R z tym obrazem wydając komendę:

```
cdrecord -v speed=0 dev=0,4,0 fdbootcd.iso
```

Następnie należy zbootować komputer z tak przygotowanej płytki. Dokładny sposób instalacji różni się nieco w zależności od wersji zwłaszcza, że instalator systemu znajduje się obecnie w fazie dość intensywnego rozwoju. Niemniej zawsze powinniśmy mieć możliwość instalacji systemu (co może wymagać przygotowania partycji za pomocą programu `fdisk` lub odpowiedniej graficznej nakładki na niego), jak i załadowania systemu bezpośrednio z płyty. Mamy też możliwość przygotowania dyskietki startowej (w niektórych wersjach zdarzają się błędy w przygotowywaniu dyskietki - komputer nie chce ładować z niej systemu).

Należy zwrócić uwagę, że jeżeli instalujemy system na istniejącej instalacji innego DOS'a, bez formatowania partycji, to, aby ładował się nasz nowy FreeDOS, należy użyć komendy SYS. Możemy przygotować dyskietkę startową (odpowiednią opcję znajdziemy w menu po załadowaniu systemu z płyty instalacyjnej), a następnie wydać komendę:

```
A:\FREEDOS\SYS A:\ C:
```

FreeDOS może korzystać z nieco innego układu katalogów niż oryginalny DOS, np. większość elementów systemu może się znajdować w katalogu C:\FDOS\BIN. Inne mogą być także nazwy plików startowych systemu: C:\FDCONFIG.SYS i C:\FDOS\FDAUTO.BAT. Inne są nazwy np. dla programu zarządzającego pamięcią XMS/EMS. Nieco inaczej w porównaniu z DOS'em 6.22 są zrealizowane rozszerzenia takie jak np. menu startowe systemu. Warto skorzystać z generowanych przy instalacji plików startowych jako przykładów, więcej informacji znajduje się w dokumentacji systemu, która podczas instalacji także jest kopiowana na dysk.

## 3.2. Linux

### 3.2.1. Instalacja systemu RedHat Linux

#### 3.2.1.1. Instalacja systemów RedHat 6.X/7.X

**Notatka:** Niniejszy opis odnosi się do starych, kilkuletnich wersji dystrybucji RedHat. Ma znaczenie głównie historyczne i w przyszłości może zostać usunięty.

Niniejszy opis instalacji dotyczy systemu Linux RedHat v6.1, v6.2, v7.2, v7.3. Opis jest też w dużej mierze prawdziwy dla systemów RH8.0/Aurox8.0 (lecz nie został sprawdzony punkt po punkcie). Sposób instalacji najnowszych systemów - RH9.0 i Aurox9.0 opisuje Sekcja 3.2.1.2. Na nowszym sprzęcie zalecane jest instalowanie v7.x (i wyżej), na starszym natomiast może być lepszy v6.2 (instalowanie trwa krócej, starsze ikony lepiej pasują do starszych kart graficznych, wymagane jest mniej miejsca na dysku).

Na tym opisie można się wzorować instalując Linuksa do zastosowań innych niż system SZARP. Staraliśmy się tu opisać częste problemy występujące podczas instalacji i konfiguracji, oraz wyjaśnić pewne niebanalne rzeczy, które wydają się nam warte opisanie.

W przypadku instalacji na dyskach IDE należy pamiętać o ograniczeniach związanych z trybem ATA. Więcej szczegółów opisano w rozdziale Sekcja 2.1.4.

W przypadku CD-ROM ATAPI należy w BIOSie ustawić CD-ROM jako pierwsze urządzenie bootujące i zbootować komputer z kompaktu. Jeśli jest to niemożliwe należy wykonać dyskietkę startową, dokumentacja jak to zrobić znajduje się na CD-ROM'ie z Linuksem.

- na ekranie powitalnym nacisnąć <Enter> (tryb graficzny - zalecane w RH 7.2), lub wpisać: text <Enter> żeby rozpocząć (szybszą) instalację w trybie tekstowym.
- W RH/Auroksie 8.x/9.x pierwszą fazą jest testowanie nośnika (sprawdzenie, czy nie uszkodzone zostały płytki instalacyjne). Ponieważ test ten trwa dosyć długo to można go sobie darować.

- W przypadku trybu graficznego: wybrać odpowiedni typ myszki (PS/2 2/3 klawisze - najlepiej 3 klawisze) i klawiatury (domyślny). W trybie tekstowym wybieranie myszki będzie później (wprowadza się powyższe ustawienia). W RedHat 7.3 myszka wybierana jest po wyborze języka i klawiatury.
- wybrać English jako język instalacji (w przypadku RedHata v6.1 jest również możliwość wybrania jęz. polskiego. Została usunięta w wersji 6.2 i 7.x z powodu błędów).
- wybrać **keyboard** us (w RH 7.2 zostawić ustawienia domyślne: model Generic 105, layout U.S. English, Enable Dead Keys). W RH 7.3 należy wybrać layout Polish.
- wybrać instalację Custom
- należy wybrać program partycjonujący : **fdisk** lub **Disk Druid**. Drugi program (DD) jest prostszy, natomiast **fdisk** ma więcej (przeważnie zbędnych) możliwości, poleca się go raczej doświadczonym „Instalatorom Linuksa”. Dalej opisano instalację za pomocą Disk Druida. W RH 7.3 jest jeszcze możliwość wybrania automatycznego partycjonowania - nie wybierać!
- w ekranie Disk Druid najpierw zrobić partycję swap o rozmiarze 2 \* wielkość pamięci komputera na którym instalujemy Linuksa
- następnie w Disk Druid utworzyć partycję Linux Native z opcją **Grow to fill disk (Use remaining)** - jako punkt zamontowania wybrać "/". W RH 7.2 i 7.3 jako typ partycji wybrać ext3. W RH 7.3 odpowiednia opcja nazywa się "Fill to maximum allowable size".
- W przypadku niektórych kontrolerów SCSI (m.in. Tekram DC-390U2W) należy wyłączyć opcję **Use linear mode**
- Sformatować wszystkie nowo utworzone partycje. Nie wybierać opcji **Check for bad blocks**. Długo trwa.
- Umieścić bootloadera w MBR. Nie podawać żadnych parametrów do kernela. Ustawić domyślne odpalenie Linuksa w LILO. W RH 7.2 i 7.3 jest możliwość wybrania GRUB-a (zalecane). Wybieramy instalację do pierwszego sektora boot-partycji, nie ustawiamy hasła dla GRUB-a (więcej informacji o GRUB-ie zawiera Sekcja 3.2.5).
- w Konfiguracja sieci na etapie instalacji na karcie eth0: Activate on boot: +; IP Address 192.168.1.XXX, gdzie XXX jest różne dla różnych maszyn, netmask automatycznie ustawi się na 255.255.255.0, broadcast automatycznie ustawi się na 192.168.1.255; nazwa hosta - dla serwera powinna być zgodna z prefixem bazy (np. kato), dla terminali - właściwie dowolna, ale pomocne jest by była ona podobna do nazwy serwera (np. kat1). **NIE WPISYWAĆ default gateway!** Jeśli instalator ustawi tam jakąś wartość to ją wywalić. W RH 7.3 trzeba odznaczyć wcześniej opcję "Configure using DHCP".
- w RH 7.2 i 7.3 pojawi się opcja ustawienia firewala - w przypadku komputera niepodłączonego bezpośrednio do internetu wybieramy "No firewall". Także w RH 7.2 pojawi się ekran wybory dodatkowych języków - wybieramy "Polish" i jako domyślny także ustawiamy "Polish".
- w ekranie wyboru strefy czasowej wybrać "System clock uses UTC"/"Hardware clock set to GMT" i wybrać Warsaw w zakładce "Location"
- ustawić hasło roota i jednego usera - np. palacz; należy mu ustawić dowolne hasło, a potem aby ustawić mu hasło puste należy po instalacji jako root uruchomić:

```
passwd -d <username>
```
- ustawić **authentication options**: shadow i MD5 - Enabled. NIS, LDAP, Kerberos i SMB (RH 7.2 i 7.3) - Disabled.

- w ekranie wyboru pakietów do instalacji wybrać wszystkie (oznaczyć każdą grupę pakietów z osobna), dodatkowo oznaczyć opcję „Everything”.
- wybrać "Use graphical login"
- przetestować grafikę: wybrać poprawnie ilość pamięci na karcie (np. zapamiętać przy bootowaniu). Jeżeli nie może sam wykryć karty, wybrać Customize X Configuration i tam najmniejsze co może być to: 640x480x8; jeżeli nie działa po prostu pójść dalej bez testowania X-ów (zostawić Use graphical login ustawiony).
- skip boot disk creation
- reboot już z dysku twardego
- /tmp/install.log - zapis historii instalacji
- instalator może mieć kłopoty z prawidłową detekcją parametrów karty graficznej - w takim przypadku należy po instalacji rekonfigurować przy pomocy /usr/X11R6/bin/Xconfigurator, jeżeli nie wykryje sam karty, wybrać ją z listy; monitor np. Samsung 17GLsi; wybrać tryb 1024x748x8 i ustawić żeby X-y automatycznie startowały przy boocie. W RH 7.x należy wybrać jako domyślne środowisko graficzne KDE.

W przypadku, gdyby to narzędzie też nie dawało efektu należy ręcznie uzupełnić wpisy w pliku /etc/X11/XF86Config na podstawie odpowiedniego pliku README dla danej karty graficznej /usr/X11R6/lib/X11/README.XXX; w przypadku dokonywania takich zmian należy zrobić kopię bezpieczeństwa istniejącego pliku XF86Config, na wypadek pomyłki..

W przypadku niektórych starych kart graficznych lepiej jest zainstalować starsze X-y - w wersji 3.3.6 - mają one lepsze wsparcie dla starego sprzętu. Instalację starszych X-ów przeprowadzamy przez instalację pakietu XFree86-SVGA-3.3.6-\*.rpm (dla innych kart, wymagających innego X-serwera, należy oczywiście wybrać odpowiedni pakiet, typu XFree86-S3-3.3.6-\*.rpm).

W przypadku kombinacji X'ów w wersji 3.3.6 (dostarczanych np. z RedHatem 6.2) i niektórych kart S3 Trio występują problemy z kursorem - zamiast niego wyświetlany jest biały kwadrat. W takim przypadku należy dokonać ręcznej modyfikacji pliku /etc/X11/XF86Config należy dodać w sekcji "Device" opisującej daną kartę graficzną linię postaci:

```
Option "sw_cursor"
```

Przy instalacji X'ów w wersji 3.3.6 z kartą S3 Vision 964 Paradise do pliku /etc/X11/XF86Config należy do sekcji "Device" dodać wpisy, tak aby cała sekcja wyglądała następująco:

```
Section "Device"
Identifier "Vision 964-P"
VideoRam 1024
Ramdac "bt9485"
Option "nonlinear"
Option "nomemaccess"
ClockChip "ics9161a"
```

Jeśli używamy XFree86 w wersji co najmniej 4.0.1 (można to sprawdzić przez XFree86 -version), to być może najprostszym sposobem skonfigurowania X-serwera będzie wpisanie XFree86 -configure. Ta wersja X-ów wprowadza wiele innych zmian - inny jest format wspomnianego wyżej pliku XF86Config, jest tylko jeden wspólny X-serwer, a dla poszczególnych kart są oddzielne sterowniki. W przypadku problemów warto zaopatrzyć się w jak najnowszą wersję XFree86 - im nowsza, tym więcej kart powinna obsługiwać.

Istnieją karty graficzne, które ani nie są wykrywane przez instalatora ani nie można ich wybrać z listy dostępnych kart graficznych. Z takim przypadkiem można się spotkać (przy najmniej na razie) w przypadku kart z chipsetem Intelu i810/i815. Na szczęście Intel uruchomił support Linuksa pod

adresem <http://support.intel.com/support/graphics/intel815/linuxsoftware.htm>. Należy stąd ściągnąć pliki (mogą się pojawić nowsze wersje): I810Gtt-0.2-4.src.rpm i XFCom\_i810-1.2-3.i386.rpm oraz [release\\_linux.pdf](#) (z linku [ReleaseNotes](#)). Ostatni z nich (pdf) to opis instalacji i konfiguracji sprzętu. Dwa pierwsze to odpowiednio: moduł do jądra i X-server. Można również ściągnąć patcha na źródła XServera (XFCom\_i810.1.2.patch.tar.gz), ale rozwiązanie z gotowym serwerem (prezentowane poniżej) jest chyba lepszym rozwiązaniem. Należy więc wydać polecenia:

```
rpm --recompile I810Gtt-0.2-4.src.rpm
rpm -Uvh XFCom_i810-1.2-3.i386.rpm
```

Można przetestować czy moduł **agpgart** daje się załadować (`modprobe agpgart`) i czy `/etc/X11R6/bin/XF86_SVGA` jest linkiem na `/etc/X11R6/bin/XFCom_i810`. I teraz najbardziej przykra część instalacji: ręczna modyfikacja pliku `/etc/X11/XF86Config`. Zmieniamy odpowiednio sekcje "Device" i "Screen" (tutaj **Driver** "svga" i **Subsection** odpowiednio do pamięci na karcie i rodzaju monitora). Przeważnie należy również zmodyfikować sekcję "Monitor", szczególnie **HorizSync** i **VertRefresh**. Czasem wypada też poprawić sekcję "Pointer", opisującą mysz.

Po powyższych zmianach możemy już próbować testować ikisy (**startx**).

- Jeszcze innym przypadkiem trudności z uruchomieniem karty graficznej, może być nowa karta nVidii, która na szczęście ma dobry support do Linuksa <http://www.nvidia.com>, niestety nie do każdego (poniżej opisano to dla wspieranego Red Hat v7.2). Konieczna jest instalacja dwóch rpm'ów: GLX i interfejsu między kerneliem, a tymże GLX. Tego ostatniego trzeba instalować najpierw, można to zrobić z pakietu binarnego `NVIDIA_kernel-1.0-2880.rh72up`, ale lepiej zrobić to ze źródła `NVIDIA_kernel-1.0-2880.src.rpm` - szczególnie jest to konieczne w przypadku kernela innego niż oryginalne z Red Hat v7.2:

```
rpm --rebuild NVIDIA_kernel-1.0-2880.src.rpm
rpm -ihv /usr/src/redhat/RPMS/i386/NVIDIA_kernel-1.0-2880.i386.rpm
rpm -ihv NVIDIA_GLX-1.0-2880.rh72up.i386.rpm
```

- jeżeli nie zainstalował się odpowiedni XServer, należy go zainstalować ręcznie:
  - jako root:
 

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```
  - potem:
 

```
rpm -ihv /mnt/cdrom/RedHat/RPMS/<brakujący_XServer>
```

 gdzie brakujący serwer to może być np. `XFree86-SVGA-3.3.5-3.i386.rpm`. Czasami może być konieczne użycie innych opcji do programu `rpm`, np. `rpm -Uvh` albo `rpm -Uvh --force`.
  - uruchomić Xconfigurator ponownie
- aby przełączyć mysz z np. COM2 na port PS/2:
  - uruchomić `mouseconfig`
  - przestawić w `/etc/sysconfig/mouseconfig` na "ps2" (taki jest parametr w wywołaniu `gpm -t`)
  - sprawdzić, czy odpalany jest przy starcie `gpm` (link w `/etc/rc.d/rc5.d`)
  - ustawić w `/etc/X11/XF86Config` w sekcji "Pointer" typ na "PS/2"
- jeżeli po instalacji po reboocie przy wybraniu jako bootmanagera `lilo` komputer nie startuje, tylko wypisuje coś w rodzaju:

```
L 80 80 80 80 80 80 80 80 80 80 ...
albo:
```

```
L
L
L
L
```

może to oznaczać (szczególnie w przypadku kontrolerów SCSI), że w trakcie instalacji błędnie została zaznaczona opcja **Use linear mode**. Aby temu zaradzić należy w pliku `/etc/lilo.conf` zamienić linijkę:

```
linear
na:
lba32
```

- sprawdzenie czy karta sieciowa zainstalowała się poprawnie: jako root odpalić

```
/sbin/ifconfig
```

- powinien pokazać się `eth0`; dla próby jeszcze odpalić

```
ping <ip_jakiegoś_podłączonego_komputera>
```

jeżeli nie działa, pójść do DOSa, odpalić program do ustawiania parametrów karty sieciowej i sprawdzić, czy jest właściwe medium - lepiej ustawić ręcznie RJ45/BNC niż auto. Poza tym dla kart, które obsługują funkcję Boot PROM - lepiej tę funkcję wyłączyć.

- aby karta sieciowa zainstalowała się, w systemie musi być odpowiedni moduł

```
/lib/modules/<kernel_number>/net/<modul_karty>.o
```

- jaki moduł potrzebny jest do jakiej karty sieciowej można odczytać w

`/usr/doc/HOWTO/Ethernet-HOWTO` (parametr Driver Name przy opisie danej karty); potem należy wydać jako root komendę:

```
lsmod
```

i sprawdzić, czy odpowiedni moduł nie jest już załadowany - jeżeli jest, problem prawdopodobnie tkwi w karcie; jeżeli nie, należy wydać komendę:

```
insmod <modul_karty>
```

(uwaga: podajemy nazwę modułu bez rozszerzenia `.o`) (dostępne opcje podane są w

`/usr/src/linux/Documentation/networking/net-modules.txt`); komenda:

```
rmmmod <modul_karty>
```

usuwa driver

żeby

```
/sbin/ifconfig
```

pokazał `eth0` (prawidłowo zainstalowana sieć na karcie) należy uruchomić:

```
/etc/rc.d/init.d/network restart
```

żeby samo wchodziło przy starcie komputera, w pliku

```
/etc/conf.modules
```

należy wstawić linijkę:

```
alias eth0 <modul_karty>
```

- Można użyć także narzędzia do półautomatycznej instalacji karty sieciowej. Może to być konieczne także jeśli karta zostanie źle wykryta przy instalacji, bądź trzeba zmienić numer IP komputera.

W RH6.x oraz RH7.0, 7.1 i 7.2 dostępne jest narzędzie **netconf**, które pracuje w trybie tekstowym jak i graficznym. Tu opiszemy wersję tekstową tego narzędzia.

Po uruchomieniu należy wybrać z menu opcję **Basic host configuration**, wcisnąć **Enter**, po czym wpisać następujące informacje: nazwę komputera (Host name), w sekcji Adaptor 1 (jeśli konfigurujemy pierwszą kartę) zaznaczyć opcje *Enabled* i Config Mode - Manual. Potem wpisujemy

jeszcze raz nazwę komputera pełną (z nazwą domeny), a więc np. mail.praterm.com.pl. Kolejne pole, które musimy wypełnić, to numer IP komputera (IP address), w notacji z kropkami (a więc np. 192.168.1.1). W polu Mask wpisujemy maskę sieci (w sieci lokalnej jest to zwykle 255.255.255.0), a na koniec podajemy jeszcze nazwę urządzenia (dla pierwszej karty jest to eth0).

Potem klawiszem **Tab** przechodzimy do przycisku **Accept** i **Enterem** akceptujemy wprowadzone zmiany. Jeżeli komputer ma być podłączony do Internetu przez inny komputer (bramkę), to należy jeszcze wpisać numer IP komputera będącego bramką (routera) w pole, które pojawi się po wybraniu z menu głównego opcji *Routing and gateways*, a następnie *Set default*. Należy także zaznaczyć opcję *Enable route*.

W najnowszych wersjach RedHata (RH7.3, RH8.x, RH9.x) program **netconf** został usunięty. W tym przypadku powinniśmy skorzystać z programu **redhat-config-network** (jednego z suity programów redhat-config), który ma tylko wersję graficzną.

Po uruchomieniu możemy kliknąć zakładkę Hardware, żeby przekonać się jakie urządzenie mogą być potencjalne wykorzystane do kontaktu z siecią. Z reguły będzie to karta sieciowa (eth0).

W zakładce Devices możemy kartę tę skojarzyć z jednym lub więcej numerem IP. W tym celu klikamy przycisk Add, wybieramy urządzenie Ethernet i zatwierdzamy (OK). Pojawi się nowe okno. Po pierwsze musimy nadać temu skojarzeniu jakąś nazwę (Nickname). Zwykle, dla porządku, skojarzenia nazywane są tak jak urządzenia, do których są przypisane (eth0, eth1 itd). Aliasy IP mają dodatkowo numer aliasa po dwukropku (np. eth0:0, eth0:1 itd). Opcję "Activate Device when Computer Starts" zostawiamy włączoną i przechodzimy do zakładki Protocols. Klikamy na przycisk Edit, co spowoduje wyświetlenie się kolejnego okienka.

Jeżeli nie korzystamy z DHCP (z reguły nie korzystamy) to wyłączamy opcję "Automatically obtain IP address settings with" i ustawiamy odpowiednio adres IP, maskę podsieci i gateway (maska podsieci dla sieci prywatnych to zwykle 255.255.255.0, a w sieciach bez dostępu do Internetu pole gateway można pozostawić puste). W zakładce Hostname wpisujemy nazwę naszego hosta natomiast do zakładki Routing w ogóle nie wchodzimy (chyba że chcemy stworzyć jakieś zaawansowane tablice routingu). Całość zatwierdzamy (OK).

Jesteśmy teraz znowu w okienku Ethernet Device. Klikamy na zakładkę Hardware Device i tam wybieramy które urządzenie chcemy konfigurować. Jeśli dodajemy alias IP to musimy zaznaczyć opcję "Enable Device Alias Support" i ustawić numer aliasu (zwykle 0). Całość zatwierdzamy. I przechodzimy do okienka głównego (Network Configuration).

Klikając na zakładce Hosts możemy zmodyfikować plik `/etc/hosts` (można to zrobić również później - ręcznie). Zakładka DNS pozwoli nam ustawić nazwę hosta (Hostname) i naszą domenę (Domain), chociaż tę można zostawić pustą (jest to odpowiednik domain z `resolv.conf-a`). Najważniejszą funkcją jest możliwość dodania numerów IP serwerów DNS. Można również ustawić tzw. Search Domain (czyli search z `resolv.conf-a`). Jak łatwo zauważyć, wszystkie powyższe czynności w zakładce DNS można również zrobić później (modyfikując plik `/etc/resolv.conf`). Zmiany zatwierdzamy przyciskiem Zastosuj a następnie Zamknij.

- W RH 6.x instalacja w przypadku wyboru zarówno pakietu Gnome jak i KDE domyślnie ustawia jako graficzny menedżer logowania gdm, z którym są kłopoty w sieci. Aby zmienić gdm na kdm należy jako root w pliku `/etc/X11/prefdm` ustawić linijkę:

```
preferred=kdm
```

tuż przed warunkiem pod koniec pliku:

```
if [ -n "$preferred" ] && which $preferred >/dev/null 2>&1; then
    exec `which $preferred` $*
```

aby zmiany zostały uwzględnione należy zrestartować system lub wpisać komendy:

```
init 3
```

```
init 5
```

Jeśli to nie zadziała - dalej będzie startował gdm, lub X nie wystartuje w ogóle (pokaże się komunikat:

```
INIT: Id "x" respawning too fast: disabled for 5 minutes
```

trzeba wówczas przeładować system i przy starcie systemu po napisie LILO: napisać:

```
linux single
```

(dostępny jest wówczas bash bez uruchamianych X-ów) w `/etc/inittab` należy na miejscu linii:

```
x:5:respawn:/etc/X11/prefdm -nodeamon
```

wstawić linię:

```
x:5:respawn:/usr/bin/kdm -nodeamon
```

- standardowo po instalacji z wybraną opcją "Use graphical login" serwer startując w runlevel 5 (linia `id:5:initdefault` w pliku `/etc/inittab`) uruchamia X'y także na serwerze; jeżeli chcemy, żeby login w X'ach uruchamiany był na wszystkich terminalach, ale nie na serwerze (w celu zniechęcenia użytkowników do dotykania serwera) należy w pliku `/etc/X11/xdm/Xservers` wykomentować (wstawiając na początku znak "#") linię:

```
# :0 local /usr/X11R6/bin/X
```

- aby przełączyć desktop w trakcie pracy należy pod Gnome/KDE uruchomić: Panel | System | Desktop Switching Tool, uruchamiający program:

```
/usr/bin/switchdesk
```

lub z poziomu konsoli:

```
/usr/bin/switchdesk-helper
```

- w Red Hat v6.1 przy logowaniu do serwera z terminala spod Windows w KDE pojawia się komunikat: "Sorry your X11 server lacks DPMS support"; odpowiedzialny za to jest element KDE - `kcmdpms` - żeby ten komunikat nie wyskakiwał przy logowaniu się w KDE należy w pliku `/usr/share/applnk/Settings/Desktop/kcmdpms.kdelnk` linię:

```
Init=kcmdpms -init
```

zamienić na:

```
Init=
```

- czas jest ustawiany w skrypcie `/etc/rc.d/rc.sysinit` wywołanie:

```
/sbin/hwclock --hctosys -u
```

(w `/etc/rc.d/rc.sysinit - $CLOCK $CLOCKFLAGS`); jeżeli `hwclock` na podstawie RTC nie ustawia prawidłowo czasu systemowego należy sprawdzić, czy plik `/etc/localtime` jest kopią (albo soft linkiem) pliku `/usr/share/zoneinfo/Poland`, lub

```
/usr/share/zoneinfo/Europe/Warsaw, lub /usr/share/zoneinfo/posix/Poland
```

(wszystkie te pliki są identyczne), co zapewnia właściwą zmianę czasu na letni i zimowy; plik

```
/etc/sysconfig/clock
```

powinien mieć postać:

```
UTC=true
```

```
ARC=false
```

```
ZONE="Poland"
```

Ostatnia linijka nie jest konieczna.

- jeżeli trzeba zmienić nazwę komputera ustawioną przy instalacji należy w pliku

```
/etc/sysconfig/network
```

ustawić odpowiednio linię

```
HOSTNAME=hostname.hostdomain
```

Ponadto nazwę hosta w formacie `hostname.hostdomain` należy umieścić w pliku `/etc/HOSTNAME`.

- Żeby poprawnie działała używana często w plikach konfiguracyjnych SZARPa komenda `hostname -s` należy pamiętać o wpisaniu w pliku `/etc/hosts` postaci np.:

```
127.0.0.1 hostname hostname.hostdomain localhost localhost.localdomain
albo - zakładając, że numer IP komputera to 192.168.1.2:
```

```
127.0.0.1 localhost localhost.localdomain
192.168.1.2 hostname hostname.hostdomain
```

- dodawanie użytkowników z poziomu konsoli:

```
adduser <username>
```

żeby użytkownik nie miał hasła (może się wtedy logować nie podając hasła):

```
passwd -d <username>
```

- pewne rzeczy można wyrzucić z `/etc/rc.d/rc5.d`, aby system startował szybciej i był mniej obciążony. W katalogu `/etc/rc.d/rc5.d` powinny zostać linki:

```
K01parstart
K03smb
K10linuxconf
K20gpm
K30xfs
K40keytable
K50lpd (opcja, patrz niżej)
K60inet
K65crond
K70atd (opcja, patrz niżej)
K75syslog
K80random
K85portmap (opcja, patrz niżej)
K90network
S10network
S11portmap (opcja - potrzebny do NFS'a)
S20random
S30syslog
S40atd (opcja - można tego nie zamieszczać)
S41crond
S50inet
S60lpd (opcja - tylko gdy chcemy drukować lokalnie bądź zdalnie)
S75keytable
S90xfs
S91gpm
S97linuxconf
S94smb
S98parstart
S99local
```

oczywiście jest to minimum, mogą się przydać jeszcze: `http`, `nfs`, `netfs`, `upsd`, `smb`, `nfslock` etc. (numery po literach K i S nie mają znaczenia, byleby linki zachowały tą samą kolejność) do konfiguracji uruchamianych serwisów można użyć narzędzia **ntsysv**.

- Czasem w RH 6.x (co się jednak nie powinno zdarzać) mogą wystąpić błędy na partycji z systemem Linux (może to nastąpić w skutku np. nagłego wyłączenia zasilania). Większość napraw jest on w stanie wykonać bez interakcji z użytkownikiem, należy mu to jednak umożliwić - w pliku `/etc/rc.d/rc.sysinit` należy zamienić linię (~158 linii):

```
initlog -c "fsck -T -a $fsckoptions /"
```

na:

```
initlog -c "fsck -T -y $fsckoptions /"
```

- żeby w KDE w RH 6.x używane były polskie czcionki: uruchomić z menu KDE Centrum Sterowania / Desktop / Czcionki i dla każdej opcji (Czcionka ogólna, Czcionka proporcjonalna, Czcionka tytułów okna, Czcionka przycisków panelu, Czcionka zegara panelu) wybrać Kodowanie iso-8859-2, nacisnąć Zastosuj.

By w RedHat 6.2 standardowa czcionka Helvetica obsługiwała poprawnie kodowania iso-8859-2 należy zmienić czcionkę przed zmianą języka na Polski (patrz niżej).

- żeby w KDE w Rh 6.x używany był język polski: uruchomić z menu KDE Centrum Sterowania (KDE Control Center) / Desktop / Język (Language) i ustawić wszystkie na polski, nacisnąć Zastosuj (Apply); po tym najlepiej zamknąć i powtórnie uruchomić KDE Centrum Sterowania - już z odpowiednimi ustawieniami
- żeby w KDE w RH 6,x używana była polska klawiatura programisty: uruchomić z menu KDE Centrum Sterowania / Urządzenia wejściowe / Międzynarodowe Ustawienia na karcie Ogólne w opcji Mapy klawiatury dodać "Pl Język: Polski. Kodowanie: domyślne. Klawiatura programisty" (uwaga - jeśli nie działa, wybierać "nPl Język: Polski. Kodowanie: domyślne. Klawiatura Międzynarodowa"); istniejącą klawiaturę amerykańską usunąć; na karcie Start zaznaczyć opcję Autostart; W oknie Zaawansowane wybrać "Codes for pc101 keyboard" i Menu globalne; nacisnąć Zastosuj - zmiany zostaną uwzględnione dopiero przy następnym loginie
- W RH 7.2 zamiast powyższych działań w celu ustawienia polskich liter należy po pierwszym zalogowaniu się użytkownika zaakceptować ustawienia standardowe magika (kraj i język). Pamiętać należy również o wyłączeniu Antyaliasingu (Centrum Sterowania / Czcionki / Stosuj Wygładzanie). W przeciwnym razie niektóre polskie znaki będą źle wyświetlane.

W RH 7.3 polskie litery działają od razu. W celu uzyskania polskich liter na klawiaturze należy w pliku /usr/X11R6/lib/X11/xbk/symbols/pl należy usunąć przecinek po "EuroSign".

- powyższe ustawienia w KDE należy wykonać dla każdego użytkownika oddzielnie
- Żeby na ekranie powitalnym w KDE (w kdm) była właściwa konfiguracja i odpowiednie napisy po polsku: zalogować się jako root, uruchomić z menu KDE Centrum Sterowania / Aplikacje / Menedżer logowania / karta Wygląd / opcja Wygląd / tekst powitalny - np. "Witamy w SZARP v2.1", opcja Język - polski; karta Czcionki / Wybierz czcionkę / przycisk Zmień czcionkę dla każdej opcji w menu: Powitanie, Błąd, Zwyczajła: w okienku Wybierz czcionkę / Żądana czcionka / Kodowanie iso-8859-2, nacisnąć OK; na karcie Użytkownicy przeciągnąć wybranych użytkowników, którzy będą się logować w KDE z okna Wszyscy do okna Wybrani, zaznaczyć opcję Pokazuj jedynie wybranych oraz Pokazuj; na karcie Sesje: w menu Pozwól na zamknięcie systemu ustawić Tylko konsola (jeżeli wszyscy mają mieć możliwość zamknięcia systemu - ustawić Wszyscy), z okna Dostępne typy usunąć wszystko prócz KDE (jeżeli tylko w ten sposób użytkownicy mają się logować), nacisnąć Zastosuj.
- Żeby ustawić automatyczne logowanie dla KDE (dotyczy to tych dostarczanych z RH 7.x) w KDE Centrum Sterowania / System / Menedżer logowania / karta Ułatwienia / Logowanie automatyczne - ustawiamy opcję włącz logowanie automatyczne, Zaloguj po starcie i ustawiamy wybranego użytkownika (np. palacz).
- dla tych, którzy często używają program mc - aby poprawnie obsługiwał polskie litery należy ustawić opcję: 8-bit display. Bardzo przydatna jest też opcja: Definiuj klawisze.
- konfigurowanie locate; standardowo to się samo nie robi w systemie, tylko trzeba raz na początku odpalić ręcznie jako root:

```
/etc/cron.daily/slocate.cron
```

W RH 7.2 zwykle jest to dobrze skonfigurowane i nic nie trzeba robić, czasami trzeba odpalić **updatedb**.

- Polskie fonty i klawiaturę w trybie tekstowym można osiągnąć na kilka sposobów:

- W RH 7.2 wystarczy w pliku `/etc/sysconfig/keyboard` zamienić linijkę

```
KEYTABLE="us"
na
```

```
KEYTABLE="pl"
```

a następnie uruchomić (jako root) `/etc/init.d/keytable restart`.

- Wywołać komendę `anaconda --reconfig`. W tym przypadku (w przeciwieństwie do wersji instalacyjnej) będziemy mieli możliwość wyboru jęz. polskiego. (uruchamianie tego nie jest zalecane ze względu na pewne błędy i nadmierne rzeczy generowane przez anacondę)

- na końcu pliku `/etc/rc.d/rc.local` należy dodać linijki:

```
loadkeys /usr/lib/kbd/keymaps/i386/qwerty/pl.kmap.gz
consolechars -f /usr/lib/kbd/consolefonts/lat2u-16.psf.gz
```

- zmienić zawartość pliku `/etc/sysconfig/i18n` dopisując na jego końcu linijki:

```
LANG="pl_PL"
SYSFONT="lat2-sun16"
SYSFONTACM="iso02"
```

a w pliku `/etc/sysconfig/keyboard` zamiast linijki:

```
KEYTABLE="us"
```

wstawić linijki:

```
KEYTABLE=pl
KBCHARSET=iso-8859-2
```

aby zadziałał którykolwiek z dwóch wymienionych sposobów należy zresetować komputer

Uwaga! W RedHat 6.1 zamiast czcionki `lat2-sun16` należało użyć `lat2u-16`. Inną możliwością jest czcionka `iso02grf`.

- aby z poziomu Linuksa widoczne były dyski dosowe należy odpowiednio zmodyfikować (jako root) plik `/etc/fstab` wstawiając tam linijki:

```
/dev/hda1 /dos/c msdos defaults 0 0
/dev/hda5 /dos/d msdos defaults 0 0
/dev/hda6 /dos/e msdos defaults 0 0
/dev/hda7 /dos/f msdos defaults 0 0
/dev/hda8 /dos/g msdos defaults 0 0
/dev/hda4 /win98 vfat defaults 0 0
```

gdzie odpowiednie ścieżki do kolejnych dysków dosowych (nazwy w pierwszej kolumnie) można w przypadku dysku IDE odczytać poprzez wywołanie komendy:

```
/sbin/fdisk /dev/hda
```

w przypadku dysku SCSI:

```
/sbin/fdisk /dev/sda
```

w przypadku macierzy RAID:

```
/sbin/fdisk /dev/md0
```

- żeby nie trzeba było przy naprawie plików DBF korzystając z NU i CID wychodzić z Linuksa, lub robić to na innej maszynie trzeba zainstalować dobre `dosemu`. Trzeba ściągnąć `dosemu-1.0.0-1.i386.rpm` (lub nowszy), odinstalować dostarczane standardowo z RH 6.1 pakiety 0.99 wydając jako root komendy:

```
rpm -e xdosemu
rpm -e dosemu-freedos
rpm -e dosemu
```

zainstalować dosemu-1.0.0 wydając jako root komendę:

```
rpm -ihv dosemu-1.0.0-1.i386.rpm
```

w katalogu `/var/lib/dosemu` należy zrobić podkatalog `drives`, a w nim linki na dyski DOSowe widziane pod Linuksem (musi być odpowiednio zmodyfikowany `/etc/fstab`). Zakładając, że jest to katalog `/dos` i podkatalogi `c`, `d`... (jak w powyższym przykładzie ustawiania pliku `/etc/fstab`) w katalogu `/var/lib/dosemu/drives` trzeba zrobić jako root następujące soft linki:

```
for i in `ls /dos`
do ln -s /dos/$i $i
done
```

w pliku `/etc/dosemu.conf` zrobić wpisy: zamiast

```
$_hdimage="hdimage.first"
```

wstawić:

```
$_hdimage="drives/* commands"
```

zamiast:

```
$_rawkeyboard = (0)
```

wstawić:

```
$_rawkeyboard = (1)
```

zamiast:

```
$_term_char_set = ""
```

wstawić:

```
$_term_char_set = "ibm"
```

wtedy można korzystać z `Ctrl+Alt+Fx` do przełączania sesji. Katalog

`/var/lib/dosemu/commands` zawiera różne programy dosemu, m.in. `exitemu.com`. Ten katalog zostanie przyporządkowany w środowisku dosemu ostatniej (prócz ramdysku) literze dysku

DOSowego. Żeby mieć pełny dostęp znaków ASCII jak pod DOSem trzeba wcześniej wyłączyć polski font pod konsolą. Z dosemu należy korzystać jako root wywołując `dos` pod konsolą, a kończyć przez wywołanie:

```
<ostatni_dostepny_dysk_dosowy>:exitemu.com
```

tak skonfigurowane dosemu bierze `autoexec.bat` i `config.sys` z DOSa, które na pewno są nieodpowiednie dla środowiska dosemu, ale to bardzo nie przeszkadza. Można też sobie zrobić skonfigurowaną pod potrzeby dosemu sekcję w DOSowych `autoexec.bat` i `config.sys` pod tytułem np. `[dosemu]` i wybierać ją przy starcie dosemu. Inna możliwość to stworzenie odpowiednio zmodyfikowanych plików `autoexec.bat` i `config.sys` pod nazwami np. `autoexec.emu` i `config.emu` i zamiast:

```
$_emusys = ""
```

```
$_emubat = ""
```

wstawić:

```
$_emusys = "emu"
```

```
$_emubat = "emu"
```

- Niektóre nowe płyty główne, np. Intel D845BGL, nie obsługują standardu APM (Advanced Power Management), tylko ACPI (Advanced Configuration and Power Interface) opracowane przez Intela, Toshiba i Microsoft. Poniżej opisane są czynności konieczne do wykonania przy instalacji Red Hat v7.2 na maszynie z płytą główną Intel D845BGL, przy założeniu domyślnego kernela 2.4.7-10:

Konieczna jest rekonfiguracja kernela - w domyślnej włączone jest APM, a wyłączone ACPI: Y

```
cd /usr/src/linux-2.4
```

```
make menuconfig
```

Teraz należy odpowiednio włączyć i wyłączyć odpowiednie opcje:

```

General Setup:
[*] Power Management support
[*]   ACPI Support
< >   ACPI Debug Statements
<*>   ACPI Bus Manager
<*>   System
<M>   Processor
<*>   Button
< >   AC Adapter
< >   Embedded Controller
< >   Advanced Power Management BIOS support

```

Na tym etapie można też zrobić inne rzeczy, aby odchudzić kernela i przyspieszyć samą kompilację, wyrzucając niepotrzebne sekcje jak np. **SCSI**, **ISDN**, **RAID** itp. w zależności od sprzętowej konfiguracji maszyny. Po tym należy wybrać **Exit** i **Save configuration**. Teraz trzeba zrekompilować jądro:

```
make dep bzImage modules modules_install
```

Na etapie **modules\_install** może dojść do błędu **depmod**, ale nie koniecznie uniemożliwi to poprawne działanie tak przygotowanego kernela. Po kompilacji trzeba przygotować system do zmiany aktywnego kernela (zmiana `System.map` nie jest konieczna). W tym celu najpierw kopiujemy nowe jądro do katalogu `/boot`:

```

cp System.map /boot/System.map-2.4.7-10custom
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.7-10custom
cd /boot
ln -sf System.map-2.4.7-10custom System.map
mkinitrd /boot/initrd-2.4.7-10custom.img 2.4.7-10custom

```

a następnie konfigurujemy menadżer bootowania w taki sposób, żeby uruchamiał nowe jądro (tak jako to opisują Sekcja 3.2.5 i Sekcja 3.2.7). Warto na wszelki wypadek zostawić tymczasowo możliwość uruchamiania systemu ze starego jądra.

Teraz komputer należy zrestartować. Jeżeli instalacja karty graficznej była standardowa, wszystko powinno być gotowe. Jeżeli nie, należy ponownie zrobić moduły z pakietów źródłowych i ponownie zrestartować.

### 3.2.1.2. Instalacja systemów RedHat 9.x, Aurox 9.x i Fedora 1.x

Instalacja systemów RH 9.x/Aurox 9.x/Fedora różni się na tyle wyraźnie od instalacji starszych systemów, że wyodrębnienie tego podpunktu stało się koniecznością. Żeby jednak nie powtarzać informacji, tam, gdzie to tylko możliwe, znajdują się odniesienia do odpowiednich fragmentów z dokumentacji powyżej.

Wraz z wydaniem wersji 8.0, RedHat zabronił innym firmom wydawania i sprzedawania wersji systemu pod szyldem RedHat. Jest to jedna z przyczyn powstania „polskiego RedHata”, czyli Auroksa.

W roku 2003 RedHat podjął decyzję o wycofaniu się z segmentu rynku dla odbiorcy indywidualnego. Tradycję dystrybucji biurkowej „dla każdego” podtrzymuje jednak w ramach inicjatywy Fedora (<http://fedora.redhat.com/>), która jest rozwijana przez społeczność programistów (podobnie jak np. system Debian).

Na poziomie pakietów dystrybucje RH/Aurox (w odpowiednich wersjach) są niemalże identyczne. System Aurox jest oczywiście bardziej spolszczony: instalacja przebiega po polsku a wszystkie domyślne ustawienia są dobrane do języka polskiego i naszej strefy czasowej. Polskie komunikaty widzimy również w trakcie uruchamiania się systemu i startowania serwisów. Co więcej, w Auroksie

obecne są pakiety multimedialne, których w „zwykłym” RedHacie najbardziej brakuje: obsługa formatu mp3 oraz formatów video (program **mplayer**). Dodatkowo systemy RH z rodziny 8.x i 9.x charakteryzują się (domyślnie ustawionym) wyglądem interfejsu użytkownika niemalże identycznym w systemie Gnome jak i KDE (interfejs BlueCurve). W systemie RH 9.0 standardowo jest instalowany jest system CUPS, który w tandemie z programem **kudzu** instalują nam i konfigurują drukarkę (np. HP 815 - bezbłędnie). RH 9.0 wspiera również obsługę kółeczka w OEM-owych myszkach Logitech (bez żadnych patchy) i odczyt z płyt zapisanych w pakietowym formacie UDF. RH/Aurox (8.x jak i 9.x) posiadają całą masę narzędzi biurowych (m.in. OpenOffice). Teraz pliki Excela, Worda, PowerPointa można otwierać i modyfikować pod Linuksem bez większego problemu.

Sposób instalacji systemu w wersji 8.0 jest analogiczny do wcześniejszych wersji. W przypadku systemu RH 9.x (Aurox 9.x, Fedora) mamy nowość - na wzór instalacji systemu Windows, instalacja systemu została rozdzielona na 2 części:

- Pierwsza część to (tradycyjnie) program instalacyjny uruchamiany automatycznie z płytki:
  - Kilka początkowych kroków, czyli ENTER po tekście powitalnym, testowanie nośnika, wybór języka (w Auroksie zaproponowany będzie jęz. polski), ustawienia klawiatury i rodzaj myszy oraz typ instalacji („Custom”, czyli „Decyduj Sam”) i podział dysku na partycje, wykonujemy wg opisu ogólnego (por. wyżej). Niestety, w RH 9.0 (przynajmniej w wersji graficznej) nie mamy do dyspozycji programu **fdisk**. Dysk musimy podzielić na partycje za pomocą programu **Disk Druid**.
  - Jako boot loadera wybieramy GRUB-a w MBR (por. wyżej).
  - W kolejnym kroku ustawiamy sieć (por. wyżej). Musimy wybrać przycisk „Edycja” jeśli chcemy zmienić domyślne ustawienie DHCP. Dopiero wtedy uaktywnią nam się opcje „Bramka” i serwery DNS. W opcjach firewalla włączamy **ssh**.
  - Następnie wybieramy język polski (jako standardowy jęz. dla systemu), wybieramy strefę czasową warszawską (w Auroksie wszystko to jest domyślnie) oraz ustawiamy hasło roota, i potwierdzamy opcje bezpieczeństwa systemu: shadow i MD5 (por. wyżej).
  - W menu wyboru grup pakietów, przewijamy listę na dół i zaznaczamy opcję „Everything” (w Auroksie - „Wszystko”). Nie ma potrzeby wybierania każdej z grup z osobna. Uwaga! Wymagane około 5GB przestrzeni dyskowej.
  - Teraz zaczyna się instalacja systemu RH (trzy płyty) - zależności od systemu trwa od kilkudziesięciu minut do 2-3 godzin. Po zakończeniu instalowania pakietów zostaniemy poproszeni o stworzenie dyskietki startowej (nie chcemy) oraz wejdziemy do części konfiguracji interfejsu graficznego (X Window System). Wybieramy odpowiednią kartę graficzną (powinna zostać wykryta), monitor oraz parametry wyświetlania (głębina kolorów i rozdzielczość)

Po zainstalowaniu i ponownym uruchomieniu systemu przechodzimy do części drugiej instalacji. Jeżeli instalujemy korzystając z systemu Kickstart to tej drugiej części może nie być w ogóle.
- Przy pierwszym uruchomieniu systemu uaktywniany jest post-installer. To tutaj właśnie:
  - Dodajemy pierwszego zwykłego użytkownika (nie roota). Trzeba nadać użytkownikowi hasło. Inaczej nie przejdziemy do następnego kroku.
  - Ustawiamy datę i czas. Możemy skonfigurować system **ntpd** (to nowość!) i podać adres komputera, z którym się będziemy synchronizować.
  - Konfigurujemy (a właściwie akceptujemy zaproponowane) ustawienia karty dźwiękowej.
  - Rejestrujemy się (lub nie) w Red Hat Network. Rejestracja daje nam automatyczny upgrade pakietów, łatki itd. Jeśli nie chcemy się rejestrować (a innej opcji nie testowaliśmy ;-) to pozostanie

nam z spadku denerwująca ikonka (czerwony wykrzyknik) na dole ekranu, pokazujący brak połączenia z serwerami RH. Usuwamy ją następująco: klikamy prawym klawiszem na wykrzykniku i wybieramy opcję „Configuration”. Na ekranie „Red Hat Network Alert Notification Tool” klikamy „Dalej”. Na następnym ekranie („Terms of Service”) klikamy na klawisz „Remove from Panel”.

- Instalujemy pakiety z dodatkowych CD-Romów RH (jeśli chcemy, jeśli mamy): „RH Linux Documentation CD”, „RH Linux Installation CD”, „Additional CDs”.

Przy pierwszym logowaniu się należy upewnić się, że uruchamiamy KDE. Wprawdzie Bluecurve powoduje, że oba systemy wyglądają tak samo, ale wcale nie zachowują się tak samo (skrótów klawiszowe, ustawienia). Wybierając KDE zostaniemy zapytani, czy chcemy danego użytkownika skojarzyć z sesją KDE na przyszłość. Odpowiadamy, że tak.

RH 9.0 ma domyślnie ustawione kodowanie znaków UTF-8 (unicode). Poprawiamy to ustawiając w pliku `/etc/sysconfig/i18n` zmienną `LANG` na „`pl_PL`” a zmienną `SYSFONT` na „`lat2-16`”.

Uwaga! Program Raporter z dystrybucji SZARP źle współpracuje z Tcl 8.3 obecnym w systemie Aurox 9.x (9.0, 9.1, 9.2) - pożera ok. 60MB pamięci RAM. Nie jest to związane z konkretną wersją Raportera. Na RH7.2 i 7.3, gdzie jest TclTk 8.3.3-x, nie ma tego problemu, podobnie na Debianie z TclTk 8.0, 8.2, 8.3, 8.4. Rozwiązanie problemu:

- Usunąć TclTk (parę pakietów od niego zależy - dlatego należy zastosować opcję `--nodeps`):

```
rpm -e tk --nodeps
rpm -e tcl --nodeps
```

- Zainstalować nowe rpm-y TclTk 8.4 (przetestowane, prawdopodobnie nowsze wersje też będą działać poprawnie), np.

```
rpm -i tcl-8.4.4-1.i386.rpm
rpm -i tk-8.4.4-1.i386.rpm
```

- Zrobić odpowiednie linki:

```
ln -s tclsh8.4 /usr/bin/tclsh
ln -s wish8.4 /usr/bin/wish
```

Uwaga! W systemie Aurox 9.2 (dystrybucja czteropłytkowa), przeinstalowane pakiety systemu wxWindows (`wxGTK-2.4.1-9.2`) mają błąd, który powoduje, że nie mogą być uruchamiane aplikacje w programie SCC. Należy więc zrobić upgrade systemu wx do wersji 2.4.2. Podejrzenie zachowuje się również program **rpm** - zdarza mu się zawiesić się przy instalowaniu lub odinstalowywaniu pakietów. Bardzo ożywczo działa na niego reboot systemu.

### 3.2.1.3. Automatyzacja instalacji RedHata: **KickStart**

**Notatka:** Niniejszy opis tworzony/uaktualniany był na dystrybucjach RedHat 9.0 i wcześniejszych. Autorzy nie są w stanie nic powiedzieć o jego przydatności przy nowszych wersjach RedHat'a/Fedory.

RedHat KickStart (KS) daje możliwość zainstalowania RedHata według pewnego schematu. Może to być przydatne w sytuacjach, gdy instalowanych jest wiele, identycznie skonfigurowanych hostów; dzięki RH KS można również (przynajmniej teoretycznie) zlecić instalowanie RH laikowi, bo wybór elementów do instalacji oraz konfiguracja systemu zostaną przeprowadzone automatycznie (*laik* będzie wówczas odpowiedzialny za wymianę płytek instalacyjnych i naciskanie Entera). KickStart pozwala na:

- Wybór pożądaných pakietów (ich dołożenie/usunięcie w stosunku do oficjalnej dystrybucji)
- Zainstalowanie wybranych pakietów przez sieć (z serwera FTP lub HTTP)
- Batchową konfiguracją wielu elementów systemu
- Łatwość i szybkość konfiguracji samego KS
- Przeniesienie konfiguracji KS na inną wersję RH niż wersja, na/dla której utworzono pierwotną konfigurację KS.

Dokumentację do systemu RedHat KickStart można znaleźć na płycie RedHata „Dokumentacja”.

Ponadto w RH 7.3 jest standardowo instalowany plik

`/usr/share/doc/anaconda-7.3/kickstart-docs.txt`, zawierający pewne informacje dotyczące KS-a (KS jest elementem Anacondy - programu instalacyjnego dla RH). Niniejsze streszczenie powstało na podstawie tej dokumentacji oraz na podstawie własnych doświadczeń z KS na systemach RH 6.2 i RH 7.3.

Żeby uaktywnić KickStarta w czasie instalacji RH, należy przekazać do jądra parametr `ks=<lokalizacja pliku konfiguracyjnego KS (ks.cfg)>`<sup>1</sup>. Plik `ks.cfg` może się znajdować na dyskietce (`ks=floppy`), jak również w sieci (`ks=nfs:` lub `ks=http:`), może być reprezentowany przez urządzenie w katalogu `/dev/` (`ks=hd:`) albo może być umieszczony na płycie (`ks=cdrom:`). I nie musi się nazywać `ks.cfg`, lecz np. `praterm-ks.cfg`. Wówczas parametr „ks=” musi zawierać nazwę pliku, np. „ks=hd:fd0/praterm-ks.cfg”.

Z KickStarta najłatwiej skorzystać tworząc odpowiednią dyskietkę startową RH, np. przy pomocy komendy **dd** (załóżmy, że w `/mnt/cdrom` jest zamontowana płyta z oryginalnym RedHatem):

```
dd if=/mnt/cdrom/images/boot.img of=/dev/fd0,
```

Kilka słów należy się dyskietce instalacyjnej RedHata. Do uruchomienia i sparametryzowania procesu instalacji RH używa narzędzia zwanego SysLinux. Prosty boot manager pozwala użytkownikowi wybrać jedną z możliwości instalacji; różnice są wprowadzane na poziomie parametrów przekazywanych do jądra. Dyskietka ma DOS-owy system plików, łatwo jest więc zmodyfikować owe parametry lub dodać swoje własne rozszerzenia. Plikiem konfiguracyjnym dla SysLinuksa jest `syslinux.cfg`, którego zawartość do złudzenia przypomina format pliku `lilo.conf`, nie będzie więc tu opisywana. Żeby uruchamiany był KS, należy dodać linię `ks=floppy` do parametrów jądra w danym trybie, po czym przekopiować przygotowany wcześniej plik `ks.cfg` na dyskietkę. Instalacja KickStart uruchomi się wówczas automatycznie. Dodatkowo, możliwe jest, podobnie jak w przypadku oryginalnej instalacji RH, wprowadzenie różnych opcji instalacji, np. „Instalacja minimalna (wymagane 900MB miejsca na dysku)”, „Instalacja pełna (2.5GB)” albo „Instalacja sieciowa”. W tym celu modyfikujemy odpowiednio plik `syslinux.cfg` oraz pliki zawierające wyświetlane informacje, np. `boot.msg`, `minimalna.msg`, `pelna.msg`, `sieciowa.msg`. Niezbędne stają się też stworzenie kilku plików konfiguracyjnych dla KickStarta, i ustawianie innej opcji „ks=” w każdym z trybów, np. „ks=hd:fd0/ks-minimalna.cfg”, „ks=hd:fd0/ks-pelna.cfg” itd.

Obraz dyskietki instalacyjnej znajduje się na CDROM-ach instalacyjnych RedHata i jest używany w procesie instalacji. Zmodyfikowanej wersji dyskietki można zatem użyć przy tworzeniu własnej dystrybucji RH.

Format pliku `ks.cfg`

Plik `ks.cfg` zawiera odpowiedzi na pytania (niekoniecznie wszystkie), które są normalnie zadawane w procesie instalacji. Jeżeli pewnej odpowiedzi nie ma w pliku, użytkownik zostanie o to zapytany w

trakcie instalacji. Ponieważ pewne ustawienia są stosowane zawsze przy instalacji RH dla potrzeb SZARP-a, np. wybór języka, strefy czasowej, firewalla, hasło dla roota, bootloader, i inne (por. Sekcja 3.2.1), `ks.cfg` z tym przypadkiem będzie zawierał odpowiedzi na te pytania. Na inne (np. numer IP hosta) użytkownik będzie musiał odpowiedzieć samemu podczas instalacji.

Format pliku `ks.cfg` różni się pomiędzy wersjami 6.x i 7.x RedHata. Różnice są wprawdzie niewielkie i duży podzbiór komend jest wspólny dla obu wersji, ale należy być świadomym tego przy korzystaniu z istniejących plików `ks.cfg` oraz dokumentacji do systemu KS. Należy również pamiętać o tym, że w RH6.2 komendy KS miały znacznie mniej parametrów (np. konfigurując X Window System w KS z RH6.2 nie można było ustawić rozdzielczości, podczas gdy w RH7.3 można ustawić nie tylko rozdzielczość ale i głębię kolorów).

Automatyczną konfigurację w `ks.cfg` dla RH 7.3 niektórych z opcji przedstawiono poniżej; ustawiany jest tu język instalacji (angielski - `en_US`), wsparcie i ustawienie jako domyślnego języka polskiego, ustawienia klawiatury, myszy, hasła „na roota”, opcji firewalla (otwarcie 3-ch portów), włączenie shadowingu dla haseł, ustawienie strefy czasowej oraz żądanie zainstalowania bootloadera (GRUB-a):

```
install
lang en_US
langsupport --default pl_PL en_US.iso885915 pl_PL
keyboard us
mouse generic3ps/2 --device psaux
rootpw --iscrypted $1$Ë0ÑÜIÅa$hmRXwbpzFytxg0bU/0TML/
firewall --medium --port smtp:tcp --port http:tcp --port ssh:tcp
authconfig --enablesshadow --enablemd5
timezone --utc Europe/Warsaw
bootloader
```

Jeżeli konfigurujemy np. 20 hostów z identyczną kartą graficzną (i z podobnym monitorem) to możemy automatycznie skonfigurować też ikisy, jak poniżej:

```
xconfig --card "RIVA128" --videoram 4096 --hsync 31.5-48.5 --vsync 50-70 --resolution
```

Pominięcie symbolu karty graficznej (lub/i rozmiaru RAM-u na karcie) spowoduje, że system spróbuje automatycznie ją wykryć i, jeśli mu się powiedzie, użytkownik nie zostanie zapytany o kartę.

```
xconfig --hsync 31.5-48.5 --vsync 50-70 --resolution 1024x768 --depth 16 --startxonboo
```

Możliwe jest też automatyczne partycjonowanie dysku. Poniżej pokazano partycjonowanie dysku wg następującej zasady:

- 200MB dla dosa
- RAMx2 (wartość rekomendowana) dla swapa
- reszta - / dla Linuksa

Przed dodaniem jakiegokolwiek partycji wyrzucone będą z dysku wszystkie istniejące dotychczas partycje (`clearpart --all`):

```
clearpart --all
part /mnt/dos --fstype vfat --size 200
part swap --recommended
```

```
part / --fstype ext3 --size 10000 --grow
```

KickStart pozwala również określić pakiety oraz moduły (zbiory pakietów, poprzedzone znakiem „@”) do zainstalowania, np.:

```
%packages
@ Printing Support
@ Classic X Window System
@ X Window System
@ KDE
@ Network Support
@ Router / Firewall
@ Authoring and Publishing
@ Emacs
@ Utilities
@ Software Development
@ Kernel Development
szarp
codebase
codebase-devel
```

lub po prostu:

```
%packages
@ Everything
```

Możliwe jest także umieszczenie własnych skryptów i komend po zainstalowaniu wszystkiego (np. w celu stworzenia katalogów, modyfikacji konfiguracji):

```
%post
echo "nameserver 192.168.1.1" > /etc/resolv.conf
mkdir /usr/work
```

W pliku konfiguracyjnym KS możliwa jest też konfiguracja sieci oraz przeprowadzenie całej instalacji przez sieć. Wówczas instalacja RH wyglądać może tak:

- Wkładam dyskietkę (ze skonfigurowanym KS-em) do stacji.
- Uruchamiam komputer tak, żeby wystartował z dyskietki.
- Gdy już instalacja wystartuje zwykle mamy możliwość wyboru źródła pakietów do zainstalowania (lokalny CD-ROM, dysk twardy, NFS, FTP, HTTP). W przypadku opisywanym tu wybieramy opcję: lokalny CD-ROM.

Najlepszym rozwiązaniem (minimalizacja ruchu w sieci a zarazem dostęp do najnowszych wersji pakietów) jest instalacja hybrydowa, np. całość systemu instalowana jest z płytek, natomiast reszta (najnowsze biblioteki motifa, SZARP, niestandardowe pakiety) mogą być zainstalowane przez sieć. Jest to możliwe m.in. dzięki opcjom programu **rpm**, którego wywołanie można umieścić w sekcji %post w pliku konfiguracyjnym KS w formie jak niżej:

```
rpm -i ftp://user:password@host:/path/packet.rpm
```

Po każdej instalacji RedHata w katalogu `/root` generowany jest plik `anaconda-ks.cfg` z konfiguracją, wg której przebiegała instalacja. Plik ten może posłużyć za wzorzec do tworzenia własnych plików konfiguracyjnych `ks.cfg`.

Przykładowy plik konfiguracyjny `ks.cfg` dla RH6.2 przedstawiono poniżej:

```
install
lang en_US
keyboard us
mouse generic3ps/2 --device psaux
auth --useshadow --enablemd5
rootpw --iscrypted $1$E0ÅÑÜIÅa$hmRXwbpzFytxg0bU/0TML/
timezone --utc Europe/Warsaw
lilo --location mbr
# W nie ma polecenia bootloader, jest tylko lilo
xconfig --hsync 31.5-48.5 --vsync 50-70 --startxonboot
# Nie ma możliwości ustawienia typu i systemu plików dla partycji
clearpart --all
part /mnt/dos --size 200
part swap --size 200
part / --size 1000 --grow
network --bootproto static --ip 192.168.1.111 --netmask 255.255.255.0

%packages
# Nie ma opcji @ Everything
@ Printer Support
@ X Window System
@ Mail/WWW/News Tools
@ DOS/Windows Connectivity
@ Utilities
@ Graphics Manipulation
@ Multimedia Support
@ Networked Workstation
@ Dialup Workstation
@ Authoring/Publishing
@ Emacs
@ Development
@ Kernel Development
@ Workstation Common
@ GNOME
@ KDE
@ Network Server
@ News Server
@ NFS Server
@ SMB (Samba) Server
@ Anonymous FTP Server
@ Web Server
@ DNS Name Server
@ Postgres (SQL) Server
@ Network Management Workstation
%post
mkdir /usr/work
/usr/sbin/useradd palacz
chfn -f " palacz
/usr/bin/passwd -d palacz
```

Uwaga! Plików `ks.cfg` nie powinno się pisać ręcznie, należy raczej wykorzystać istniejący już plik `anaconda-ks.cfg` lub RedHatowe narzędzie graficzne pod nazwą **ksconfig**, które służy właśnie do konfiguracji i generowania plików `ks.cfg`.

#### 3.2.1.4. Rekonfiguracja RedHat'a - **Anaconda**

RedHat Linux w trakcie instalacji używa programu o nazwie **anaconda** do skonfigurowania systemu, ustalenia zależności między pakietami i grupami pakietów rpm oraz do zręcznego poprowadzenia użytkownika przez trudy instalacji. Żeby uruchomić anacondę już na zainstalowanym systemie należy wpisać w shellu

```
anaconda --reconfig
```

W ten sposób możemy powtórnie ustawić rzeczy takie jak: klawiatura, język, strefa czasowa i ustawienia zegara systemowego, ustawienia sieci, hasło roota, parametry autoryzacji (shadow, MD5) oraz NIS. Niestety nie da się w ten sposób ustawić fontu dla konsoli.

UWAGA! Dla bezpieczeństwa lepiej uruchamiać anakondę w 3-cim runlevelu (bez X-ów). I w ogóle dla bezpieczeństwa lepiej tego programu nie uruchamiać w ogóle.

### 3.2.2. Instalacja systemu Debian GNU/Linux

Opis ten dotyczy instalacji systemu Debian GNU/Linux. W sieci można spotkać dużo innych opisów, np. Debian GNU/Linux. Desktop Survival Guide (<http://www.togaware.com/linux/survivor>) autorstwa Grahama Wiliamsa, które są bardziej wyczerpujące i kompletne.

Instrukcja z niniejszego rozdziału dotyczy specyficznej dystrybucji systemu Debian, przygotowanej przez firmę Praterm. Nasza dystrybucja mieści się na jednej płycie CD, jako bazę wykorzystujemy najnowsze dystrybucje oficjalne systemu. Przebieg instalacji jest podobny do przebiegu instalacji zwykłego systemu Debian (główną różnicą jest ilość pakietów do zainstalowania i skonfigurowania).

Żeby zainstalować system Debian z płyty instalacyjnej, komputer musi wspierać uruchamianie systemu operacyjnego poprzez urządzenie CD (tzw. bootable CD-ROM). Co robimy, krok po kroku:

- Wkładamy płytkę instalacyjną do napędu CD-ROM i uruchamiamy komputer.
- Na ekranie powitalnym „Welcome to Debian GNU/Linux” naciskamy ENTER. Uruchomiony zostanie wówczas program instalacyjny Debiana (graficzny instalator). Jeśli z jakiegoś powodu nie uda nam się zainstalować przy pomocy tego instalatora możemy użyć starego poprzez wpisanie na ekranie powitalnym "install".

Program instalacyjny charakteryzuje się m.in. możliwością powrotu do dowolnego poprzedniego kroku instalacji bez przechodzenia przez kroki pośrednie: na przykład można powrócić do części instalowania jądra i sterowników, następnie dodania potrzebnego sterownika (powiedzmy, karty sieciowej) i kontynuowania instalacji od momentu, w którym została ona przerwana.

- Pierwszym etapem jest skonfigurowanie sieci. Jeśli nie mamy w sieci DHCP, to powinniśmy wcześniej przygotować sobie te ustawienia np. poprzez kontakt z osobą kompetentną w tej sprawie. Jeśli mimo to nie znamy ustawień, możemy jako adres wpisać np. 192.168.1.181, maska 255.255.255.0, reszta obojętna, możemy naciskać enter uznając domyślne ustawienia za poprawne.

- Następnie instalator zapyta nas o nazwę komputera. Może to być np. debian, praca, dom itp. Po tym pytaniu zostaniemy zapytani o nazwę domeny, jeśli nie znamy naciskamy enter.
- Następny etapem to podział dysku na partycje (okno: „Partycjonuj Dyski”).

Najpierw wybieramy sposób partycjonowania. Mamy tu dwie możliwości:

- „Wyczyść cały dysk” - ta opcja usunie z dysku wszystkie partycje a następnie zaproponuje podział na partycje wg. jednego z kilku planów („Wszystkie pliki na jednej partycji”, „Osobna partycja na katalogi domowe”, „System wieloużytkownikowy”). Tę opcję możemy wybrać w sytuacji, gdy instalujemy system na nowym dysku (nie będzie trzeba wówczas dzielić ręcznie dysku na partycje) lub gdy chcemy skasować wszystkie istniejące systemy operacyjne na dysku. Doświadczeni użytkownicy z tej opcji raczej nie korzystają, lecz wybierają (opisaną w kolejnym punkcie) opcję „Modyfikuj ręcznie tablicę partycji”.

Opcja „Wyczyść cały dysk” jest dobra natomiast dla użytkowników niedoświadczonych w partycjonowaniu dysków. Ponadto, jeśli nie mamy doświadczenia w dzieleniu dysku na partycje (albo trudno nam oszacować przyszłe potrzeby dyskowe), to powinniśmy wybrać możliwość „Wszystkie pliki na jednej partycji”. Na dysku powstanie wówczas jedna duża partycja do zamontowania jako „/” (ang. *root partition*) oraz druga, mniejsza, dla przestrzeni wymiany (ang. *swap space*). Ten podział potwierdzamy wybierając opcję „Zakończ partycjonowanie i zapisz zmiany na dysku”.

- „Modyfikuj ręcznie tablicę partycji” - wybieramy jeśli chcemy ustawić partycje ręcznie (opcja zalecana). Po uruchomieniu tej opcji wyświetli się nam okno z wieloma interesującymi możliwościami (m.in. konfiguracją RAID-a programowego, konfiguracją menadżera wolumenów logicznych, partycjonowanie z przewodnikiem). Z większości opcji będziemy korzystać tylko w szczególnych okolicznościach (np. gdy odpowiednio: chcemy zainstalować system na macierzy RAID; skorzystać z partycji „wirtualnych”, tj. stworzyć/zmodyfikować partycje „wyższego rzędu” zbudowane ze „zwykłych” partycji; dostać więcej podpowiedzi od systemu).

W najczęstszym przypadku powinniśmy przesunąć kursor do listy partycji (rozpoczynającej się od linii oznaczającej cały dysk), w której wyświetlone są również fragmenty wolnej przestrzeni na dysku, i wybrać jedną z możliwości.

Zwykle musimy stworzyć dwie partycje: jedną dużą partycję dla głównego systemu plików i jedną partycję o wielkości co najmniej 2xRAM (512 MB jest OK) na przestrzeń wymiany (ang. *swap space*). Natomiast partycja dla systemu plików powinna mieć co najmniej 5GB. (Uwaga dla zaawansowanych użytkowników. Możemy stworzyć wiele partycji dla Linuksa o rozmiarach co najmniej: / - 2GB /home - 2 GB /opt - 3 GB).

Jeżeli wybraliśmy partycję już stworzoną, to przechodzimy do okna ustawiania partycji (opisanego 2 akapity dalej).

Jeżeli natomiast wybraliśmy wolną przestrzeń na dysku, system zapyta nas czy chcemy utworzyć nową partycję, czy pozwolić systemowi na automatyczne partycjonowanie przestrzeni. Wybieramy opcję „Utwórz nową partycję”. Będziemy mogli tu ustawić wielkość partycji i jej rodzaj (Podstawowa/Logiczna). Następnie zostanie wyświetlone okienko z możliwością ustawienia partycji (poniżej).

Okno ustawiania partycji. Mamy tu wyboru kilka opcji (uwaga! nie wszystkie opcje są widoczne od początku; kolejne opcje będą się pojawiać w miarę uzupełniania danych w kolejnych wierszach):

- „Sposób użycia”, a w nim: formatowanie partycji (dla nowych partycji jest to czynność domyślna), użycie partycji jako przestrzeni wymiany (czyli zamiana partycji na tzw. *swap disk*), nie używanie partycji (jeżeli mamy tam jakieś dane, których nie chcemy stracić) i inne. Jeżeli chcemy na partycji zainstalować system, wybieramy opcję „Formatuj Partycję”.

- „System plików” - tu wybieramy system plików dla partycji (do wyboru mamy ext2, ext3, ReiserFS, JFS, XFS, FAT16 i FAT32). Zalecane jest wybranie systemu z księgowaniem: ReiserFS.
- „Punkt montowania” - czyli miejsce, gdzie w drzewie katalogów pojawi się dana partycja (gdy mamy tylko jedną partycję to musi być katalog główny, czyli „/”).
- „Opcje montowania” - dodatkowe opcje (nie zmieniamy - „defaults” jest ok).
- „Flaga uruchomieniowa” - przeważnie nie zmieniamy. Uwaga! Na nowszych (jest koniec 2007 roku) płytach Intel'a trzeba ustawić flagę uruchomieniową przynajmniej dla partycji z katalogiem głównym „/”. W innym przypadku po restarcie system nie wstanie i pojawi się komunikat, że system nie ma skąd się uruchomić. Można wtedy (np. po odpaleniu systemu z płyty) narzędziem *cfdisk* zmienić flagę uruchomieniową partycji. Innym wyjściem jest instalacja wszystkiego od nowa.
- „Rozmiar” - nie zmieniamy.

Po ustawieniu właściwości danej partycji zatwierdzamy zmiany wybierając opcję „Zakończono ustawianie partycji”. Czynność powtarzamy dla każdej partycji, którą chcemy dostosować do własnym potrzeb (najczęściej wystarczy jednak tylko jedna partycja).

Jeżeli zakończyliśmy modyfikację partycji, wybieramy opcję „Zakończ partycjonowanie i zapisz zmiany na dysku”.

Jeżeli dokonywaliśmy jakiś zmian w ustawieniach partycji, konfigurator dopyta nas czy na pewno chcemy je wprowadzić (potwierdzamy). W ten sposób zakończy się operacja partycjonowania dysku.

- (teraz system automatycznie przepartycjonuje dysk wg. wskazówek od użytkownika udzielonych w poprzednim kroku i przekopiuje podstawowe pliki na dysk)
- Opcjonalnie Instalator zapyta nas, czy zegar komputera ma być ustawiony na UTC. Jeśli na komputerze jest tylko Linux, powinniśmy wybrać odpowiedź „Tak (Yes)”. Jeśli mamy zainstalowany jakiś inny system, np. Windows, wybieramy „Nie (No)”.
- W oknie („Ustaw użytkowników i hasła”) ustawiamy hasło dla administratora systemu (root).
- W kolejnym oknie zostaniemy poproszeni wpisanie pełnej nazwy użytkownika, następnie jego nazwy w systemie (tzw. *login*) oraz hasła dla tego użytkownika (nazwa dowolna).
- Podczas instalacji systemu podstawowego zostaniemy zapytani o docelowe jądra w systemie. Z listy wybieramy jądro odpowiednie dla naszego procesora (intel - 686, amd - k6) np. linux-image-2.6.16-1-686.
- W trakcie instalacji będziemy musieli wybrać jądro. Jeśli mamy intel'a, wybierzmy 686, jeśli amd k7, w innym przypadku sugerowany jest wybór 486.
- Po instalacji systemu podstawowego instalator poinformuje nas że nie mógł połączyć się z serwerem lustrzanym. Wybieramy „Ignoruj” i naciskamy enter. Instalator zatrzyma się na dłuższą chwilę na przeszukiwaniu repozytorium z poprawkami bezpieczeństwa i poinformuje nas że nie może uzyskać do niego dostępu. Naciskamy dalej.
- W oknie „Wybór oprogramowania” ODZNACZAMY wszystkie opcje, tak by nic nie było instalowane. Wszystkim zajmie się później instalator przygotowany przez pracowników Praterm S.A.
- W oknie „Instaluj program uruchomieniowy GRUB na dysku twardym” potwierdzamy, że chcemy zainstalować GRUB-a w głównym rekordzie uruchomieniowym. (System Debian może poinformować nas również o wykrytych istniejących systemach operacyjnych, które zostaną dodane do menu GRUB-a).

- Po zainstalowaniu programu GRUB, program instalacyjny wyświetli okno „Zakończ instalację”, (potwierdzamy) i uruchomi system z nowo zainstalowanej dystrybucji Debiana.

*Uwaga! Należy pamiętać o wyciągnięciu płyty instalacyjnej: system powinien się uruchomić z dysku nie z płyty!*

- Pierwszym elementem uruchamianym w nowo instalowanym systemie jest GRUB. Ponieważ, jak wspomniano, w systemie Debian GRUB stara się automatycznie wykryć inne systemy operacyjne (dodając je do listy), w menu GRUB-a powinniśmy zobaczyć wszystkie lub większość systemów zainstalowanych na komputerze. System Debian, który instalujemy aktualnie, będzie umieszczony na pierwszej pozycji (wybór domyślny).
- Po uruchomieniu się nowego systemu, logujemy się jako root i wydajemy polecenie:
 

```
deb4szarp-installer
```
- (opcjonalnie) Pojawi się prośba o włożenie naszej płyty instalacyjnej do napędu cdrom.
- Może pojawić się okno konfiguracji pakietu locales. Nasz wybór to pl\_PL.
- W oknie „Wybierz profil” mamy następujący wybór:
  - **Standardowy** - instaluje wszystkie pakiety potrzebne do kompilacji SZARP'a, oraz X'y i KDE.
  - **Minimalny** - instaluje tylko pakiety potrzebne do kompilacji SZARP'a, nie ma X'ów, ani nic zbędnego.
  - **Pełny** - instaluje to, co w profilu Standardowym + inne przydatne rzeczy nie tylko na ciepłowni, np. Thunderbird, OpenOffice, mtr, itd...
- W oknie „Dodatki” wybieramy elementy mogące się przydać w danym przypadku:
  - **cups** - jeśli będziemy chcieli korzystać z drukarek,
  - **isdn** - w przypadku używania ISDN-a do łączenia się z Internetem,
  - **samba** - jeśli chcielibyśmy się dzielić zasobami dyskowymi z komputerami z systemem Windows,
  - **apt-proxy** - jeśli chcielibyśmy zainstalować proxy pakietów Debian'a,
  - **edge-gprs** - jeśli będziemy używać modemów Sony Ericsson GC 75/75e/85/89 do komunikacji ze światem zewnętrznym,
  - **openoffice** - jeśli chcielibyśmy zainstalować pakiet biurowy OpenOffice.org 2.0.
- W oknie „Konfigurowanie Exim v4” wybieramy „ośrodek internetowy” jako typ konfiguracji poczty. Jako odbiorcę poczty ustawiamy użytkownika **root**.
- Gdy zostaniemy zapytani o rozdzielczość naszego monitora, powinniśmy zaznaczyć optymalną dla posiadanego przez nas urządzenia (np. 1024x768).
- Gdy zostaniemy zapytani o mysz i port do którego jest podłączona możemy wybrać /dev/input/mice.
- (Uwaga! To pytanie pojawia się tylko wtedy, gdy w poprzednim punkcie wybraliśmy element „cups”). W oknie „Foomatic Printer Filter Configuration” wybieramy opcję „cups”.
- (opcjonalnie) To pytanie pojawia się tylko wtedy, gdy wybraliśmy wcześniej opcję ISDN. Na pytanie „Does firmware need to be loaded?” odpowiadamy „none”.
- (opcjonalnie) Poniższe pytania dotyczą konfiguracji ippd i pojawiają się tylko wtedy, gdy wybraliśmy wcześniej opcję ISDN. Na pytanie „What is your ISP phone number?” zostawiamy opcję „manual”. Następnie w pytaniu o kraj wypieramy opcję „other” (ponieważ Polski nie ma na liście). W kolejnym

oknie wpisujemy skrót dla Polski („PL”). Pytanie „What is your area code?” pozostawiamy bez odpowiedzi. Na kolejne pytanie odpowiadamy domyślnie „quit”.

- (W kolejnym oknie potwierdzamy przyjęcie do wiadomości, że libsensors3 nie jest aktualnie w pełni funkcjonalne).
- (opcjonalne) Jeżeli wybraliśmy w jednym z poprzednich kroków instalację Samby, to zostaniemy zapytani o kilka rzeczy:
  - System zapyta nas o nazwę grupy roboczej lub domeny. Wpisujemy i potwierdzamy (może być nazwa pusta).
  - Na pytanie: „Use password encryption?” odpowiadamy „Yes”.
  - Na kolejne pytanie: „Modify smb.conf to use WINS settings from DHCP?” odpowiadamy „No”.
  - Na pytanie „How do you want to run Samba?” wybieramy opcję „daemons”.
  - Na pytanie „Create samba password database, /var/lib/samba/passdb.tdb?” odpowiadamy „No”.

*(Uwaga! Powyższe ustawienia możemy zmodyfikować w zależności od potrzeb - np. szyfrować bez hasła jeśli jest taka potrzeba).*

- (opcjonalne) Jeżeli w jednym z poprzednich kroków wybraliśmy instalację systemu cups, to teraz zostaniemy zapytani o to, czy zainstalować interfejsy zgodności z drukowaniem BSD („BSD lpd compatibility server”). Odpowiadamy przecząco.
- Teraz powinna się rozpocząć instalacja wszystkich skonfigurowanych pakietów.
 

*Uwaga! Ponieważ w niektórych przypadkach robione jest przeinstalowanie istniejących pakietów, system może wykryć pewne konflikty i zaproponować przerwanie instalacji („Do you want to stop now?”). Odpowiadamy „N”, bo nie chcemy przerywać instalacji.*
- *Uwaga(2)! Dla niektórych pakietów nie można sprawdzić ich podpisów cyfrowych (ostrzeżenie „(...) cannot be securely authenticated”). Pojawia się wówczas propozycja przerwania instalacji. Nie należy tego robić - na pytanie „Continue?” odpowiadamy: „Y”.*

W ten sposób Debian został zainstalowany.

*Uwaga! Po zainstalowaniu Debiana należy uruchomić powtórnie system. Po uruchomieniu system będzie działał na nowym jądrze.*

### 3.2.2.1. Dodatki i problemy podczas instalacji

Poniżej przedstawiono garść informacji dotyczących problemów, które mogą się pojawić po zainstalowaniu systemu.

#### 3.2.2.1.1. Konfiguracja X-Windows

Może się zdarzyć, że w czasie instalacji źle skonfigurowaliśmy X-Windows (np. wybraliśmy opcję automatycznego wykrywania środowiska, która się programowi instalującemu nie udała) i ikсы nie wstają po uruchomieniu systemu. Należy się wówczas zalogować w trybie tekstowym i wydać z konsoli komendę:

```
dpkg-reconfigure xserver-xorg
```

Uruchomiony zostanie konsolowy program konfiguracyjny, który pozwoli nam na wprowadzenie nowych ustawień.

### 3.2.2.1.2. Konfiguracja sieci

W Debianie nie ma jednego spójnego narzędzia do zarządzania systemem. Istnieje za to wybór narzędzi do konfiguracji poszczególnych elementów systemu (np. narzędzie **etherconf** do konfiguracji sieci lokalnej).

Debiana natomiast jest stosunkowo łatwo konfigurować ręcznie: ustawienia sieci są w pliku `/etc/network/interfaces`. Plik ze skonfigurowanymi urządzeniami `lo` i `eth0` ma następującą postać:

```
auto lo eth0

iface lo inet loopback

iface eth0 inet static
    address 192.168.1.166
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

Pierwsza linia mówi które interfejsy są uruchamiane automatycznie przy starcie systemu. Opis każdego interfejsu rozpoczyna się od słowa **iface**. Komenda, co widać powyżej, może przyjmować różne parametry w zależności od rodzaju interfejsu.

Po wprowadzeniu poprawki w pliku `interfaces` należy uruchomić ponownie interfejsy sieciowe:

```
/etc/init.d/networking restart
```

..i sprawdzić czy interfejsy zostały zainicjowane:

```
ifconfig
```

Należy również skonfigurować listę serwerów DNS. Podobnie jak w innych Linuksach, serwery DNS są wypisane w pliku `/etc/resolv.conf`. Plik składa się z kolejnych wierszy z adresami DNS-ów, przykładowy plik `/etc/resolv.conf` wygląda zatem następująco:

```
nameserver 148.81.16.51
nameserver 195.187.245.51
nameserver 194.204.159.1
nameserver 194.204.152.34
```

Jeszcze innym plikiem jest `/etc/hosts`, który zawiera listę statycznych numerów IP i ich nazw (albo tzw. aliasów), np.:

```
192.168.1.1 server
192.168.1.10 terminal1
192.168.1.11 terminal2
192.168.1.12 terminal3 kierownik
```

Dla każdego numeru IP można wpisać więcej niż jedną nazwę hosta: do hosta 192.168.1.12 można się odwoływać równie dobrze za pomocą nazwy **terminal3** jak i **kierownik**.

### 3.2.2.1.3. Instalowanie/upgrade pakietów

W systemie Debian istnieje wygodne narzędzie do zarządzania pakietami: **apt-get**. Potrafi automatycznie zlokalizować niezbędne pakiety w sieci (lub na CD-ROM-ie), wykryć zależności pomiędzy paczkami i poprawnie zainstalować żądany pakiet. Żeby zainstalować pakiet <pakiet>, należy wywołać program następująco:

```
apt-get install <pakiet>
```

Program wyszuka w lokalnej bazie danych indeks paczki a następnie sprawdzi co jest potrzebne do poprawnego jej zainstalowania. Przed zainstalowaniem poinformuje o wielkości ściąganych z sieci danych i poprosi nas o potwierdzenie. Po ściągnięciu, pakiet zostanie automatycznie zainstalowany i skonfigurowany.

Pliki konfiguracyjne programu znajdują się w katalogu `/etc/apt` - szczególnie istotny jest plik `/etc/apt/sources.list`, który zawiera listę wszystkich hostów (lub identyfikatorów płyt CD-ROM), z których będą pobierane paczki. Dodawanie do pliku hostów odbywa się ręcznie, należy do pliku dopisać nową linię, np.:

```
deb http://sunsite.icm.edu.pl/pub/Linux/debian sid main contrib non-free
```

co oznacza, że paczki `.deb` będą ściągane z sieci za pomocą protokołu HTTP z adresu: `sunsite.icm.edu.pl/pub/Linux/debian` (`http://sunsite.icm.edu.pl/pub/Linux/debian`).

Jeżeli posiadamy płyty z pakietami Debiana, to należy każdą z takich płyt dodać do listy za pomocą komendy **apt-cdrom add** - program poprosi nas o włożenie płyty, stworzy sobie lokalny indeks pakietów i wpisze identyfikator płyty do pliku `/etc/apt/sources/list`.

Jeżeli korzystamy z sieci do ściągania/odświeżania paczek, to co pewien czas program **apt-get** wymaga odświeżenia danych (na serwerach zmieniają się pakiety, powstają nowe zależności itp). Jeżeli więc komenda z opcją „install” np. nie znajduje poprawnie zależności pomiędzy paczkami, należy wywołać ją z opcją „update”:

```
apt-get update
```

### 3.2.2.1.4. Wtyczka Macromedia Flash dla programu Mozilla

Ponieważ wtyczka Macromedia Flash nie jest częścią programu Mozilla (Mozilla Firefox), należy ją oddzielnie zainstalować, żeby na stronach wyświetlały się programy napisane we Flashu. W tym celu wywołujemy komendę (zakłada się, że program **apt** został już poprawnie skonfigurowany):

```
apt-get install flashplugin-nonfree
```

### 3.2.2.1.5. Logowanie się bez hasła

W Debianie (ze względów bezpieczeństwa) została zablokowana możliwość logowania się bez hasła. Żeby ją odblokować, należy otworzyć plik `/etc/pam.d/common-auth` i znaleźć następującą linię:

```
auth    required    pam_unix.so
```

i zamienić ją na następującą:

```
auth    required    pam_unix.so nullok
```

### 3.2.2.1.6. Polska klawiatura w KDE

(Uwaga! zwykle klawiatura jest poprawnie skonfigurowana od razu po zainstalowaniu systemu)

W celu skonfigurowania w KDE (3.1) polskiej klawiatury tak, żeby nasze znaczki narodowe otrzymywane przy pomocy kombinacji ALT-<klawisz>, należy:

- Otworzyć **Centrum sterowania** a tam: **Regionalne i Dostępność/Układ klawiatury**.
- Zaznaczyć (na górze): „włącz układy klawiatury”.
- W części **dostępne układy** dodać układ „polski”.
- Ustawić podstawowy układ „polski” .
- Sprawdzić, czy na dolnym pasku jest wyświetlone wyświetlone słowo „pl” - co oznacza, że nasz układ klawiatury został włączony.

### 3.2.2.1.7. Konfiguracja Menedżera logowania

Możemy jeszcze skonfigurować Debiana tak, żeby wyświetlał miłe przywitania i automatycznie logował się na użytkownika palacz. W tym celu należy:

- Otworzyć **Panel sterowania**, następnie wybrać **Administracja systemu** i tam **Menedżer logowania**.
- W zakładce **Wygląd** zmienić **Powitanie** na np. „Witamy w systemie SZARP 2.2”. Widoczne logo KDE można zmienić na ikonę umieszczoną w pliku `/opt/szarp/resources/documentation/new/common/praterm-logo.png`.
- W zakładce **Użytkownicy** kliknąć na **obrazki użytkowników** i wybrać np. `default2.png`.
- Na koniec w zakładce **Ułatwienia** zaznaczyć **Włącz automatyczne logowanie** i wybrać użytkownika „palacz”.

### 3.2.2.1.8. WindowMaker

Jeżeli instalujemy system **WindowMaker**, to należy pamiętać, żeby nazwa menadżera okien w KDM-ie była taka sama, jak nazwa programu wykonywalnego (w Debianie: **wmaker**).

### 3.2.2.1.9. CD-ROM

W systemie Debian CD-ROM jest domyślnie montowany w katalogu `/cdrom` lub w nowszych wersjach (zgodnie z LSB 2.0) w katalogu `/media/cdrom`, a nie, jak w przypadku systemu RedHat, w katalogu `/mnt/cdrom`.

### 3.2.2.1.10. GRUB

W systemie Debian istnieje skrypt o nazwie **update-grub**, który automatycznie wykrywa wszystkie dostępne jądra Linuksa i odpowiednio konfiguruje plik `/boot/grub/menu.lst`. Niestety, nie dodaje on partycji Windowsowych. Jeżeli chcielibyśmy skorzystać z tego wygodnego skryptu to należy pamiętać, że dodane przez nas ręcznie partycje (Windowsowe) zostaną usunięte przez skrypt **update-grub**, jeśli umieścimy je pomiędzy liniami:

```
### BEGIN AUTOMAGIC KERNELS LIST
```

oraz

```
### END DEBIAN AUTOMAGIC KERNELS LIST
```

Najlepiej więc dodać własną sekcję na końcu pliku po oznaczeniu:

```
### END DEBIAN AUTOMAGIC KERNELS LIST
```

### 3.2.2.1.11. Zarządzanie modułami jądra

W Debianie, podobnie jak w innych Linuksach, konfiguracja i zarządzanie modułami wykonywane jest za pomocą programów **modprobe** (ładowanie modułów), **depmod** (tworzenie pliku z zależnościami) itp. Tradycyjnie, plikiem konfiguracyjnym jest tu `/etc/modules.conf` (w powodu pewnych zaszczości historycznych, czasami nazywanym `/etc/conf.modules`, ale już rzadko). W pliku tym wypisane są nazwy modułów do załadowania przez system, ich parametry oraz inne możliwe nazwy plików (tzw. aliasy). Dodatkowym plikiem związanym z modułami jest `/etc/modules`, który zawiera listę modułów ładowanych przy starcie systemu (w trakcie działania nie jest używany).

Niestety plik `/etc/modules.conf` jest wykorzystywany przez wiele różnych programów, pojawiają się więc kłopoty wynikające dodawania doń przez programy własnych wpisów (których już zwykle nie usuwają, a jeśli usuwają stosują własne „magiczne triki”, żeby rozpoznać to, co same nawpisywały). Wprowadzono więc w Debianie nowy mechanizm polegający na tym, że oryginalna konfiguracja jest umieszczona w wielu oddzielnych plikach w katalogu `/etc/modutils`. Następnie plik `/etc/modules.conf` jest generowany przy użyciu komendy **update-modules**.

Uwaga! W przypadku jądra w wersji 2.6.x zamiast pliku `/etc/modules.conf`, mamy plik `/etc/modprobe.conf` a zamiast katalogu `/etc/modutils/` mamy katalog `/etc/modprobe.d/`.

### 3.2.2.1.12. Konfiguracja karty graficznej S3 VT835 ProSavage KM266/KL266

Niektóre modele kart graficznych *X-server* nie są poprawnie konfigurowane na etapie instalacji. Objawia się to brakiem obrazu w trybie graficznym. Jedną z takich kart jest:

```
VGA compatible controller S3 Inc. VT8375 [ProSavage9 KM266/KL266]
```

W celu ręcznej konfiguracji należy przełączyć się na konsole tekstową, zalogować się na root-a, a następnie wydać polecenie:

```
dpkg-reconfigure xserver-xfree86
```

W oknie *Select the desired X server driver* należy wybrać *vesa*.

Identyfikator karty graficznej w oknie: *Enter an identifier for your video card* należy pozostawić niezmienny.

W oknie *Please enter the video card's bus identifier* pozostawiamy puste pole. Podobnie postępujemy w oknie *Enter the amount of memory (in kB) to be used by your video card*.

Zawartość okna *Please select the XKB rule set to use* również pozostawiamy niezmienną.

W oknie *Please select your keyboard model* należy wybrać *pc104*.

W oknie *Please select your keyboard layout* należy wpisać *pl*.

Pole oknie *Please select your keyboard variant* pozostawiamy puste, podobnie jak pole w oknie *Please select your keyboard options*.

W oknie *Please select your mouse port* należy wybrać ścieżkę do portu, do którego jest podłączona mysz. Pozycja domyślna (ustawiona przez skrypt konfiguracyjny) jest zalecana, gdyż może zapobiec ewentualnym problemom.

W oknie *Please choose the entry that best describe your mouse* należy wybrać typ myszy. Dla myszy bez kółka należy wybrać pozycję *PS/2*, natomiast dla myszy z kółkiem należy wybrać *imPS/2*. Uwaga: również w tym przypadku pozycja domyślna jest zalecana.

W oknie *Emulate 3 button mouse?* odpowiadamy twierdząco tylko, gdy chcemy programowo emulować trzeci klawisz myszy przy pomocy jednoczesnego wciśnięcia lewego i prawego przycisku myszy. Jeżeli posiadamy mysz trójprzyciskową odpowiadamy przecząco.

W oknie *Enable scroll events from mouse wheel* odpowiadamy twierdząco jeżeli *X-server* ma obsługiwać kółko od myszy.

W oknie *Enter an identifier for your monitor* należy wpisać identyfikator posiadanego monitora. Można też pozostawić istniejącą nazwę proponowaną przez skrypt konfiguracyjny.

W oknie *Is your monitor an LCD device?* odpowiadamy twierdząco, gdy posiadamy panel LCD. W przypadku monitorów CRT (z kineskopem) należy wybrać *Nie*.

W oknie *Please choose a method for selecting your monitor characteristics* należy wybrać pozycję *Advanced*.

W oknie *Enter your monitor's horizontal sync range* zaleca się wpisanie wartości 28-50. Jest to wartość dość uniwersalna, zapewnia poprawną pracę monitora i powinna być bezpieczna dla większości monitorów.

W oknie *Enter your monitor's vertical refresh range* zaleca się wpisanie wartości 43-75. Podobnie jak poprzednio wartość ta jest uniwersalna i bezpieczna.

W oknie *Select the video modes you would like the X server to use* należy zaznaczyć tryby graficzne, w których ma pracować *X-server*. Zaleca się zaznaczenie opcji *800 x 600* oraz *1024 x 768*. W oknie *Please select your desired default color depth in bits* zaleca się wybranie pozycji *24*.

W oknie *Select the XFree86 server modules that should be loaded by default* należy sprawdzić czy zaznaczone są następujące pozycje: *GLcore, bitmap, dbe, ddc, dri, extmod, freetype, glx, int10, record, speedo, type1, vbe*.

W oknie *Write default Files section to configuration file* odpowiadamy *Tak*.

W oknie *Write default DRI section to configuration file* również odpowiadamy *Tak*.

W celu wprowadzenia zmian należy zrestartować *X-server* poleceniem:

```
/etc/init.d/kdm restart
```

### 3.2.2.1.13. Zawieszanie się instalatora podczas autodetekcji parametrów monitora

Czasami może się zdarzyć, że podczas instalacji systemu Linux, po wybraniu opcji *Attempt monitor autodetection* z odpowiedzią *Yes* instalator się zawiesi. Dzieje się tak dlatego, że podczas instalacji jest wywoływany program *get-edid*, który w przypadku niewykrycia poprawnych parametrów monitora zawiesza się. W celu odblokowania instalatora należy przełączyć się na inną, wolną konsolę, zalogować się a następnie wywołać polecenie:

```
killall get-edid
```

*Uwaga!* Tę czynność należy wykonać kilkakrotnie także podczas instalowania pakietów *XServer*.

## 3.2.3. Przygotowanie dystrybucji LiveCD

Niniejszy rozdział opisuje przygotowanie dystrybucji Linuksa typu LiveCD (uruchamianej z płytki), a konkretnie dystrybucji opartej na Ubuntu 5.10, z dołączonym oprogramowaniem SZARP, ale może być wykorzystany także przy tworzeniu innej dystrybucji.

Przy tworzeniu tego opisu korzystałem z opisu autorstwa Reinharda Schiedermeiera (<http://www.informatik.fh-muenchen.de/~schieder/ubuntu-livecd-customize/index.html>) oraz dokumentu z Ubuntu Wiki (<https://wiki.ubuntu.com//LiveCDCustomizationHowTo>).

Ogólna idea jest prosta - bierzemy gotową dystrybucję LiveCD i modyfikujemy ją do naszych potrzeb. Poniżej znajduje się dokładny opis wszystkich kroków. W repozytorium CVS SZARP'a, w katalogu `script/livecd` znajduje się kilka gotowych skryptów wykorzystanych do tworzenia dystrybucji SZARP Demo LiveCD. Przy opisie poszczególnych poleceń znajduje się informacja, jak nazywa się skrypt wykonujący dany krok.

### 3.2.3.1. Przygotowania

Będziemy potrzebowali w miarę szybki komputer, z dużą pamięcią i szybkim dyskiem z dużą ilością wolnego miejsca. Autorzy obu wyżej wymienionych dokumentów zalecają wykonywanie kompresji systemu plików w pamięci, do czego swap powinien być mniej więcej dwa razy większy od obrazu systemu plików, który chcemy stworzyć. W poniższym opisie zakładam kompresję z pliku do pliku, niemniej duży swap także nie zaszkodzi.

Na naszym dużym i szybkim dysku zakładamy sobie pusty katalog, od tej pory zakładam, że wszystkie operacje będą wykonywane w tym katalogu.

### 3.2.3.2. Pobieramy obraz ISO

Ze strony Ubuntu (<http://www.ubuntulinux.org>) ściągamy obraz CD wersji dystrybucji LiveCD. Jeśli obraz mamy na płycie, to kopiujemy zawartość płytki do nowo utworzonego katalogu:

```
mkdir ubuntu-livecd
mount /cdrom
cp -a /cdrom/. ubuntu-livecd
umount /cdrom
```

Jeśli mamy obraz ściągnięty, ale nie wypalony, możemy go podmontować i wtedy kopiować:

```
mkdir ubuntu-livecd
mkdir tmp
mount -t iso9660 -o loop ubuntu.iso tmp
cp -a tmp/. ubuntu-livecd
umount tmp
rm -rf tmp
```

### 3.2.3.3. Kasujemy niepotrzebne programy

Płyta Ubuntu Live zawiera trochę dodatkowych rzeczy, w tym programy przeznaczone dla Windows, przeglądarkę płyty pod Windows itp. Możemy je usunąć, kasując niepotrzebne katalogi i pliki:

```
rm -rf bin disctree doc programs README.diskdefines start.bmp \
start.exe start.ini
```

Zaoszczędzamy w ten sposób sporo miejsca - wśród programów była między innymi pełna wersja OpenOffice 2.0.

### 3.2.3.4. Umieszczamy dokumentację dla użytkowników Windows

Ponieważ nasza płytka będzie zapewne wkładana do napędu przez użytkowników Windows, zadbajmy, aby po włożeniu płyty otworzył im się na przykład opis jak uruchomić naszą płytę. Przygotowujemy plik w HTML'u, zapisujemy go w katalogu `livecd`, np. jako `index.html`, dodajemy ewentualne grafiki i ikonę płyty dla Windows. Ikonę możemy wykonać w programie GIMP, zapisując obrazek o rozmiarach 32x32 punkty jako typ pliku Ikona Windows. Następnie przygotowujemy plik `autorun.info`, zawierający odpowiednie informacje o płycie dla mechanizmu autouruchamiania płyty przez Windows:

```
[autorun]
shellexecute=index.html
icon=praterm.ico
```

Komenda `shellexecute` zadziała tylko na Windows 2000 i XP, jeśli chcemy aby autouruchamianie działało na starszych wersjach Windows, musimy się trochę więcej natrudzić - poszukajmy odpowiedniego opisu w sieci (<http://en.wikipedia.org/wiki/Autorun>).

**Notatka:** W repozytorium SZARP znajdują się odpowiednie pliki `autorun.inf`, `index.html`, `praterm.ico` i `praterm-logo.png`.

### 3.2.3.5. Modyfikujemy konfigurację isolinux

Płyta Ubuntu LiveCD wykorzystuje do ładowania systemu program ISOLINUX. Sam program i pliki konfiguracyjne znajdują się na płycie w katalogu `isolinux`. Możemy je zmodyfikować:

- Nie zależy nam na dodatkowych opcjach ładowania systemu - kasujemy pliki z ich opisami:

```
rm -f isolinux/f*.txt
```

- Modyfikujemy plik `isolinux.cfg` - kasujemy wszystkie sekcje `LABEL` i `F*`. Do linii parametrów jądra możemy dać dwa dodatkowe parametry - `debian-installer/locale=pl_PL` i `debconf/priority=critical`. W ten sposób przy uruchamianiu nie będziemy pytani o wybór języka i klawiatury (ustawienie priorytetu debconfa na `critical` jest bezpieczne, bo poza tym i tak instalator nie zadaje żadnych pytań).

Po tych operacjach cały plik może mieć zawartość:

```
DEFAULT /install/vmlinuz
APPEND casper/enable=true \
casper-udeb/snapshot/backing-file=/cdrom/casper/filesystem.cloop vga=normal \
initrd=/install/initrd.gz ramdisk_size=1048576 root=/dev/rd/0 \
debian-installer/locale=pl_PL debconf/priority=critical rw --
DISPLAY isolinux.txt
TIMEOUT 0
PROMPT 1
```

- Wpisujemy do pliku `isolinux.txt` tekst wyświetlany jako powitanie po załadowaniu płyty. Nie kasujemy pierwszej linii odpowiedzialnej za załadowanie grafiki!
- Przygotowujemy własny plik z wyświetlaną przy starcie grafiką. Plik robimy w Gimpie, szerokość 640 punktów, wysokość 200 do 300 punktów. Zmieniamy mu paletę na 16 kolorów - w menu wybieramy Obraz (Image), Tryb (Mode), Indeksowany (Indexed), wybieramy 16 kolorów, można poeksperymentować z ditheringiem. Zapisujemy całość jako BMP, a następnie używamy dwóch narzędzi z pakietów `netpbm` i `syslinux`:

```
bmptoppm < splash.bmp | ppmtolss16 > splash.rle
```

Uwaga - konkretne kolory z palety używane są też w innym celu - do ustalania koloru tła i tekstu.

**Notatka:** W repozytorium SZARP dostępny jest katalog `isolinux` z gotowymi plikami `isolinux.cfg`, `isolinux.txt` i `splash.rle`.

**Notatka:** Pełen opis składni pliku `isolinux.cfg` można znaleźć pod adresem <http://syslinux.zytor.com/faq.php#config>. W szczególności możemy użyć:

•

```
DEFAULT <etykieta> | kernel <opcje>
```

Podajemy albo słowo *kernel* i domyślne opcje do jądra, albo jedną ze zdefiniowanych później etykiet.

- Etykiety definiujemy następująco:

```

LABEL <etykieta>
kernel <ścieżka do jądra>
append <opcje do jądra>

```

- Domyślny plik wyświetlany przy starcie podajemy tak:

```
DISPLAY <ścieżka do pliku>
```

Plik wyświetlany po naciśnięciu klawisza funkcyjnego (np. F2) definiujemy przez podanie:

```
F2 <ścieżka do pliku>
```

- Możemy podać czas po jakim system zostanie automatycznie załadowany z domyślnymi opcjami. Czas podajemy jako wielokrotność 1/10 sekundy, ustawienie 0 (domyślnie) powoduje, że system zostanie załadowany tylko po wybraniu odpowiedniej opcji przez użytkownika:

```
TIMEOUT <czas>
```

- 

```
PROMPT 1
```

Dyrektywa ta steruje wyświetleniem znaku zachęty *boot*:

W wyświetlanych plikach możemy użyć następujących sekwencji znaków:

- *Ctrl-L* - czyści ekran wypełniając go aktualnym kolorem.
- *Ctrl-O*<*f*><*b*> - ustawia kolor tekstu i tła - zamiast *f* i *b* należy użyć kodów kolorów w postaci pojedynczych cyfr szesnastkowych (od 0 do *f*). Paleta może zależeć między innymi od załadowanego obrazka.
- *Ctrl-X*<*nazwa pliku*> - wyświetla obrazek z pliku o zadanej nazwie (patrz wyżej opis tworzenia pliku w odpowiednim formacie). Po nazwie pliku musi nastąpić znak nowej linii.

Odpowiednie znaki można wprowadzać, np. w edytorze vim, przez poprzedzenie ich wciśnięciem kombinacji klawiszy *Ctrl-V*.

### 3.2.3.6. Rozkompresowujemy system plików

Ubuntu używa skompresowanego systemu plików cloop, zawierającego obraz całego systemu. Musimy go rozkompresować, żeby móc go modyfikować. Do pracy ze skompresowanymi obrazami potrzebne są narzędzia z pakietu *loop-utils* - musimy je zainstalować. Potem robimy:

```

extract_compressed_fs casper/filesystem.cloop > old-fs.ext2
mkdir old
mount -o loop old-fs.ext2 old

```

Dekompresja systemu plików może trwać długo. Moglibyśmy teraz pracować na oryginalnym systemie plików, ale jeśli chcemy stworzyć większy obraz (a po to właśnie zwalnialiśmy miejsce na płycie), musimy stworzyć nowy obraz. Tworzymy plik o pożądanej wielkości - muszą się na nim zmieścić wszystkie dane i aplikacje, które mają się znaleźć na płycie, musi zostać kilkaset (?) MB wolnego miejsca do działania systemu, a całość musi się zmieścić na płycie. Nie obejdzie się zapewne bez eksperymentów. W każdym razie, aby stworzyć obraz o wielkości 3 GB, wykonujemy komendy:

```

dd if=/dev/zero of=ubuntu-fs.ext2 bs=1M count=3000
mke2fs ubuntu-fs.ext2
mkdir new
mount -o loop ubuntu-fs.ext new

```

mke2fs poskarży się, że ma do czynienia ze zwykłym plikiem, a nie urządzeniem, potwierdzamy, że tak ma być (ok). Ostatnim krokiem jest skopiowanie danych z oryginalnego obrazu do nowego. Po tej operacji możemy usunąć stary obraz:

```
cp -a old/. new
umount old
rm -rf old old-fs.ext2
```

### 3.2.3.7. Modyfikujemy obraz systemu plików

W katalogu `new` mamy obraz systemu plików naszej dystrybucji. Chcemy go zmodyfikować, w tym celu „chroot’ujemy” się na nowy system plików:

```
chroot new /bin/bash
```

Jeśli chcemy korzystać z sieci, warto podmontować system plików `proc` i ewentualnie `sysfs` (jeśli korzystamy z niego):

```
mount -t proc proc /proc
mount -t sysfs sysfs /sys
```

W plikach `/etc/apt/sources.list` dodajemy odpowiednie wpisy dla repozytoriów, z których będziemy chcieli korzystać - Ubuntu Universe i Muliverse, SZARP Ubuntu. Konfigurujemy DNS’y (nie zapomnijmy potem skasować pliku `/etc/hosts`) lub dodajemy wpisy do `/etc/hosts`.

Następnie za pomocą `apt-get`’a możemy kasować i dodawać oprogramowanie. Listę zainstalowanych pakietów, w kolejności od największego do najmniejszego, możemy uzyskać następująca komenda:

```
dpkg-query -W --showformat='${Installed-Size;10} ${Package}\n' \
|sort -gr | less
```

Można spokojnie skasować OpenOffice’a, Gimp’a, wszystkie biblioteki związane z Javą, jedną z przeglądarek (Galeon lub Firefox), program pocztowy Evolution, CUPS’a wraz ze sterownikami do drukarek, narzędzia do systemu plików reiserfs, oprogramowanie do łączenia z Palmem itp., niepotrzebne nam pakiety językowe, fonty dla języków azjatyckich, część dokumentacji, słowniki itp. Nie kasujemy na pewno GNOME, Metacity, Nautilus.

Z dodatkowego oprogramowania instalujemy SZARP’a (pakiet `szarp-viewer` - wersję dla Ubuntu), ściągamy potrzebne bazy SZARP’a, warto także zainstalować np. polski pakiet językowy dla Firefoksa. Aktualizację danych SZARP’a ustawiamy na automatyczną, z użytkownikiem np. „livecd”.

Po skończeniu modyfikacji pakietów, czyścimy cache `apt`’a:

```
apt-get clean
```

Warto także zajrzeć do katalogów `/tmp` czy `/var/log`, żeby sprawdzić czy nie przyrosły nam jakieś śmieci. Na koniec, komendą `exit` wychodzimy z `chroot`’a. Pamiętajmy, że wcześniej musimy odmontować wszystkie zamontowane dodatkowo systemy plików (np. `proc`)!

### 3.2.3.8. Dodatkowa konfiguracja

Jest jeszcze kilka rzeczy, które możemy zmienić w naszej dystrybucji. Chcielibyśmy na przykład, aby na starcie uruchamiała się przeglądarka z wyświetloną stroną z informacją o naszej dystrybucji. Możemy to osiągnąć na kilka sposobów. Jeśli chcemy korzystać z Firefoksa, do katalogu `/etc/X11/Xsession.d` dodajemy plik `X99firefox`, zawierający komendę:

```
/usr/bin/firefox /opt/livecd/index.html &
```

Oczywiście podajemy ścieżkę do naszej dokumentacji. Możemy także w katalogu `/etc/skel` dać odpowiedni podkatalog `.mozilla/firefox/` z wpisem dla profilu użytkownika i z ustawioną w pliku `prefs.js` stroną startową przeglądarki na naszą dokumentację.

Jeśli chcemy wykorzystać Galeona, możemy zapisać sesję GNOME z uruchomionym Galeonem i potem odpowiednie pliki skopiować do `/etc/skel`.

Jeśli w menu GNOME pojawiają się jakieś martwe wpisy (po aplikacjach, które usunęliśmy), możemy skorzystać z edytora menu GNOME i usunąć je, a następnie konfigurację menu skopiować do `/etc/skel`. Menu GNOME zapisane jest w katalogach `.config` i `.local` w katalogu domowym.

**Notatka:** Repozytorium SZARP zawiera pliki z dokumentacją - w podkatalogu `doc - index.html`, `draw.html`.

### 3.2.3.9. Zamykanie systemu plików

Kolejną rzeczą do wykonania jest uaktualnienie spisu zainstalowanych pakietów:

```
chroot new dpkg-query -W --showformat='${Package} ${Version}\n' \
> ubuntu-livecd/casper/filesystem.manifest
```

**Notatka:** W repozytorium SZARP'a dostępny jest skrypt `make_manifest.sh`.

Nasz system plików jest gotowy, ale w miejscu wolnym (z punktu widzenia systemu plików) nadal pozostały dane po usuniętych plikach itp., które zostaną umieszczone w skompresowanym systemie plików. Musimy je wyzerować:

```
dd if=/dev/zero of=new/dummy
rm new/dummy
```

Zwróćmy uwagę na komunikat o ilości skopiowanych przez `dd` danych - tyle właśnie wolnego miejsca zostanie do pracy po uruchomieniu naszego systemu. Jeśli ilość ta będzie za mała, nie będą mogły się uruchamiać aplikacje (w tym Gnome) - brak miejsca na pliki tymczasowe i użytkownika. Wyzerowany obszar, niezależnie od wielkości, kompresuje się właściwie w 100 procentach. *Uwaga!* - jeśli dokonywaliśmy wielu operacji na plikach (zwłaszcza dużo kasowania), system plików mógł ulec częściowej defragmentacji. W takim wypadku dobrze byłoby go zdefragmentować, kopiując całość do innego katalogu, usuwając całą zawartość i kopiując na nowo, a dopiero po tym wyzerować.

**Notatka:** W repozytorium SZARP'a dostępny jest skrypt `make_zero.sh`.

Ostatnim krokiem jest odmontowanie naszego nowego systemu plików:

```
umount new
```

### 3.2.3.10. Kompresja nowego systemu plików

Autorzy obu dokumentów, na których się opierałem, zalecali kompresję w pamięci, ja jednak z sukcesem posłużyłem się prostym poleceniem:

```
create_compressed_fs ubuntu-fs.ext2 ubuntu-livecd/casper/filesystem.cloop
```

Kompresja, nawet na szybkim komputerze, może zająć kilka minut.

**Notatka:** W repozytorium SZARP'a dostępny jest skrypt `make_casper.sh`.

### 3.2.3.11. Tworzenie obrazu płyty

Zanim przystąpimy do tworzenia obrazu, musimy jeszcze uaktualnić sumę kontrolną płyty:

```
find ubuntu-livecd -type f -print0 | xargs -0 md5sum > ubuntu-livecd/md5sum.txt
```

**Notatka:** W repozytorium SZARP'a dostępny jest skrypt `make_md5.sh`.

Ostatni krok to stworzenie bootowalnego obrazu płyty:

```
mkisofs \
-o ubuntu-szarp.iso \
-b isolinux/isolinux.bin \
-c isolinux/boot.cat \
-no-emul-boot \
-boot-load-size 4 \
-boot-info-table \
-r \
-V "SZARP Demo LiveCD" \
-cache-inodes \
-J \
-l \
ubuntu-livecd
```

**Notatka:** W repozytorium SZARP'a dostępny jest skrypt `make_iso.sh`.

Obraz możemy wypalić na płytce poleceniem:

```
cdrecord speed=24 -v ubuntu-szarp.iso
```

Jeśli nagrywamy na płytke CD-RW (zalecane dopóki nie upewnimy się, że obraz jest bez błędów), możemy użyć komendy:

```
cdrecord blank=fast speed=10 -v ubuntu-szarp.iso
```

### 3.2.3.12. Testowanie, poprawki

Prawdopodobnie po uruchomieniu komputera z płyty okaże się, że jeszcze nie wszystko jest idealnie. Aby wprowadzić zmiany w obrazie, musimy powtórzyć wszystkie czynności od generowania pliku `filesystem.manifest` włącznie. Jeśli dokonujemy poprawek na płycie, ale poza obrazem, nie zapomnijmy uaktualnić sumy kontrolnej.

Warto także włożyć naszą płytke do komputera z Windows, aby upewnić się że poprawnie działa autouruchamianie.

## 3.2.4. Szablony katalogów domowych (/etc/skel)

Tworząc nowych użytkowników często wykonujemy te same czynności, typu dodawanie polskich literek w KDE, kopiowanie pliku `.vimrc` itp. Czynności tych można sobie oszczędzić przez odpowiednie skonfigurowanie katalogu `/etc/skel`. Wszystkie zawarte w tym katalogu pliki i podkatalogi są przy tworzeniu nowego użytkownika standardowo kopiowane do jego katalogu domowego, z automatyczną zmianą właściciela. Jeżeli chcemy utrzymywać kilka rodzajów konfiguracji dla użytkowników, można utworzyć kilka szablonów katalogów domowych i używać ich w następujący sposób (pod RedHat'em):

```
adduser -m -k katalog
```

Zamiast "katalog" należy wstawić oczywiście nazwę odpowiedniego katalogu z szablonem. Więcej informacji można znaleźć w manualu do komendy **adduser**.

Pod Debianem nie da się z linii komend zmienić katalogu z domyślną konfiguracją, katalog ten jest konfigurowany w pliku `/etc/adduser.conf`.

## 3.2.5. Konfiguracja GRUB-a

GRUB (GRand Unified Bootloader) jest stosunkowo nowym menadżerem bootowania (ang. *bootmanager*) o znacznie większych możliwościach niż program LILO (LILO opisuje Sekcja 3.2.7). Dość wspomnieć, że wszystkie opcje bootowania można zmodyfikować ręcznie w trakcie procesu bootowania (GRUB ma własną prostą „powłokę” dostępną w trakcie bootowania), oraz możliwe jest (i stosunkowo łatwe) robienie rzeczy niestandardowych (typu migracja z jednego dysku na inny, przełączanie z mastera na slave'a itd.). Ponadto, GRUB jest dobrze udokumentowany - opis systemu jest standardowo obecny we współczesnych dystrybucjach w postaci **texinfo** (można więc wydać komendę: **info grub**, żeby dowiedzieć się czegoś więcej o systemie). Również w sieci nie brakuje interesujących dokumentów. Przykłady to m.in.: Multiboot with GRUB Mini-HOWTO (<http://www.tldp.org/HOWTO/Multiboot-with-GRUB.html>) oraz Linux+Win9x+Grub HOWTO (<http://www.tldp.org/HOWTO/Linux+Win9x+Grub-HOWTO/index.html>).

Podobnie jak w systemie LILO, na działanie GRUB-a można wpływać modyfikując pliki konfiguracyjne. W przeciwieństwie do LILO jednak, pliki te odczytywane są każdorazowo z dysku, a nie instalowane do bootsectora zawsze, gdy zmieniane jest cokolwiek w konfiguracji (tak, tak - oznacza to, że GRUB potrafi „zajrzeć” wewnątrz systemu plików). Z tego powodu pliki konfiguracyjne GRUB-a znajdują się w katalogu `/boot`.

Najbardziej istotnym plikiem konfiguracyjnym systemu GRUB jest `/boot/grub/menu.lst`. To tu opisane są wszystkie systemy operacyjne uruchamiane przez GRUB-a. Pierwsza część pliku to opcje globalne - w tej części nie należy zwykle nic zmieniać. W drugiej części pliku opisane są systemy operacyjne, które pojawiają się w menu menadżera bootowania po uruchomieniu komputera. Każda z sekcji rozpoczyna się od słowa **title**. Kolejność sekcji jest istotna - wpływa na kolejność wierszy w menu menadżera bootowania. Przykładowa sekcja ma postać jak poniżej:

```
title           Debian GNU/Linux, kernel 2.6.2-1-686
  root          (hd0,0)
  kernel        /boot/vmlinuz-2.6.2-1-686 root=/dev/sda1 ro
  initrd        /boot/initrd.img-2.6.2-1-686
  savedefault
  boot
```

W lewej kolumnie są zmienne (**title**, **root**, **kernel** itd.), a w prawej kolumnie ich wartości. Znaczenie poszczególnych zmiennych jest następujące:

- **title** jest tekstem, który pojawi się z menu wyboru menadżera bootowania. Ponieważ tekst ów nie wpływa na funkcjonalność systemu, może być dowolnym ciągiem znaków (np. „Moj ulubiony system - Debian”, „RedHat Aurox, jadro 2.6.2” itp.). Warto jednak, szczególnie przy rozbudowanych menu, żeby poszczególne wiersze były łatwo rozróżnialne.
- **root** opisuje dysk i partycję, na której znajduje się katalog `/boot/` z konfiguracją systemu GRUB (uwaga! Niekoniecznie będzie to główna partycja systemu plików).

Należy zwrócić uwagę na to, że GRUB korzysta ze specyficznego nazewnictwa dla dysków i ich partycji. Kolejne dyski są oznaczane jako **hd<numer>**, gdzie **<numer>** to indeks twardego dysku (uwaga! Indeksujemy od zera!), np.: **hd0** to linuksowy `/dev/hda`, **hd1** to linuksowy `/dev/hdb` itd. Grub nie rozróżnia dysków ATA/SATA/SCSI - wszystkie są nazywane **hdX**. Do dyskietek odwołujemy się przez **fd0**.

Druga liczba (ta po przecinku) to oznaczenie numeru partycji (również liczone od zera). W przykładzie powyżej (dysk oznaczony jako **(hd0,0)**) odwołujemy się do pierwszego dysku i jego pierwszej partycji (w standardowym linuksowym systemie plików do partycji tej odwoływalibyśmy się poprzez `/dev/hda1`).

Uwaga! Dyski z rozszerzonych partycji numerowane są od czwórki - czyli pierwszą partycją *extended* będzie **(hd0,4)**.

- **kernel** to ścieżka do pliku z jądrem systemu operacyjnego.
- **initrd** to ścieżka do ramdysku ładowanego przy starcie systemu (Uwaga! Ramdysk musi być w 100% kompatybilny z jądrem).
- **savedefault** to opcja oznaczająca, że dla tego systemu operacyjnego nasz wybór zostanie zapamiętany i automatycznie podświetlony następnym razem (Uwaga! Żeby ten mechanizm zadziałał wymagana jest globalna opcja **default saved**).

W przypadku dodawania systemów operacyjnych Windows nie określamy lokalizacji jądra systemu, a partycję z lokalnym windowsowym programem bootującym (umieszczonym w bootsektorze partycji). Program ten następnie uruchamia jądro Windowsa i resztę systemu. Poniżej pokazano konfigurację GRUB-a dla Windowsa zainstalowanego na trzeciej partycji pierwszego dysku:

```
title Windows 9x
  rootnoverify (hd0,2)
  chainloader +1
```

Uwaga! Jeżeli GRUB nie był wcześniej instalowany na danej maszynie (w sytuacji, gdy np. przechodzimy z systemu LILO na system GRUB) to należy GRUB-a zainstalować. Robi się to przez wydanie komendy **grub-install <partycja do zainstalowania>**, np. **grub-install /dev/hda**.

Uwaga(2)! W celu konfiguracji GRUB-a w Debianie można się posłużyć skryptem do generowania pliku `menu.lst`. Opis znajduje się tutaj.

### 3.2.6. Przywracanie GRUB'a po instalacji Windows

Częstym problemem spotykanym przez użytkowników korzystających na tym samym komputerze z Linuksa i Windows jest niemożność uruchomienia Linuksa (a dokładniej, wyboru uruchamianego systemu operacyjnego) po reinstalacji Windows.

Na dysku komputera, zwykle w tzw. głównym sektorze uruchomieniowym (MBR - Master Boot Record) znajduje się program uruchomieniowy, odpowiedzialny za ładowanie systemu operacyjnego. Linux instaluje tam zwykle program GRUB, pozwalający na wybór ładowanego systemu operacyjnego. Niestety, Windows przy instalacji nie sprawdza czy na dysku są inne systemy - po prostu zapisuje w MBR program ładujący bezpośrednio właśnie Windows.

Poniżej opisano sposób na odratowanie „popsutej” przez Windows instalacji, przy założeniu, że korzystaliśmy z GRUB'a (zobacz też Sekcja 3.2.5). Jeśli mamy dość nową dystrybucję Linuksa, to raczej na pewno mamy do czynienia właśnie z GRUB'em.

Potrzebna nam będzie płyta CD z Linuksem, z której będziemy mogli uruchomić komputer. Może to być płyta instalacyjna którejś z dystrybucji (jeśli oferuje tryb *rescue*), może to być jakaś dystrybucja typu *live* (uruchamiana z płyty CD/DVD) - np. Knoppix. Zakładam jednak, że posłużymy się *Timo's Rescue CD Set* - niewielką dystrybucją Debiana, przygotowaną specjalnie do ratowania systemu. Poniższa instrukcja jest jednak dość uniwersalna i może być wykorzystana z inną płytą. Aby zmniejszyć ryzyko wystąpienia problemów, warto posłużyć się płytą, zawierającą podobną wersję jądra, bibliotek systemowych i GRUB'a jak nasz system (czyli - najlepiej ta sama dystrybucja i z podobnego okresu).

#### Ostrzeżenie

Nieumiejętne posługiwanie się opisanymi narzędziami może w najgorszym przypadku doprowadzić do niemożności załadowania z dysku jakiegokolwiek systemu czy też utraty danych - przed zastosowaniem proszę dokładnie przeczytać cały opis. Kopię tej instrukcji należy wydrukować, albo mieć otwartą na innym komputerze.

W szczególności niniejsza instrukcja nie może być zastosowana dosłownie, jeżeli masz na dysku oddzielną partycję `/boot`.

Nie zabieraj się za wykonywanie opisanych tu czynności, jeśli nie znasz znaczenia takich pojęć, jak np. partycja lub nie znasz przynajmniej podstaw korzystania z komend Linuksa - poproś kogoś o pomoc.

Po kolei:

- Potrzebny nam będzie komputer z dostępem do internetu i nagrywarką CD, a także czysta nagrywalna płyta CD.
- Ze strony <http://rescuecd.sourceforge.net/download.html> ściągamy obraz płyty - plik z rozszerzeniem *iso*. Będziemy musieli wybrać jeden z serwerów lustrzanych, z którego chcemy ściągać płytę - bywa, że jedne z nich zapewnią nam dużo większą prędkość ściągnięcia niż inne - jeśli ściągnięcie idzie wolno, spróbujmy wybrać inny.
- Ściągnięty plik jest obrazem, który musimy wypalić na płycie CD. Uwaga! - to nie jest plik, który należy umieścić na płycie, ale obraz całej płyty! W programie k3b odpowiednia opcja nazywa się **Burn CD image**, w innych programach powinna się nazywać podobnie - np. „Wypal obraz z pliku”.
- Jeśli prawidłowo stworzyliśmy płytkę, to możemy z niej uruchomić komputer z „popsutym” GRUB’em. Jeżeli komputer nie uruchamia się z płytki, może być konieczne ustawienie w BIOS’ie jako pierwszego urządzenia do ładowania systemu CD-ROM’u.
- Po załadowaniu systemu z płyty Rescue-CD pojawi się napis „ISOLINUX” - mamy do wyboru menu, z którego wybieramy pierwszą opcję - *RescueCD*. Robimy to przez wciśnięcie **Enter**. Możemy też po prostu chwilę poczekać, jest to opcja domyślna i zostanie po chwili uruchomiona sama.

Jeżeli korzystamy z innej dystrybucji typu *live*, zwykle wystarczy nacisnąć **Enter**, żeby załadować system w domyślnej konfiguracji. Jeśli korzystamy z płyty instalacyjnej, za pomocą klawiszy funkcyjnych możemy obejrzeć opcje uruchamiania systemu, wśród których powinien być tryb *Rescue*.

- Czekamy cierpliwie aż uruchomi się Linux. Powinniśmy w końcu zobaczyć zachętę do logowania

```
rescuecd login:
```

Wpisujemy nazwę użytkownika *root*, system prosi nas o hasło. Dla *RescueCD* hasło brzmi *rescue*; w trakcie wpisywania nie będzie ono widoczne. Po zalogowaniu zobaczymy tekstową planszę tytułową dystrybucji.

Jeśli korzystamy z dystrybucji *Live*, uruchomiony może zostać tryb graficzny. Następne kroki należy więc wykonywać albo po przełączeniu się na konsolę tekstową (za pomocą kombinacji **Ctrl+Alt+F1**), albo w oknie konsoli, ale zawsze jako użytkownik *root*. W Ubuntu nie ma dostępnego użytkownika *root*, uprawnienia administratora zdobywamy poprzedzając każdą komendę komendą *sudo* - wpisujemy ją i dopiero po spacji właściwą komendę. Jeśli korzystamy z płyty instalacyjnej w trybie *rescue*, to bardzo możliwe, że nie będziemy musieli się logować - od razu dostaniemy konsolę użytkownika *root*.

- Po zalogowaniu wydajemy komendę

```
fdisk -l
```

Powinna ona nam wyświetlić dostępne dyski i partycje.

**Notatka:** Nazwa dysku pod Linuksem wygląda zwykle np. tak: */dev/hda*. Początkowe */dev/* jest wspólne dla wszystkich urządzeń. Od litery *h* zaczynają się nazwy dysków IDE (ATA), ostatnia litera nazwy oznacza o który dysk chodzi - *a* to urządzenie *master* na pierwszym kanale IDE, *b* to urządzenie *slave*, *c* i *d* do odpowiednio *master* i *slave* na drugim kanale. Czyli np. dysk podpięty jako *master* na drugim kanale będzie się nazywał */dev/hdc*.

Dyski SCSI i Serial ATA (jak również urządzenia takie jak pendrive) mają trochę inne nazewnictwo - zamiast *h* będziemy mieli *s*, czyli kolejne dyski będą się nazywały */dev/sda*, */dev/sdb* itd.

Kolejne partycje na dysku otrzymują kolejne numery (uwaga - Linux wyświetla także oddzielnie partycję „rozszerzoną”, na której umieszczone są dyski logiczne, widziane także jako partycje). Tak więc kolejne partycje na pierwszym dysku IDE będą się nazywały `/dev/hda1`, `/dev/hda2` itp.

Możemy otrzymać np. coś w stylu:

```
Disk /dev/hda: 40.0 GB, 40007761920 bytes
255 heads, 63 sectors/track, 4864 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	50	400000	c	W95 FAT32 (LBA)
Partition 1 does not end on cylinder boundary.						
/dev/hda2		50	949	7222811	83	Linux
/dev/hda3		963	4788	30725730	5	Extended
/dev/hda4		4789	4864	610470	82	Linux swap / Solaris
/dev/hda5		964	2365	11261565	83	Linux
/dev/hda6		2366	4788	19462716	83	Linux

Przykładowy spis partycji zawiera dwie partycje podstawowe - jedną z DOS'em lub Windows (`/dev/hda1`), jedną z Linuksem (`/dev/hda2`) a następnie partycję rozszerzoną (`/dev/hda3`), zawierającą dwie partycje z Linuksem (`/dev/hda5` i `/dev/hda6`) oraz partycję wymiany (`/dev/hda4`). Jeśli mamy więcej dysków, partycje zostaną wyświetlone dla każdego z nich. Istotne jest, aby ustalić, która z partycji zawiera główny system plików. W grę wchodzi partycję z identyfikatorem (Id) 83 - Linux. W naszym przykładzie są to partycje 2, 5 i 6. Możemy zgadywać, że główny system znajduje się na pierwszej z tych partycji. Jeśli się pomylimy nic się nie stanie - trzeba będzie próbować z kolejnymi partycjami. Wypisaną przez program **fdisk** tabelkę warto spisać - może się przydać później.

**Notatka:** Może się zdarzyć, że Twój system będzie używał zupełnie innego schematu nazewnictwa dysków i partycji - np.

```
/dev/ide/host0/bus0/target0/lun0/part1
```

Kolejne fragmenty ścieżek oznaczają typ urządzenia (IDE), szynę/kanal (`host0/bus0/target0`), dysk (`lun0`) i partycję (`part1`). Nazwa w takiej postaci jest równie „dobra” jak `/dev/hda2` - należy nazwy te po prostu konsekwentnie stosować zamiast przykładowych z tego opisu, prawdopodobnie wszędzie, poza parametrem `root=` przy ładowaniu jądra (opisane dużo niżej).

- Kolejnym krokiem jest zamontowanie partycji z głównym systemem plików tak, aby była dla nas dostępna. Tworzymy sobie tymczasowy katalog, np.

```
mkdir /mnt/root
```

i montujemy w nim partycję, która powinna zawierać główny system plików. W naszym przykładzie będzie to komenda:

```
mount /dev/hda2 /mnt/root
```

Zamiast `/dev/hda2` musimy wpisać oczywiście nazwę partycji odpowiednią dla naszego systemu.

Jeśli korzystamy z płyty instalacyjnej w trybie *rescue* lub dystrybucji *live*, to być może system sam wykryje naszą główną partycję, lub też podmontuje wszystkie dostępne partycje, np. w katalogach o nazwach odpowiadających nazwom partycji - `/mnt/hda1` itp. W takim razie musimy odpowiednio zmodyfikować parametry podanych dalej komend.

- Sprawdzamy czy dobrze trafiliśmy - wyświetlamy zawartość naszej partycji za pomocą komendy:

```
ls -l /mnt/root
```

Jeśli w wyświetlonym spisie zobaczymy między innymi katalogi `bin` i `boot`, to znaczy że dobrze trafiliśmy. Jeśli nie, odmontowujemy partycję za pomocą komendy:

```
umount /mnt/root
```

i powtarzamy poprzedni punkt dla kolejnej partycji.

**Notatka:** Wykrycie czy podmontowaliśmy odpowiednią partycję może być trudniejsze, jeśli na dysku mamy np. kilka wersji Linuksa. Trzeba się upewnić, że podmontowaliśmy na pewno prawidłową partycję - warto sprawdzić na przykład zawartość pliku `/boot/grub/menu.lst` i obecność plików z jądrem (zwykle `/boot/vmlinuz2.X.Y...` gdzie X i Y to odpowiednie numery wersji jądra). Do szybkiego przeglądania zawartości dysku możemy użyć programu **mc** jeśli mamy do niego dostęp - jest to klon programu Norton Commander. Pliki podglądamy przez F3, wychodzimy z programu przez F10.

- Mamy ustaloną partycję z głównym systemem plików. Podmieniamy teraz główny system plików na tę partycję, wykonując komendę:

```
chroot /mnt/root
```

Musimy jeszcze podmontować system plików *proc*:

```
mount -t proc proc /proc
```

**Notatka:** Jeżeli mamy oddzielną partycję `/boot`, to możemy ją teraz podmontować w odpowiednie miejsce, na przykład:

```
mount /dev/hda2 /boot
```

Nie gwarantuje to jednak sukcesu w instalacji za pomocą skryptu *grub-install*, być może będziemy musieli spróbować sposobu „ręcznego”.

- Dokonujemy teraz reinstalacji GRUB’a na dysku. Jeśli mamy więcej niż jeden dysk, to jako ostatni argument musimy podać nazwę dysku, z którego ładowany jest system - zwykle jest to ten sam dysk, na którym znajduje się nasza partycja z Linuksem. Jeśli źle trafimy (i nadal ładuje się tylko Windows), możemy albo całą procedurę powtórzyć od nowa wybierając inny dysk, albo przestawić kolejność ładowania systemu z dysków w BIOS’ie.

Wydajemy komendę (jeszcze raz - ostatni argument to nazwa naszego dysku, podana tu jest przykładowa, choć najczęściej spotykana):

```
grub-install --no-floppy --recheck /dev/hda
```

Uwaga - zdarzają się wersje skryptu *grub-install*, które nie obsługują opcji „--nofloppy” - jeżeli program poskarży się na niewłaściwe opcje, po prostu wpiszmy polecenie bez tej opcji.

- Jeśli poprzedni krok zakończył się bez błędu, to GRUB został ponownie zainstalowany na naszym dysku. Sprawdźmy czy nigdzie w wypisanych komunikatach nie ma napisów *Error* albo na przykład *Segmentation fault*. Jeśli uważasz, że wszystko poszło w porządku, możesz zresetować komputer pamiętając, żeby uruchomił się z dysku, a nie z płytki. Jeżeli GRUB wypisał informację o błędzie lub komputer nie chce się w ogóle uruchamiać, lub też nadal dostępny jest tylko system Windows, uruchom ponownie komputer z płyty CD i spróbuj kroków opisanych w kolejnych punktach. Jeśli po uruchomieniu widać tylko napis

```
grub >
```

zobacz punkt o ręcznym uruchamianiu systemu za pomocą GRUB’a.

- Jeżeli GRUB poinformował nas o błędzie, możemy spróbować nieco innego sposobu - powtarzamy wszystkie opisane kroki aż do komendy **chroot**, ale już bez niej (montując ewentualnie partycję `/boot` w `/mnt/root/boot`) i próbujemy zainstalować GRUB’a następującą komendą:

```
grub-install --no-floppy --recheck --root-directory=/mnt/root /dev/hda
```

Komenda ta wskazuje, że główny katalog naszej odratowywanej instalacji znajduje się w innym miejscu (/mnt/root). Główna różnica między tym sposobem a poprzednim polega na tym, że teraz używamy GRUB'a pochodzącego z płytki, podczas gdy w poprzednim sposobie próbowaliśmy użyć GRUB'a z naszej instalacji. Uwaga - pamiętajmy, że teraz, po wykonaniu ostatniej komendy (o ile oczywiście się przynajmniej częściowo powiodła), w katalogu /boot/grub naszej partycji mamy pliki z wersji GRUB'a z płytki CD, a nie te z naszej instalacji.

- Jeżeli nadal się nie udało, kolejnym krokiem może być powtórzenie obu poprzednich sposobów, ale podając jako ostatni argument nazwę urządzenia w formacie GRUB'a, czyli (*hdX*), gdzie zamiast *X* podstawiamy odpowiedni numer - pierwszy dysk ma 0, kolejne 1 itp., niezależnie od tego czy są tu urządzenia IDE (ATA), Serial ATA czy SCSI. Możemy też podejrzeć odpowiednie nazwy w pliku /boot/grub/menu.lst. Warto tego spróbować, jeżeli GRUB'owi nie podobają się np. podawane przez nas nazwy urządzeń.

- Jeżeli to też nie pomogło to, jeśli mamy stację dysków, możemy spróbować wykonać dyskietkę uruchomieniową za pomocą komendy:

```
grub-install --recheck (fd0)
```

Próbujemy uruchomić komputer z przygotowanej dyskietki. Zobacz także niżej o ręcznym podawaniu GRUB'owi komend do uruchomienia systemu.

- Jeżeli żaden z powyższych sposobów nie zadziałał lub GRUB wypisał informację o błędzie, zawierającą coś w stylu „/dev/hda does not have any corresponding BIOS drive” (nazwa urządzenia na początku będzie pewnie inna), to oznacza to, że GRUB nie jest w stanie wyciągnąć z BIOS'u poprawnej informacji o dyskach zainstalowanych w systemie - zobacz w takim razie następny punkt.
- Powtarzamy wszystkie operacje, bez wywołania komendy **grub-install**.

## Ostrzeżenie

Operacje opisane w tym punkcie i w następnych mogą uniemożliwić załadowanie z dysku jakiegokolwiek systemu - nie wykonuj ich jeżeli nie rozumiesz co robisz, lepiej poproś o pomoc kogoś z większą wiedzą na temat Linuksa i GRUB'a.

Skoro GRUB źle wykrywa nasze dyski, musimy mu powiedzieć który dysk jest który. W tym celu edytujemy plik /boot/grub/menu.lst. Możemy użyć dowolnego edytora tekstowego (na przykład edytora dostępnego z Midnight Commander pod klawiszem **F4**), ale trzeba się przygotować, że będziemy mieli dostęp tylko do edytora **vim** lub nawet jego starszego brata, **vi**.

**Notatka:** Podstawy obsługi edytora vi(m):

- W dowolnym momencie dwukrotne wciśnięcie **Esc** przenosi nas do tzw. trybu komend.
- W dowolnym trybie możemy klawiszami strzałek przesuwać kursor.
- Będąc w trybie komend naciskamy **i** żeby wejść do trybu edycji, w którym możemy wpisywać i kasować tekst.
- Jeśli nie działają klawisze **Backspace** i **Del**, kasować znaki możemy w trybie komend klawiszem **x**.
- Aby zapisać zmiany, w trybie komend wpisujemy **:w** i naciskamy **Enter**.

- Aby wyjść z edytora, w trybie komend wpisujemy **:q** i naciskamy **Enter**. Jeżeli wprowadziliśmy jakieś zmiany w pliku to, żeby wyjść bez ich zapisywania wpisujemy komendę **:q!** i naciskamy **Enter**.

Zakładając że używamy edytora vim wpisujemy komendę:

```
vim /boot/grub/menu.lst
```

Mamy otwarty plik mapujący urządzenia fizyczne na urządzenia w formacie używanym przez grub'a, np.:

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hd1) /dev/sda
```

Musimy doprowadzić go do sensownej postaci, tzn:

- Jeżeli nie zamierzamy tworzyć dyskietki startowej, możemy usunąć linię z urządzeniem *fd0*.
- Jeśli w pliku mamy jakieś urządzenia, których na pewno nie mamy w systemie (czyli np. */dev/sda* jeśli mamy tylko dyski IDE), to usuwamy je.
- Jeśli naszego dysku brakuje, musimy go dodać.
- Pierwsza kolumna musi zawierać dla pierwszego dysku (*hd0*), dla kolejnych (jeśli mamy więcej niż jeden dysk), (*hd1*) itp. - muszą to być kolejne liczby.
- Druga kolumna powinna zawierać nazwy urządzeń w formacie */dev/hda* itp., przy czym dyski powinny być w takiej kolejności jak są widziane przez BIOS (czyli najpierw *master* na pierwszym kanale itp.), urządzenia SCSI czy Serial ATA po IDE - choć może się to zmienić jeżeli ładujemy system z dysku SCSI czy Serial ATA.

**Notatka:** Jeżeli w komputerze mamy jakieś dyski, które nie są nam potrzebne przy ładowaniu systemu Linux - a więc np. dyski używane tylko przez Windows, czy dysk na którym mamy partycję */home* to, aby ułatwić sobie sprawę, możemy na czas odratowywania GRUB'a odpiąć je od zasilania - tak żeby nie były widoczne w systemie.

Jeśli plik z mapą urządzeń nie był poprawny i nanieśliśmy jakieś poprawki, zapisujemy go i wychodzimy z edytora.

- Teraz zmuszamy GRUB'a do użycia naszej prawidłowej mapy urządzeń, podając komendę

```
grub --device-map=/boot/grub/menu.lst
```

Jesteśmy w powłoce GRUB'a - wydajemy polecenia wykonywane przez GRUB'a, a nie przez system operacyjny. Pierwszym poleceniem jest wskazanie partycji głównej - tej na której jest zainstalowany grub. Partycję podajemy w formacie urządzeń GRUB'a, czyli, w nawiasach, najpierw nazwę dysku (*hd* i kolejny numer), a potem po przecinku kolejny numer partycji. Partycje numerowane są od zera, tak więc np. partycji */dev/hda2* odpowiadać będzie (*hd0,1*) (oczywiście przy założeniu, że pierwszy dysk to właśnie */dev/hda*). Cała komenda będzie więc wyglądać na przykład następująco:

```
root (hd0,1)
```

Zwróćmy uwagę, że w powłoce GRUB'a działa dopełnianie po naciśnięciu klawisza **Tab**. Jeśli wpiszemy:

```
root (
```

i wciśniemy **Tab**, to GRUB podpowie nam dostępne dyski, np.:

```
grub> root (
    Possible disks are:  hd0 hd1
```

Jeśli podamy dysk, może nam podpowiedzieć także partycje:

```
grub> root (hd0,
Possible partitions are:
  Partition num: 0, Filesystem type is ext2fs, partition type 0x83
  Partition num: 4, Filesystem type unknown, partition type 0x6
  Partition num: 5, Filesystem type is reiserfs, partition type 0x83
```

GRUB nie tylko wyświetli nam typ partycji i systemu plików, ale także pominie te, które na pewno nie zawierają prawidłowych plików GRUB'a. Ułatwia nam to wybranie poprawnej partycji.

Druga komenda dokonuje właściwej instalacji - jako parametr podajemy tylko dysk, np.:

```
setup (hd0)
```

Wychodzimy z GRUB'a:

```
quit
```

Restartujemy system i uruchamiamy komputer z dysku. Powinien się uruchomić GRUB, ale bez menu do wyboru systemu - zobacz następny punkt.

- Jeśli nasze operacje doprowadziły do tego, że po uruchomieniu komputera z dysku zamiast menu widzimy powłokę GRUB'a:

```
grub>
```

to oznacza, że ogólnie instalacja GRUB'a się powiodła, ale brakuje mu informacji, jaki system operacyjny ma załadować. Musimy podać następujące komendy:

```
root (hd0,1)
```

Podajemy partycję, na której znajduje się jądro systemu Linux, czyli naszą główną partycję - zobacz uwagi wyżej o nazwach urządzeń w GRUB'ie i o dopełnianiu komend za pomocą klawisza **Tab**.

Następnie określamy położenie pliku z jądrem, które chcemy załadować. Komenda ma postać:

```
kernel /boot/vmlinuz-2.6.14-1-686 root=/dev/hda2
```

Po wpisaniu początkowego *kernel /* warto użyć klawisza **Tab** - GRUB będzie nam podpowiadał znalezione na dysku pliki i katalogi. Jeśli jądro jest na oddzielnej partycji */boot*, to oczywiście ścieżka nie będzie zawierała katalogu */boot*. Jeśli wśród plików nie ma katalogu */boot* czy jądra to znaczy, że jako parametr do komendy *root* GRUB'a podaliśmy złą partycję - możemy komendę *root* powtórzyć z inną partycją. Możemy mieć też kilka jąder do wyboru - wybierzmy to, na którym zwykle pracowaliśmy.

Bardzo ważny jest parameter *root=/...* do jądra. Oznacza on co innego niż wydawana przed chwilą komenda *root* GRUB'a. Po znaku „=” musi wystąpić nazwa głównej partycji naszego odratowywanego Linuksa. Partycja musi być podana w postaci dla jądra Linuksa, a nie GRUB'a, czyli np. właśnie */dev/hda2*. Będzie to inna partycja niż ta z GRUB'em i jądrem tylko wtedy, jeśli mamy oddzielną partycję */boot*. Jeśli jądro zacznie się ładować, ale w trakcie ładowania wyświetli się błąd, że nie można znaleźć głównego systemu plików (np. *Kernel panic - init not found*), to znaczy że podaliśmy zły parametr *root* do jądra - spróbujmy z inną partycją. Oczywiście, jeśli nasze jądro do uruchomienia wymaga jakiś dodatkowych parametrów, to także należy je dopisać.

Kolejne polecenie to podanie ścieżki do pliku *initrd* z obrazem modułów jądra. Większość współczesnych dystrybucji wykorzystuje ten plik. Powinien się znajdować w tym samym katalogu co jądro i musi to być ta sama wersja. Przykładowa komenda to:

```
initrd /boot/initrd.img-2.6.14-1-686
```

Oczywiście znalezienie pliku ułatwia użycie klawisza **Tab**. Jeśli nie możemy znaleźć pliku *initrd*, to może mamy jądro skonfigurowane w ten sposób, żeby z niego nie korzystać - możemy spróbować pominąć więc to polecenie.

Ostatni krok to polecenie załadowania systemu:

```
boot
```

Jeśli się nie powiodło, to znaczy że coś źle wpisaliśmy - jeśli jądro w ogóle nie zaczyna się ładować, sprawdźmy czy dobrze podaliśmy partycję i położenie pliku z jądrem. Jeśli jądro wypisuje błąd już w trakcie uruchamiania Linuksa, to sprawdźmy opcję `root=` i położenie pliku `initrd`.

- Jeśli uruchomił się nasz Linux, to jesteśmy prawie w domu. Oczywiście nie chcemy za każdym razem wpisywać do GRUB'a poleceń przy starcie systemu. Musimy zainstalować porządnie GRUB'a z naszej dystrybucji. Tym razem powinno wystarczyć coś w stylu:

```
grub-install /dev/hda
```

I jeszcze jedna rzecz - upewniamy się, że mamy prawidłowy plik `/boot/grub/menu.lst` - taki jak przed reinstalacją Windows. Jeśli plik został uszkodzony lub skasowany to musimy stworzyć nowy. W Debianie można do tego użyć polecenia:

```
update-grub
```

Zwróćmy uwagę, że plik ten zawiera dokładnie takie same komendy, jak te, które wpisywaliśmy przed chwilą ręcznie.

- Jeśli chcemy z jakiegoś powodu „ręcznie” uruchomić z powłoki GRUB'a system Windows lub DOS, to potrzebna będzie sekwencja następujących komend (oczywiście, z odpowiednimi nazwami partycji):

```
root (hd0,0)
makeactive
chainloader +1
```

Jeśli Windows z jakiegoś powodu uzna, że uruchamia się z nieodpowiedniego dysku, to możemy zamienić dyski albo w BIOS'ie (co będzie wymagało poprawek także w konfiguracji ładowania Linuksa), albo przez wpisanie po `makeactive` polecenia do „zamiany” dysków:

```
map (hd0) (hd1)
map (hd1) (hd0)
```

- Nie wszyscy wiedzą, ale możemy też modyfikować polecenia wykonywane przez GRUB'a przy starcie systemu także gdy mamy prawidłowe menu. W tym celu najeżdżamy na pozycję menu i naciskamy `e`. Potem wybieramy linię i naciskam ponownie `e` żeby ją edytować. Możemy wpisywać oczywiście dowolne komendy GRUB'a.
- Jeśli nie udało nam się ostatecznie przywrócić uruchamiania Linuksa, a usunęliśmy możliwość uruchamiania Windows, to można przeprowadzić odratowywanie systemu Windows z płytki instalacyjnej - w ten sposób będziemy mogli przynajmniej uruchomić Windows.

### 3.2.7. Konfiguracja LILO

Ten opis dotyczy bootmanagera LILO, który w nowszych systemach zostaje wypierany przez potężniejszy manager GRUB.

- Jeśli zainstalowaliśmy Linuksa, należy zmodyfikować bootmanagera. Należy uruchomić Linuksa, zalogować się jako root i wyedytować plik `/etc/lilo.conf`. Następnie należy uruchomić program `/sbin/lilo` i zresetować system. Jak modyfikować plik `/etc/lilo.conf` dowiesz się:

```
man lilo
man lilo.conf
```

**W skrócie:** do `/etc/lilo.conf` trzeba dodać:

```
other=/dev/sda1
```

```
label=dos
table=/dev/sda
```

Najprawdopodobniej sda1 trzeba zamienić na coś, co odpowiada konfiguracji konkretnego komputera. Dyski SCSI to: sda, sdb, sdc... a dyski IDE to: hda, hdb, hdc... Partycje oznaczane są kolejnymi numerami, np. /dev/sda3. Aby dowiedzieć się, na której partycji jest zainstalowany DOS, trzeba jako root napisać:

```
/sbin/fdisk /dev/<odpowiedni_dysk> (np: /dev/sda),
nacisnąć p i obejrzeć tablicę partycji. Aby wyjść z fdisk należy nacisnąć q.
```

- Czasami występuje konieczność uruchamiania z jednego komputera raz Linuksa, raz Windows czy DOSa. Do tego służy lilo, gdzie przy starcie użytkownik może napisać odpowiedni identyfikator ustawiany w /etc/lilo.conf, jednak dla lamerów potrzebne jest coś prostszego - lilo-colors. Należy ściągnąć lilo-0.21-110.src.rpm, jako root zainstalować:

```
rpm -ihv lilo-0.21-110.src.rpm
```

To spowoduje rozpakowanie do katalogu /usr/src/redhat/SOURCES plików: lilo-21.tar.gz oraz lilo-colors-1.21.1.tar.gz. Następnie należy w katalogu /usr/local/src rozpakować plik lilo-21.tar.gz:

```
tar -xzvf lilo-21.tar.gz
```

co utworzy katalog /usr/local/src/lilo. Tamże należy rozpakować plik lilo-colors-1.21.1.tar.gz:

```
tar -xzvf lilo-colors-1.21.1.tar.gz
```

co utworzy katalog /usr/local/src/lilo/lilo-colors-1.21.1. Tamże należy uruchomić:

```
make patch
```

natomiast w pliku Makefile zmienić ostatnią linijkę:

```
install -m 644 sample.bootmsg ${rpm_build_dir}/boot/msg/bootmsg
na linijkę:
```

```
install -m 644 bootmsg ${rpm_build_dir}/boot/msg/bootmsg
```

następnie w katalogu /usr/local/src/lilo uruchomić:

```
make
```

```
make install
```

W katalogu /usr/local/src/lilo/lilo-colors-1.21.1 należy odpowiednio zmienić zawartość pliku bootcreator. Zamiast ostatnich linijek:

```

${vlin1} linux      ..... Linux 2.2.4                runlevel 3      ${vline}${s2}
${vlin1} redhat    ..... Original Redhat Kernel runlevel 3      ${vline}${s2}
${vlin1} x         ..... XWindow 3.3.3                runlevel 5      ${vline}${s2}
${vlin1} dos       ..... Dos                          runlevel 5      ${vline}${s2}

```

przykładowo na linijki:

```

${vlin1} 1         ..... Linux 2.2.12                runlevel 5      ${vline}${s2}
${vlin1} 2         ..... DOS 6.22                      runlevel 5      ${vline}${s2}
${vlin1} 3         ..... Windows 98                    runlevel 5      ${vline}${s2}
${vlin1} 4         ..... Linux 2.2.14                runlevel 5      ${vline}${s2}

```

Tak naprawdę ten tekst nic nie znaczy, jest tylko wyświetlaną na ekranie podpowiedzią dla użytkownika, co ma napisać na klawiaturze, aby uruchomić pożądaný system operacyjny. Oryginalne napisy (pierwsza kolumna za \${vlin1}) są w porządku, ale za długie. W zmienionej wersji są krótkie - wystarczy nacisnąć jeden klawisz z cyfrą i Enter. Aby tak było, musi być właściwie skonfigurowane lilo. Należy więc odpowiednio zmodyfikować /etc/lilo.conf. Przykład dość rozbudowany zgodny z powyższym plikiem bootcreator:

```
boot=/dev/hda
```

```

map=/boot/map
install=/boot/boot.b
prompt
timeout=100
default=2
message=/boot/msg/bootmsg

image=/boot/vmlinuz-2.2.12-20
label=1
initrd=/boot/initrd-2.2.12-20.img
read-only
root=/dev/hda3

other=/dev/hda1
label=2
change
  partition=/dev/hda1
  activate
  partition=/dev/hda4
  deactivate

other=/dev/hda4
label=3
change
  partition=/dev/hda1
  deactivate
  partition=/dev/hda4
  activate

image=/boot/vmlinuz-2.2.14
label=4
initrd=/boot/initrd-2.2.12-20.img
read-only
root=/dev/hda3

```

Najważniejsza jest linijka:

```
message=/boot/msg/bootmsg
```

Oprócz tego należy odpowiednio zsynchronizować pola `label` z zawartością pliku `bootcreator`. Dobrze jest też ustawić odpowiednio duży `timeout` podawany w 0.1 sekundy. Aby zakończyć instalację, należy jeszcze w katalogu `/usr/local/src/lilo/lilo-colors-1.21.1` uruchomić:

```
make
make install
```

i wreszcie uruchomić:

```
/sbin/lilo
```

W przypadku konieczności dalszych modyfikacji plików `bootcreator` i `/etc/lilo.conf` należy powtórzyć powyższe kroki od punktu edycji pliku `bootcreator`.

### 3.2.8. Opis konfiguracji UPS-ów

Ten rozdział opisuje jak zainstalować i skonfigurować UPS-y firm: APC (Sekcja 3.2.8.1), EVER (Sekcja 3.2.8.3), Fideltronik (Sekcja 3.2.8.2), Powerware (Sekcja 3.2.8.4).

### 3.2.8.1. Konfiguracja UPS-a firmy APC

instalacja UPSa APC (wersja Smart 700): po instalacji rpm'a `smupsd.rpm` - instalacja:

```
rpm -ivh smupsd.rpm
```

zrobić link w `/etc/rc.d/rc5.d/` uruchomić `/usr/lib/powerchute/Config.sh`, wybrać: 2 (Smart-UPS); Measure-UPS - no; TCP/IP installed - yes, Use Default Color Scheme; port `/dev/ttyS1` (na `/dev/ttyS0` zwykle jest modem); execute command files as root - yes; send e-mail as root - no; zatwierdzić wybór; increment `UpsTurnOffDelay`; ustawić hasło dla użytkownika `prchute`.

### 3.2.8.2. Konfiguracja UPS-a firmy Fideltronik

Instalacja oprogramowania polega na zainstalowaniu rpm'a `fideltronik` zawierającego oprogramowanie `UPSMonitor`:

```
rpm -i fideltronik-1.0.0-1.i386.rpm
```

Po instalacji należy jeszcze sprawdzić, czy na koniec pliku `/etc/rc.d/init.d/halt`, przed komendą

```
eval $command ...
```

zostało dopisane polecenie `/etc/UPS/upsd/upsoff` tak, że cała linia wygląda mniej więcej tak:

```
/etc/UPS/upsd/upsoff; eval $command ...
```

Podobnie, w pliku `/etc/rc.d/rc.local` pod koniec powinno się pojawić polecenie:

```
/etc/UPS/upsd/upsd
```

Jeżeli używaną dystrybucją jest RedHat 6.1, może okazać się konieczna zmiana lokalizacji plików wykonywalnych z `/etc/UPS/upsd` do `/usr/bin`. Należy więc skopiować tam pliki `upsd` i `upsoff` i zmienić odpowiednie wpisy w `/etc/rc.d/init.d/halt` i `/etc/rc.d/rc.local` - zamienić katalog `/etc/UPS/upsd` na `/usr/bin`.

Po zainstalowaniu rpm'a w katalogu `/etc/UPS` pojawi się także plik `manual.txt`, do którego można zajrzeć w razie problemów. Opisany jest tam jednak sposób instalacji z "czystego tgz'a", a nie z rpm'a.

Należy pamiętać o odpowiednim ustawieniu dipswitchy na UPS Fideltronik:

- 1 - OFF
- 2 - OFF
- 3 - OFF
- 4 - ON

aby po zaniku napięcia i ponownym jego powrocie UPS startował automatycznie i podawał napięcie na wyjście dla komputera.

Testowanie UPS Fideltronik: wstawić `upsd` do `/etc/rc.local` i `upsoff` do `/etc/init.d/halt`, ustawić czas podtrzymania na 5 minut lub krócej i zresetować maszynę. Wypiąć UPS z zasilania i poczekać, czy maszyna zasygnalizuje brak zasilania. Nie czekając 5-ciu minut włączyć UPS do gniazdka - czekać aż maszyna zasygnalizuje powrót zasilania. Potem wyłączyć i sprawdzić, czy nastąpi shutdown i wyłączenie UPS-a, potem po włączeniu do prądu UPS-a powinien wstać komputer.

### 3.2.8.3. Konfiguracja UPS-a firmy EVER

Oprogramowanie do tego UPS-a dostajemy razem z urządzeniem, bądź też możemy ściągnąć ze strony: <http://www.ever.com.pl/pl/download.php> Po ściągnięciu rozpakowujemy archiwum i uruchamiamy program instalacyjny:

```
tar xzvf powersoft-<wersja>.tar.gz
cd powersoft-<wersja>
./install
```

W pliku `/etc/powersoft/powersoft.conf` dostosowujemy konfigurację (najważniejsze jest ustawienie właściwego portu szeregowego). Po tym możemy już uruchomić demona sterującego:

```
/sbin/psman init
```

Możemy w każdym momencie podejrzeć status UPS-a uruchamiając komendę:

```
/sbin/psman status
```

Jeśli chcemy (a zapewne chcemy), by oprogramowanie uruchamiała się przy każdym starcie systemu, dodajemy dwie linijki do pliku `/etc/rc.local`:

```
/bin/echo "Uruchomienie serwisu PowerSoft"
/sbin/psman init
```

### 3.2.8.4. Konfiguracja UPS-a firmy Powerware

Powerware produkuje szeroką gamę UPS-ów, dołączając do nich oprogramowanie dla kilkunastu systemów operacyjnych, szczególnie Uniksów (w tym dla Linuksa). Oprogramowanie to występuje w dwóch wariantach o uroczych nazwach *FailSafe* i *LanSafe*. *LanSafe* (opisany poniżej) ma większe możliwości, ale niestety nie chodzi pod Windows. Opis instalacji dotyczy modelu Powerware 5115.

Hardware: UPS-a podłączamy do prądu, włączamy (kilkakrotne popiskiwanie na początku jest normalne - to taki autotest) i czekamy kilkanaście godzin aż się porządnie sformatuje. Łączymy UPS-a z COM-em w komputerze za pomocą dostarczonego z zestawem kabelka. Obciążamy UPS-a: można do niego podłączyć do czterech urządzeń na raz.

Software: może się zdarzyć, że na dołączonej płytce nie ma oprogramowania *LanSafe* dla modelu 5115, chociaż *FailSafe* (wersja uproszczona pod Windows) działa bez zarzutu. Należy więc ze strony <http://www.powerware.com> (<http://www.powerware.com>) ściągnąć najnowszą wersję pakietu *LanSafe* dla Linuksa (w tym przypadku była to wersja 4.16-1 w pliku `ls3_416_linux.tar`). Rozpakowujemy archiwum i uruchamiamy instalację (`./install.sh`). Na pytania odpowiadamy defaultowo. Na pytanie o źródło instalacji odpowiadamy wpisując ścieżkę i nazwę archiwum tar (w tym przypadku `ls3_415_linux.tar`). Na pytanie „Is this computer part of UPS group?” odpowiadamy w zależności od konfiguracji sprzętowej: jeżeli do UPS-a jest podłączonych więcej niż 1 komputer linuksowy i są one połączone siecią to odpowiadamy YES. W przeciwnym razie: NO (domyślne). W przypadku YES-a zostaniemy zapytani czy nasz komputer jest kontrolerem grupy (jest podłączony do UPS-a kabelkiem DB9), czy jedynie członkiem grupy (ma kontakt z jakimś kontrolerem przez sieć). Pozostałe pytania dotyczą konfiguracji modelu UPS-a, portu szeregowego oraz czasów: po którym użytkownicy są informowani o zaniku prądu; po którym zamykany jest system; trwania procedur zamykania systemu.

Uwaga! W przypadku grupy należy ustawić wszystkim członkom grupy wartości czasów takie same lub mniejsze niż ma kontroler. Dlaczego? Otóż UPS po zamknięciu systemu operacyjnego kontrolera

wyłącza się. Jeżeli zwykły członek grupy ma ustawione dłuższe czasy niż kontroler, wtedy UPS mu odetnie zasilanie zanim ten zdąży się bezpiecznie zamknąć.

Uwaga(2)! W grupie musi być przynajmniej 1 kontroler.

Uwaga(3)! Windowsowy FailSafe nie współpracuje z innymi komputerami, więc w przypadku Windows nie należy przyłączać więcej niż jednego komputera do UPS-a.

Uwaga(4)! Jeżeli LanSafe działa w sieci, należy na wszystkich komputerach identycznie ustawić `/etc/hosts` przypisując aliasy do numerów IP interfejsu `eth0`, nie do interfejsu `lo` (127.0.0.1).

Ostatnim krokiem instalacji jest ustalenie hasła na UPS-a, wtedy uprawnieni użytkownicy (niekoniecznie `root`) będą mogli podglądać stan UPS-a oraz go konfigurować. UPS Powerware może być ponownie skonfigurowany przy pomocy narzędzi skryptu instalacyjnego `install.sh`, a w czasie działania przy pomocy programów: konsolowego `/usr/ls3/ls3con` (w pewnym wersjach może wyrzucać `core'a`) oraz `/usr/ls3/ls3conX` (wersja pod ikisy z animacjami i helpem). Narzędzia `ls3con` powinny poprawnie działać w przypadku kontrolera i zwykłego członka.

Instalując oprogramowanie LanSafe w wersji 4.17a z płytki dostarczonej z UPS-em należy zwrócić uwagę, aby program instalacyjny uruchamiać z katalogu, w którym on się znajduje (a więc np. `/mnt/cdrom/LANSAFE`), a poza tym nie wolno zmieniać domyślnej ścieżki instalacyjnej.

### 3.2.9. Wykorzystanie czujników z płyt głównych - `lm_sensors`

`lm_sensors` jest pakietem umożliwiającym wykorzystanie czujników obecnych na współczesnych płytach głównych. Dotyczy to czujników temperatury, napięcia jak i ilości obrotów wiatraczka coolera. Dzięki tym informacjom można sprawdzać stan hardware'u w trakcie pracy i odpowiednio reagować na niektóre awarie (np. nieodpowiednie napięcie pochodzące z zasilacza, spalony wiatraczek - jeżeli procesor się jeszcze jakoś trzyma, itd.). Możliwe jest również takie skonfigurowanie pakietu, by pewne działania były podejmowane automatycznie (nieinteraktywnie).

Poniższy opis zawiera informacje dotyczące głównie najnowszych dostępnych wersji `lm_sensors`, gdyż właściwie tylko one mają szansę obsłużyć czujniki obecne na nowych płytach głównych. W konfiguracji `lm_sensors` sporo się zmienia, więc opis ten może być nieadekwatny do starszych wersji.

Jeżeli mamy starszą płytę, możemy spróbować użyć `lm_sensors` z pakietów RPM. Należy użyć pakietów o nazwach `i2c`, `lm-sensors-drivers` i `lm-sensors` o odpowiadających sobie numerach wersji. W jądrach 2.4.x `i2c` jest standardowo dostępne i nie trzeba go instalować o ile numer wersji jest zgodny z używaną wersją `lm-sensors`. Zwykle jednak będziemy potrzebowali skompilować `i2c` i `lm_sensors` ze źródeł.

Ze stron <http://www.lm-sensors.nu> ściągamy źródła do `i2c` (np. plik `i2c-2.7.0.tar.gz`) oraz źródła do `lm_sensors` (np. `lm_sensors-2.7.0.tar.gz`).

Ponieważ skrypty instalacyjne korzystają z pliku `/usr/src/linux/.config` w przypadku jądra instalowanego z rpm'ów, należy ten plik utworzyć (oczywiście `/usr/src/linux` musi być prawidłowym linkiem na katalog z kernelem, np. `/usr/src/linux-2.4.18-3`):

```
cd /usr/src/linux
cp configs/kernel-*-i386.config .config
```

Musi być zainstalowany pakiet `kernel-source-`uname -r`.i386.rpm`

Instalacja polega na rozpakowaniu źródeł, ich przekompilowaniu i przekopiowaniu w odpowiednie miejsce:

```
tar -xzf i2c-2.7.0.tar.gz -C <katalog docelowy>
```

```
cd <katalog docelowy>i2c-2.7.0/
make all
make install
```

Ponieważ źródła i2c i lm\_sensors nie używają autoconf'a, a w Red Hat v7.2 i 7.3 nieco zmieniła się organizacja plików, więc przed kompilacją należy zmienić plik Makefile w obu katalogach ze źródłami zamieniając linię:

```
MODDIR := /lib/modules/`grep UTS_RELEASE $(LINUX_HEADERS)/linux/version.h|cut -f 2 \
-d' "' \misc
```

na linię:

```
MODDIR := /lib/modules/2.4.18-3/kernel/drivers/i2c
```

lub odpowiednią dla zainstalowanej wersji jądra. Można także zmienić prefix na */usr*, aby programy instalowały się w katalogach */usr/bin* itp. zamiast */usr/local/bin*. Masę ostrzeżeń w trakcie kompilacji można zignorować. Jeżeli wszystko poszło dobrze, to w podanym katalogu z modułami pojawiły się nowe lub podmienione moduły z przedrostkiem i2c- (np. i2c-core, i2c-dev, i2c-isa, i in.).

Kompilujemy i instalujemy lm\_sensors (razem ze sterownikami) ze źródeł (podobnie należy ustawić ścieżkę do modułów w pliku Makefile):

```
tar -xzf lm_sensors-2.7.0.tar.gz -C <katalog docelowy>
cd <katalog docelowy>lm_sensors-2.7.0/
make all
make install
```

Może się zdarzyć, że mimo braku komunikatów o błędach moduły nie zostaną poprawnie zainstalowane. Warto więc sprawdzić, czy w katalogu z modułami i2c (a więc np. */lib/modules/2.4.18-3/kernel/drivers/i2c*) znajdują się uaktualnione wersje modułów. Jeśli nie, należy ręcznie przekopiować wszystkie pliki z rozszerzeniem *.o* z podkatalogów *kernel*, *kernel/busses* i *kernel/chips* katalogów ze źródłami i2c i lm-sensors.

Można teraz uaktualnić zależności modułów i utworzyć nowy obraz modułów jądra (w RH 7.3) przez wydanie jako root komend:

```
depmod -a
cd /boot
mv initrd-`uname -r`.img initrd-`uname -r`.img.bak
mkinitrd initrd-`uname -r`.img `uname -r`
```

Po tym należy przeładować system.

Po starcie systemu uruchamiamy jako root program **sensors-detect**, który powinien nam się zainstalować podczas instalacji lm\_sensors. Na wszystkie pytanie odpowiadamy naciskając ENTER (domyślna odpowiedź). Jeżeli w trakcie działania skrypt wykryje jakieś czujniki w systemie (Success!) to dobrze, w przeciwnym razie to znaczy, że albo mamy nieobsługiwany chip, albo w ogóle nie mamy chipów-czujników na płycie (bo jest na przykład za stara), albo nie zainstalowaliśmy poprawnie modułów i2c. W tym przypadku skrypt krzyczy, że nie może załadować modułów i2c-\*. Wtedy musimy zacząć czytać README do i2c, powinno pomóc. Przykładowo - pomocnym działaniem może być ręczne załadowanie modułu i2c-core:

```
insmod i2c-core
```

Po tej operacji należy jeszcze raz spróbować uruchomić **sensors-detect**. Skrypt **sensors-detect** na samym końcu wyświetli nam informacje wpisów, które powinniśmy dodać do `/etc/modules.conf` oraz które moduły powinniśmy załadować w skryptach startowych w `/etc/rc.d`.

Uwaga! Przed uruchomieniem programu, albo w przypadku, gdy czujniki nie są wykrywane **sensors-detect**, warto się dowiedzieć jakie chipy są zainstalowane na danej płycie - dobrym punktem startu jest serwis: <http://mbm.livewiredev.com/mobolist.html>

Wpis do `/etc/modules.conf` dodajemy, natomiast ładowanie odpowiednich modułów możemy zostawić skryptowi `lm_sensors.init` - należy go skopiować do `/etc/rc.d/init.d/lm_sensors` i dodać do usług uruchamianych podczas startu systemu. Po tych operacjach znów przeładujemy system.

Teraz uruchamiamy program `sensors`. Powinien wyświetlić na ekranie jakieś wartości. Jeżeli nie wyświetli zupełnie nic a skrypt **sensors-detect** wykrył uprzednio jakieś chipy, to oznacza, że prawdopodobnie niepoprawnie zainstalował się pakiet `lm_sensors`. W tym przypadku należy go przekompilować (poprzedni punkt) i spróbować powtórnie. Jeśli wszystko poszło dobrze, program wypisze nam informacje o wykrytych czujnikach i pokazywanych przez nie wartościach. Zdarza się także, że program **sensors** nie działa (SIGSEGV), ale nie przeszkadza to w działaniu np. demona `mondo`. W takiej sytuacji trudno jest jednak zweryfikować, że czujniki w ogóle działają.

Wraz z pakietem dostarczana jest także biblioteka **libsensors**, która umożliwia sprawdzanie stanu czujników z poziomu jęz. C.

Następnie należy przejrzeć i ewentualnie poddać edycji plik konfiguracyjny `/etc/sensors.conf` (nie jest to konieczne jeżeli będziemy korzystali z `mondo` - zobacz dalej). Zawiera one dane dotyczące czujników, które chcemy obserwować i dozwolonych wartości dla obserwowanych parametrów. Dla każdego czujnika jest oddzielna sekcja (rodzaj czujnika jest wypisywany przez program `sensors`). Po znalezieniu odpowiedniej sekcji możemy pozmieniać parametry. Zwykle będziemy chcieli ignorować wartości napięć zasilających. Można to osiągnąć przez dodanie wpisów:

```
ignore in0
ignore in1
ignore in2
...
```

Podobnie możemy zignorować niepodłączone lub niedziałające prawidłowo czujniki obrotów i temperatury:

```
ignore fan1
ignore fan3
ignore temp3
...
```

Aby zmiany w konfiguracji zostały uwzględnione, należy wywołać program `sensors` z parametrem:

```
sensors -s
```

Jeżeli chcemy zmieniać dozwolone wartości parametrów, to także dobrze jest zrestartować całe `lm_sensors` przez wydanie komend:

```
service lm_sensors stop
service lm_sensors start
```

```
sensors -s
```

Plik zmieniamy dopóki program `sensors` uruchamiany bez parametrów nie wyświetli nam prawidłowych informacji o interesujących nas parametrach. Więcej informacji można znaleźć na stronach manuala dla `sensors` i `sensors.conf`.

Standardowa instalacja pakietu `lm_sensors` pozwala właściwie tylko na interaktywne odpytywanie o wartości parametrów oraz na zbieranie danych na dysku przez demona `sensord`, kompilacja którego wymaga jednak dociągnięcia dodatkowych bibliotek. Lepszym rozwiązaniem jest więc skorzystanie z demona `mondo`.

Mondo można ściągnąć z <http://mondo-daemon.sourceforge.net> (<http://mondo-daemon.sourceforge.net/download.html>). Dostępna jest wersja RPM, niestety na RH 7.3 nie chce ona działać, wobec tego lepiej jest ściągnąć i skompilować źródła:

```
tar xzf mondo-0.9.tar.gz
cd mondo-0.9
./configure --prefix=/usr --sysconfdir=/etc
make
make install
```

Kolejnym krokiem jest utworzenie pliku konfiguracyjnego dla `mondo`. W tym celu należy uruchomić program `mondo-setup`, który wygeneruje przykładowy plik konfiguracyjny. Standardowo program ten przez 2 godziny obserwuje zachowanie komputera. Jeżeli nie mamy tyle czasu, należy go zmusić do pracy przez krótszy czas (np. 10 minut) przez wydanie komendy:

```
mondo-setup -t 10 -s /etc/sensors.conf -o /etc/mondo.conf
```

Przez czas działania programu można pograć w jakąś zajmującą dużo zasobów gierkę, np. *chromium*. Po zakończeniu pomiarów należy przejrzeć plik `/etc/mondo.conf`. Parametr *DELAY* opisuje czas (w milisekundach), co jaki sprawdzane są czujniki. Następnie następują sekcje opisujące poszczególne obserwowane urządzenia i czujniki. Jeśli są tu jakieś czujniki, których nie rozpoznajemy, należy je po prostu usunąć. Przykładowa sekcja może mieć postać:

```
"w83627hf-isa-290:Processor FAN" {
ABOVE 3096.00 "halt -fp"
BELOW 3000.00 "halt -fp"
}
```

Opisuje ona górny i dolny limit obrotów wentylatora na procesorze. Przekroczenie którejkolwiek wartości spowoduje wykonanie zdefiniowanej komendy, w tym przypadku natychmiastowe wyłączenie komputera. Dla danego czujnika może być zdefiniowanych kilka linii *ABOVE* lub *BELOW*, wykonane będą wszystkie dla których przekroczone zostaną limity.

Należy zwrócić uwagę, że domyślnie ustalone przez program `mondo-setup` limity są bardzo restrykcyjne i powinny zostać zwiększone - w innym przypadku grozi ciągle podnoszenie alarmu.

W przypadku innego typu akcji, np. wysyłania maili, niewskazane byłoby wysyłanie ich co zadany czas sprawdzania czujników (domyślnie 4 sekundy). W związku z tym można ustawić (globalnie lub dla konkretnej sekcji) dwie opcje:

```
GRACE <clicks>
THROTTLE <clicks>
```

Pierwsza z nich mówi, ile razy alarm dla czujnika powinien być ignorowany. Jeśli ustawimy ją na 5, a *DELAY* na standardowe 4000, to alarm zostanie podniesiony dopiero po 20 sekundach. Jeżeli w tym

czasie obserwowany parametr powróci do akceptowanego stanu, alarm nie zostanie podniesiony. Druga opcja - *THROTTLE* określa, co ile sprawdzeń będzie wykonywana komenda przypisana do alarmu jeżeli parametr będzie cały czas przekraczał dopuszczalne wartości. Np. ustawienie *THROTTLE* na 450 spowoduje, że akcja będzie wykonywana co 30 minut. Poniższy fragment pliku opisuje konfigurację, gdy w przypadku przekroczenia temperatury 45 stopni wysyłany jest mail (co godzinę), a po przekroczeniu 55 stopni system jest zamykany:

```
DELAY 4000

"w83627hf-isa-290:temp1" {
GRACE 2
THROTTLE 900
ABOVE 45.00 "echo 'Temperatura przekroczyła 45 stopni' \
| mail -s 'Uwaga!' root@admin"
ABOVE 55.00 "halt -fp"
}
```

Przekroczenie stanów alarmowych jest także zapisywane w logach systemu. Po przetestowaniu działania można dodać mondo do usług uruchamianych przy starcie systemu (odpowiedni plik `/etc/rc.d/init.d/mondo` jest tworzony przy instalacji programu). Uwaga: wpisanie zbyt restrykcyjnych wartości w pliku konfiguracyjnym może doprowadzić do sytuacji, gdy komputer będzie się wyłączał zaraz po starcie!

Ponieważ plik konfiguracyjny mondo nie respektuje limitów ustawionych w `/etc/sensors.conf`, więc jeżeli `lm_sensors` będą używane tylko dla potrzeb mondo, można darować sobie edycję pliku `/etc/sensors.conf`.

### 3.2.10. Konfiguracja mirroringu - RAID-1

Do obsługi macierzy służą programy z pakietu `raidtools`. Najlepiej jednak (najprościej) zainstalować macierz podczas instalacji RedHata. Oto krótki opis jak to zrobić (RAID-1 z dwoma dyskami):

- Stworzyć następujący układ partycji:

```
/dev/hda1 Linux RAID 3000M
/dev/hda5 Swap 256
/dev/hda6 DOS 256
/dev/hda7 Linux RAID reszta
```

Takie same partycje należy utworzyć na drugim dysku (np. `/dev/hdc`).

- Wybrać opcję "Make RAID Device".

Należy tu połączyć partycje `/dev/hda(c)1` i `/dev/hda(c)7` jak pokazano poniżej:

```
/dev/md0 - /dev/hda1 + /dev/hdc1 - /
/dev/md1 - /dev/hda7 + /dev/hdc7 - /opt
```

Oba są typu RAID-1. W tej konfiguracji mamy po jednym swapie na każdy dysk (oba są wykorzystane).

- Zainstalować w normalny sposób system.
- Po resecie nie uruchomi się Linux, ponieważ program instalacyjny ustawił w `/etc/lilo.conf`, że `root to /dev/md0`, a to nie zadziała. Trzeba jeszcze raz odpalić program instalacyjny RedHata z płytki, po starcie X nacisnąć `Ctrl-Alt-F2`.

Następnie wykonać komendy:

```
mkdir /mnt2
mknod /dev/hda1 b 22 1
mount /dev/hda1 /mnt2
mv /mnt2/etc/lilo.conf /mnt2/etc/lilo.conf.old
cat /mnt2/etc/lilo.conf.old | sed -e "s/md0/hda1/" > /mnt2/etc/lilo.conf
/mnt2/sbin/lilo -r /mnt2
```

- Mamy stworzoną macierz dyskową z software'owym mirroringiem. Dane są równolegle zapisywane na dwa dyski, o synchronizację dba proces mdrecovery. W pliku `/proc/mdstat` znajduje się aktualny stan macierzy. Z niego można się dowiedzieć o awarii. Niestety w wersji 6.2 nie jest zbyt wygodne naprawienie macierzy dyskowej po awarii jednego z dysków. Co gorsza system nie zareaguje w specjalnie widoczny sposób w tej sytuacji. Plik `/proc/mdstat` pokaże, które dyski działają, a które nie. Dysk awaryjny zostanie wyłączony z macierzy. Poniżej pokazano przykładowy wynik wydania komendy **cat /proc/mdstat** (uwaga - przykład jest dla innej konfiguracji niż opisana powyżej):

```
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 sdb1[1]
      1052160 blocks [2/1] [_U]

md1 : active raid1 sdb2[1]
      5245120 blocks [2/1] [_U]

md2 : active raid1 sdb5[1] sda5[0]
      2096384 blocks [2/2] [UU]

md4 : active raid1 sdb6[1]
      1052160 blocks [2/1] [_U]

md3 : active raid1 sdb7[1]
      1052160 blocks [2/1] [_U]

md5 : active raid1 sdb8[1]
      25053248 blocks [2/1] [_U]
```

unused devices: <none>

„U” (czyli *up*) oznacza, że dysk działa w macierzy. „\_” oznacza, że dysk jest wyłączony lub go brakuje w macierzy.

Na przykładzie powyżej widać, że tylko jedna partycja w macierzy ma działające oba dyski ([2/2] [UU]) - partycja md2 (składająca się z partycji sdb5 i sda5). Reszta partycji macierzowych ma tylko 1 dysk ([2/1] [\_U]). Co zrobić, by zainstalować nowy dysk? Należy go sformatować w ten sam sposób (utworzyć te same partycje), co istniejący dysk. Powinien to być dysk o takim samym rozmiarze, ale w przypadku trudności ze znalezieniem dysku o takiej samej pojemności można włożyć większy, wolny obszar można przeznaczyć na dodatkową partycję, która powinna się jednak znajdować na końcu (np. `/dev/hdc9`).

Po odpowiednim podzieleniu dysku na partycje i sformatowaniu go należy wydać następujące komendy (dodające do macierzy drugi dysk):

```
raidhotadd /dev/md0 /dev/hdc1
raidhotadd /dev/md5 /dev/hdc7
```

W tle (w wolnym czasie procesora) rozpocznie się proces synchronizacji dysku z macierzą. Nie nastąpi to od razu, ale po jakimś czasie (do kilku godzin przy słabo obciążonym systemie). Proces wykonuje się całkowicie przezroczysto, nie koliduje to w żaden sposób z zapisami i odczytami z

macierzy. W pliku `/proc/mdstat` pokazywany jest aktualny stan synchronizacji wraz z przewidywanym czasem zakończenia (`finish`). Poniżej pokazany jest wynik komendy `cat /proc/mdstat` po wywołaniu polecenia `raidhotadd /dev/md0 /dev/sda1`:

```
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 sda1[2] sdb1[1]
      1052160 blocks [2/1] [_U]
      [==>.....] recovery = 18.0% (190500/1052160) finish=1.3min s
peed=10583K/sec
md1 : active raid1 sdb2[1]
      5245120 blocks [2/1] [_U]

md2 : active raid1 sdb5[1] sda5[0]
      2096384 blocks [2/2] [UU]

md4 : active raid1 sdb6[1]
      1052160 blocks [2/1] [_U]

md3 : active raid1 sdb7[1]
      1052160 blocks [2/1] [_U]

md5 : active raid1 sdb8[1]
      25053248 blocks [2/1] [_U]

unused devices: <none>
```

Uwaga! Nie wolno robić żadnych modyfikacji tablicy partycji, kiedy pracuje na nich aktywna macierz. Aby zmienić układ dysku, należy zatrzymać obie macierze (komendą `raidstop`). W konfiguracji, którą pokazałem powyżej system znajduje się na aktywnym urządzeniu `/dev/md0`, trzeba zatem oba dyski włożyć do oddzielnego systemu z Linuksem i wtedy przeprowadzać modyfikacje. Zmiany należy uwzględnić w plikach `/etc/raidtab` na obu dyskach.

### 3.2.11. Konfiguracja Specialixa

- Moduł do obsługi Specialixa nazywa się `sx` i powinien być we wszystkich aktualnych jądrach automatycznie ładowany przy starcie systemu (po wykryciu urządzenia). Z jakiś względów moduł ten nie jest obecny przynajmniej w niektórych wersjach jąder SMP (dla systemów wieloprocessorowych).
- Do działania karty potrzebny jest jeszcze firmware oraz program ładujący ten firmware - `sxboot`. Instalację najprościej przeprowadzić ściągnąjąc paczkę `sx-base` dostępną w repozytorium APT firmy Praterm. W paczce znajduje się także program `sxmkdev`, tworzący nazwy urządzeń (uwaga - nie działa to dla `udev`'a). W razie potrzeby program ten można zastąpić następującym skryptyem:

```
#!/bin/sh
cd /dev
t=0
mknod specialix_sxctl c 10 167
while [ $t -lt 64 ]
do
echo -n "$t "
mknod ttyX$t c 32 $t
mknod cux$t c 33 $t
t=`expr $t + 1`
```

```
done
echo ""
rm /etc/psdevtab
ps > /dev/null
```

### 3.2.12. Konfiguracja Moxy

Instalacja karty pod RedHat'em polega na uruchomieniu programu instalacyjnego, który sam kompiluje moduły do jądra, instaluje je, tworzy urządzenia: dla C320Turbo `/dev/ttyAnn`, dla C168H `/dev/ttyMn` i `/dev/cumn` i ustawia odpowiednie linki w `/etc/rc.d/` (do automatycznego startu podczas ładowania systemu - tylko w C320Turbo; w C168H trzeba to zrobić ręcznie). Oprócz driver'ów, oprogramowanie C320Turbo zawiera program diagnostyczny `mxdiag`, emulator terminala `mxterm`, monitor transmisji na portach `mxmon`. Programem `mxdiag` można sprawdzić stan Moxy. Dla C168H są to odpowiednio: `msdiag`, `mstern` i `msmon`. Całość nie przedstawia problemu i jest doskonale udokumentowana, do dystrybucji dołączony jest opis, który krok po kroku opisuje komendy jakie trzeba wykonać. Jedynym odstępstwem w stosunku do dokumentacji, które należy wykonać, jest wstawienie do `rc.local` opóźnienia po wywołaniu driverów moxy po to, żeby moduły zdążyły się załadować przed uruchomieniem `S99parstart`. Najprościej na końcu `rc.sysinit` bezpośrednio wywołać `rc.moxa` lub `rc.mxser`:

```
/etc/rc.d/rc.moxa
/bin/sleep 3
```

Słów parę należy jeszcze powiedzieć o konwencji nazewnictwa urządzeń Moxy. Dla serii C320Turbo jest to `/dev/ttyAxy`, dla C168/H jest to `/dev/ttyMx`. Jest trochę odmienne podejście niż w Specialixie. Do karty C320Turbo można podłączyć kilka paneli po 8 portów, `x` oznacza numer panelu, `y` oznacza numer portu w panelu. W przypadku C168H `x` oznacza numer portu od 0 do 7. Należy więc przypisać odpowiednie nazwy urządzeń do podłączonych sterowników w pliku

```
/opt/szarp/<prefix>config/parcook.cfg.
```

W przypadku instalowania karty C320Turbo w komputerze z zainstalowaną dystrybucją *SZARP DEBIAN* sprawa jest jeszcze prostsza - wystarczy zainstalować pakiety `moxa-base` i `moxa-utils` z płytki instalacyjnej dystrybucji. Skrypty konfiguracyjne zadbają o ładowanie sterownika przy starcie systemu a także utworzą odpowiednie urządzenia.

### 3.2.13. Przypisywanie wielu adresów IP jednej karcie sieciowej. Aliasy IP

W rozdziale niniejszym opisane są sposoby przypisywania wielu adresów IP do jednej samej karty sieciowej. Dodatkowe adresy zwane są niekiedy *aliasami IP*.

W warstwie IP możliwe jest ustawienie więcej niż jednego adresu z więcej niż jednej podsieci IP. Na przykład możliwe jest współistnienie w jednej fizycznie sieci Ethernet kilku sieci IP: 192.168.1.0, 192.9.200.0 itd. Jeżeli dane hosty są od sieci publicznej oddzielone routerem, możliwe jest również współdzielenie przez hosty adresu publicznego i prywatnego zarazem.

### 3.2.13.1. Systemy RedHat, Mandrake, Fedora i Aurox

Dodanie kolejnego adresu IP (aliasu IP) w systemach pochodnych od RedHata można wykonać ręcznie wywołując następujące komendy (adresy IP są tu przykładowe):

```
ifconfig eth0:0 213.186.95.70 netmask 255.255.255.248
route add default gw 213.186.95.65
```

Pierwsza komenda dodaje alias, druga ustawia domyślną bramkę dla naszego hosta.

Jeżeli chcielibyśmy ustawić alias IP (ustawienia jak wyżej) w systemie na stałe, to musimy wykonać następujące operacje:

- Kopiujemy plik `/etc/sysconfig/network-scripts/ifcfg-eth0` do `/etc/sysconfig/network-scripts/ifcfg-eth0:0`
- W pliku `/etc/sysconfig/network-scripts/ifcfg-eth0:0` zmieniamy wartość zmiennych: `DEVICE` z `eth0` na `eth0:0`, `IPADDR` ustawiamy jako `213.186.95.70` oraz `NETMASK` jako `255.255.255.248`.
- W pliku `/etc/sysconfig/network` dopisujemy linię:

```
GATEWAY=213.186.95.65
```

Testujemy czy działa. Przeładowujemy sieć `/etc/rc.d/init.d/network restart`, a następnie wywołujemy komendę **ifconfig** i sprawdzamy czy obecne jest urządzenie `eth0:0`.

### 3.2.13.2. System Debian

W systemie Debian należy tylko dodać odpowiedni interfejs do pliku `/etc/network/interfaces` (konfiguracja jak dla przykładu powyżej):

```
iface eth0:0 inet static
address 213.186.95.70
netmask 255.255.255.248
broadcast 213.186.95.71
gateway 213.186.95.65
```

Po wprowadzeniu zmian należy sieć zrestartować, np. wywołując komendę `/etc/init.d/networking restart`. I sprawdzić programem **ifconfig** czy nowy interfejs (urządzenie `eth0:0`) został dodany.

## 3.2.14. Konfiguracja sieci bezprzewodowych WLAN

### 3.2.14.1. Wstęp

Sieci radiowe WLAN (Wireless Local Area Network) są coraz popularniejszą alternatywą dla tradycyjnych sieci lokalnych. Umożliwiają łączność na odległość od kilkuset metrów nawet do kilku kilometrów (z użyciem specjalnej anteny), bez potrzeby układania kabli, za pomocą fal radiowych.

Urządzenia działają w paśmie 2.4 GHz, zgodnie ze standardem IEEE 802.11. Kolejne wersje tego standardu określają coraz większe przepustowości łącza - wersja 802.11b zapewnia 11Mbit/s, 802.11g 22Mbit/s, a obecnie (maj 2006) dopracowywany jest standard 802.11n obsługujący transfer na poziomie przekraczającym 500 Mbit/s. Używanie wyżej wymienionego pasma nie wymaga w większości krajów świata (w tym w Polsce) żadnych zezwoleń i koncesji.

Możliwa jest zarówno współpraca dwóch (lub więcej) urządzeń w trybie 'peer-to-peer', jak i tworzenie sieci z tzw. punktem dostępowym, czyli centralną jednostką, z którą łączą się wszystkie urządzenia klienckie, ona zaś sama często pełni funkcję switch'a łączącego sieć radiową z jakąś siecią "tradycyjną". W pierwszym przypadku mamy do czynienia z typem sieci *Ad-Hoc*, w drugim z *Managed* lub *Infrastructure* (używane są oba określenia).

W ramach używanej częstotliwości wyróżnianych jest zwykle kilkanaście kanałów, odpowiadających konkretnym częstotliwościom w ramach ogólnego pasma. Komunikujące się urządzenia muszą korzystać z tego samego kanału. Możliwe jest ustawienie go na stałe przy konfiguracji sieci, większość urządzeń potrafi także skanować dostępne częstotliwości w poszukiwaniu punktu dostępowego.

Kolejnym ważnym parametrem sieci jest jej identyfikator (zwykle określany skrótem *ESSID*) - jest to dowolna nazwa (tekst), która identyfikuje sieć - wszystkie komunikujące się urządzenia muszą używać tego samego identyfikatora (choć niektóre punkty dostępowe można skonfigurować, aby przyjmowały np. identyfikator 'any').

Należy zwrócić uwagę, że (zwłaszcza w połączeniu ze skanowaniem kanałów i automatycznym uzyskiwaniem identyfikatora sieci) sieć WLAN jest narażona na "nieautoryzowane" podłączanie się do niej oraz podsłuchiwanie przesyłanych danych. Pewnym zabezpieczeniem może być stosowanie szyfrowania WEP (Wireless Equivalent Privacy) będącego częścią standardu 802.11, niemniej obecne wersje uważane są za zbyt słabe, aby zapewnić bezpieczeństwo. Kwestie bezpieczeństwa sieci WLAN przekraczają jednak zakres tego dokumentu, więcej informacji można znaleźć np. w Wireless LAN Security FAQ ([http://www.iss.net/wireless/WLAN\\_FAQ.php](http://www.iss.net/wireless/WLAN_FAQ.php)).

### 3.2.14.2. Konfiguracja karty WLAN pod Linuksem

Karty sieciowe WLAN mają zwykle postać albo kart PCI, albo kart PCMCIA (w przypadku notebooków), albo modułów podłączanych przez USB. Większość producentów dołącza sterowniki tylko dla różnych wersji Windows. Niemniej większość obecnie produkowanych kart można zmusić do współpracy z Linuksem.

Producentów i modeli urządzeń jest bardzo dużo, natomiast wiele urządzeń opartych jest na tym samym chipsecie (układzie). Sterowniki pisane są nie dla konkretnych modeli urządzeń, ale dla poszczególnych chipsetów. Stąd też podstawową informacją jaką musimy uzyskać o karcie, to na jakim chipsecie jest ona oparta. Następnie należy ustalić, czy dla danego chipsetu istnieją sterowniki pod Linuksa.

Pomocne mogą to być następujące strony internetowe:

- Jean Tourrilhes Linux Wireless HOWTO ([http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html)) - zawiera obszerną listę urządzeń, wraz z uwagami o ich współpracy z Linuksem.
- <http://www.linux-wlan.org> - strona domowa projektu Linux WLAN, w ramach którego rozwijane są linuksowe sterowniki do urządzeń WLAN. W szczególności na stronie znajduje się także link do podobnej jak wyżej listy urządzeń i ich chipsetów wraz z informacją, czy są wspierane przez sterowniki tworzone w ramach projektu.

W ustaleniu chipsetu karty mogą być pomocne polecenia **lspci** i **lsusb** (dla urządzeń USB). Polecenia te wypisują informacje o zainstalowanych w komputerze urządzeniach. Szczególnie interesujący może być identyfikator PCI/USB (w postaci dwóch czterocyfrowych liczb szesnastkowych, oddzielonych dwukropkiem, np. 8086:1050, gdzie pierwsza liczba jest identyfikatorem producenta, w tym przykładzie jest to Intel, a druga - konkretnego urządzenia). Aby uzyskać identyfikator w postaci numerycznej (a nie tekstowego opisu, wyświetlanego w przypadku znanych urządzeń), należy do wyżej wymienionych poleceń dać parametr *-n*.

Po ustaleniu chipsetu należy ściągnąć odpowiednie sterowniki. Może się zdarzyć, że sterowniki będą od razu obecne w używanym przez nas jądrze, ale obecnie raczej trudno na to liczyć. Jeżeli nie znajdziemy wersji sterowników przystosowanej dokładnie do używanej dystrybucji i jądra, trzeba ściągnąć najlepiej źródła sterowników (gdyż sterowniki mają postać modułu do jądra i może być problem z ich użyciem z konkretną używaną przez nas wersją jądra) i próbować załadować moduł zgodnie z dokumentacją dołączoną do sterowników. Ten etap może różnie wyglądać zależnie od konkretnego modelu karty i sterowników. W szczególności urządzenia USB wymagają także działającego wsparcia dla USB, podobnie jest z PCMCIA. Problemy związane z ładowaniem modułu najlepiej śledzić analizując logi, a więc plik `/var/log/messages` (pod RedHat'em) lub `/var/log/kern.log` (pod Debianem). Na przykład używając sterowników z projektu `linux-wlan` należy zwrócić uwagę, jakie wersje modułów są naprawdę ładowane - zdarza się, że niektóre jądra zawierają już starsze wersje używanych sterowników i są one ładowane zamiast zainstalowanych przez nas nowych (w takim przypadku należy podpisać pliki ze starymi wersjami).

Oczywiście w przypadku zmian czy dodawania modułów warto wykonać komendę **depmod -a** (pod Debianem **update-modules**). Zakładam przy tym, że używamy dość nowej wersji jądra (wsparcie dla urządzeń WLAN pojawiło się w jądrach 2.4). W RedHat i dystrybucjach pochodnych warto utworzyć alias dla ładowanego modułu z nazwą używanego urządzenia sieciowego, dodając do pliku `/etc/modules.conf` wpis typu:

```
alias eth1 8211
```

czy

```
alias wlan0 prism2_usb
```

Umożliwi to automatyczne ładowanie modułu przez skrypty konfiguracyjne sieci. Pod Debianem moduł należy dodać do pliku `/etc/modules`.

Ostatecznym testem na poprawność załadowania modułu powinno być uruchomienie programu `iwconfig` z pakietu `wireless-tools` (obecny w nowszych dystrybucjach, w razie czego należy doinstalować z [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)). Program ten jest podobny do programu `ifconfig`, pozwala na ustawienie parametrów dotyczących interfejsów bezprzewodowych, a także wyświetla informacje o nich. Po wywołaniu bez parametrów powinien wyświetlić informacje o wszystkich dostępnych interfejsach sieciowych, sprawdzając czy są to interfejsy WLAN. Wyświetlana przez program informacja może mieć np. postać:

```
lo          no wireless extensions.

wlan0      IEEE 802.11-b  ESSID:"SSID"  Nickname:""
           Mode:Managed  Frequency:2.437GHz  Access Point: 00:02:B3:93:28:9C
           Bit Rate:11Mb/s  Tx-Power:2346 dBm
           Retry min limit:8  RTS thr:off  Fragment thr:off
           Encryption key:off
           Link Quality:32/92  Signal level:-81 dBm  Noise level:-100 dBm
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0

eth0       no wireless extensions.
```

Poszczególne parametry można ustawić za pomocą odpowiednich wywołań `ifconfig`:

```
# iwconfig wlan0 essid "MOJ_INDENTYFIKATOR"
```

```
# iwconfig wlan0 mode ad-hoc
# iwconfig wlan0 channel 2
```

Oddzielną sprawą jest automatyczne uruchamianie interfejsu podczas startu systemu. Sposób uzyskania tego różni się zależnie od dystrybucji. Dokładna instrukcja opisująca większość obecnych dystrybucji jest zawarta w pliku `DISTRIBUTIONS.txt` wchodzącym w skład pakietu `wireless-tools`. Dla RedHata w wersji 7.3 zaleca się po prostu uruchomienie narzędzia `redhat-config-network`, które posiada wsparcie dla urządzeń WLAN. Narzędzie to dodaje odpowiedni wpis do `/etc/modules.conf` oraz tworzy odpowiedni plik, np. `/etc/sysconfig/network-scripts/ifcfg-wlan0` o zawartości:

```
DEVICE=wlan0
BOOTPROTO=static
NETMASK=255.255.255.0
ONBOOT=yes
MODE=Ad-Hoc
ESSID=NAZWA
CHANNEL=2
IPADDR=192.168.2.101
```

W Debianie interfejs konfiguruje się w pliku `/etc/network/interfaces` tak jak każdy inny, z tym że należy dodać wpisy specyficzne dla sieci WLAN. Przykładowo cała sekcja może wyglądać następująco:

```
auto wlan0
iface wlan0 inet static
address 10.4.0.26
netmask 255.255.255.0
wireless-essid IDENTYFIKATOR_SIECI
wireless-mode Managed
gateway 10.4.0.1
```

Do diagnostyki połączenia do sieci bezprzewodowej przydatny może się okazać program `wavemon`, który można ściągnąć z <http://www.janmorgenstern.de/projects-software.html>. Dostarcza on na bieżąco informacji o poziomie sygnału i szumu dla połączenia. Przydatne może też być narzędzie `iwlist` z pakietu `wireless-tools`, które potrafi wyświetlić listę widocznych punktów dostępowych:

```
# iwlist wlan0 scanning
wlan0      Scan completed :
           Cell 01 - Address: 00:06:25:BC:88:3B
                        ESSID:"IDENTYFIKATOR_SIECI"
                        Protocol:IEEE 802.11b
                        Mode:Managed
                        Frequency:2.412 GHz (Channel 1)
                        Quality:0/100  Signal level:-47 dBm  Noise level:-256 dBm
                        Encryption key:off
                        Bit Rate:1 Mb/s
                        Bit Rate:2 Mb/s
                        Bit Rate:5.5 Mb/s
                        Bit Rate:11 Mb/s

#
```

### 3.2.14.3. Instalacja i konfiguracja karty OvisLink AirLive 1120PCI

Radiowa karta sieciowa OvisLink AirLive 1120PCI jest oparta na chipsecie ADM 8211, koszt karty to ok. 170 zł netto. Poza sterownikami powinniśmy mieć źródła jądra oraz zainstalowany pakiet wireless-tools w wersji co najmniej 25.

Kartę instalujemy w wolnym slotcie PCI, podłączamy antenę. Sterowniki do karty ściągamy ze strony <http://list.driverguide.com/list/company2008/index.html> - ściągamy najnowszą wersję pliku ADM8211\_src\_WERSJA.tar.gz, np. podczas pisania tego rozdziału była to wersja 105.

Rozpakowujemy i kompilujemy, czyli wykonujemy kolejno w katalogu z plikiem następujące komendy:

```
# tar xzf ADM8211_src_105.tgz
# cd ADM8211
# make
```

Otrzymujemy plik 8211.o, który kopiujemy do katalogu z modułami, zależnie od wersji jądra powinno to być coś w stylu:

```
# cp 8211.o /lib/modules/2.4.20-9/kernel/drivers/net/wireless
```

Wykonujemy komendę:

```
# depmod -a
```

Komunikatem o błędzie nie przejmujemy się - spowodowany jest on tym, że moduł 8211 nie jest rozprowadzany na licencji GPL.

Zakładam, że mamy już zwykłą kartę sieciową - interfejs eth0 i nasz interfejs radiowy ma mieć nazwę eth1. Poniżej opisuję konfigurację ręczną, można też skorzystać z narzędzia redhat-config-network.

Do pliku `/etc/modules.conf` dodajemy wpis:

```
alias eth1 8211
```

Następnie tworzymy lub modyfikujemy plik `/etc/sysconfig/network-scripts/ifcfg-eth1`:

```
DEVICE=eth1
BOOTPROTO=static
NETMASK=255.255.255.0
ONBOOT=yes
MODE=Ad-Hoc
ESSID=NAZWA
CHANNEL=2
IPADDR=192.168.2.101
```

Parametry, które można modyfikować to:

- *DEVICE* - nazwa interfejsu (musi być zgodna z nazwą edytowanego pliku).
- *NETMASK* i *IPADDR* - maska i adres sieciowy, jak dla zwykłej karty sieciowej z tym, że adres sieci wireless powinien być inny niż innych interfejsów komputera.

- *MODE* - wartość 'Ad-Hoc' jest odpowiednia, jeśli mamy kilka komputerów z kartami radiowymi komunikujących się bezpośrednio. Jeżeli mamy punkt dostępowy, ustawiamy 'Managed'.
- *CHANNEL* - określa częstotliwość, na jakiej będzie się odbywała komunikacja. Jeśli ustawimy 0, to karta będzie skanować dostępne częstotliwości w celu wybrania odpowiedniej.
- *ESSID* - unikalna nazwa identyfikująca sieć.

Inne komputery w sieci muszą mieć oczywiście inne adresy IP, ale ten sam kanał i tą samą nazwę sieci. Jeśli chodzi o ewentualną dalszą konfigurację, nasza sieć radiowa zachowuje się jak normalny Ethernet.

Po dokonaniu zmian w plikach restartujemy sieć (np. w RH komenda **service network restart**). Polecenie `ifconfig` powinno nam teraz wypisać także interfejs `eth1` z ustawionym adresem IP. Aby obejrzeć szczegóły takie jak siła sygnału, przepustowość łącza itp., posługujemy się programem `iwconfig` z pakietu `wireless-tools`.

#### 3.2.14.4. Instalacja kart WLAN opartych na chipsecie Atheros Communications AR5212

Opis został przygotowany dla karty Planet WL-8310, ale powinien sprawdzić się dla wszystkich kart z chipem AR5212.

Do uruchomienia karty potrzebne są moduły dostępne w pakiecie `madwifi`. Można je pobrać ze strony: <http://madwifi.org> (<http://madwifi.org/>) lub w postaci pakietu `madwifi-sources` z repozytorium Debiana. Przydatny może być również pakiet `madwifi-tools`.

W systemie Debian instalację modułu przeprowadza się wydając polecenie:

```
module-assistant a-i madwifi
```

Powinny załadować się moduły: `wlan`, `ath_hal`, `ath_pci` i `ath_rate_sample`. Można to sprawdzić wydając polecenie: **lsmod**. Jeżeli nie ma ich na liście, należy załadować je ręcznie przy pomocy komendy **modprobe**. Wystarczy załadować moduł `ath_pci`, reszta powinna zostać załadowana automatycznie. Po restarcie systemu wszystkie potrzebne moduły powinny załadować się automatycznie.

Konfiguracja karty nie wymaga szczególnych zabiegów. Należy postępować zgodnie z instrukcją w rozdziale: Konfiguracja karty WLAN pod Linuksem.

#### 3.2.14.5. Konfiguracja kart bezprzewodowych z wykorzystaniem sterowników Windows

System Windows posiada dość dobrze wydzieloną warstwę odpowiedzialną za komunikację z urządzeniami sieciowymi, obsługiwaną przez API o nazwie Network Driver Interface Specification (NDIS). Możliwa jest emulacja tej warstwy pod Linuksem, dzięki czemu można uruchamiać niektóre urządzenia, korzystając ze sterowników przeznaczonych dla systemu Windows. Istnieje kilka projektów w ramach których powstaje oprogramowanie do emulacji NDIS, jednym z nich jest `ndiswrapper` (<http://ndiswrapper.sf.net>). Ten rozdział opisuje konfigurację kart bezprzewodowych z wykorzystaniem `ndiswrappera`, opisany sposób został przetestowany z kartami na chipsecie Realtek 8180 (uwaga - na niektórych komputerach karta nie działała, na innych ruszyła) oraz z kartą D-Link DWL-520+ (chipset Texas Instruments ACX 100, identyfikator PCI 104c:8400).

**Notatka:** Karty na chipsecie TI ACX100, w tym D-Link DWL-520+ działają także na sterownikach tworzonych w ramach projektu `acx100.sf.net` (<http://acx100.sf.net>).

Pierwszym elementem jest ściągnięcie ndiswrappera - narzędzia i modułu odpowiedniego dla używanego jądra. Warto także skorzystać z jak najnowszego jądra (zbyt stare nie będą wspierane przez ndiswrappera lub sprzęt może nie działać).

W przypadku Debiana należy upewnić się, że mamy pakiet z nagłówkami dla używanego jądra, a następnie ściągnąć narzędzia i źródła do modułów:

```
# apt-get install ndiswrapper-utils ndiswrapper-source wireless-tools
```

Ściągniętych zapewne zostanie kilka dodatkowych pakietów, w tym module-assistant.

Kolejnym krokiem jest zbudowanie modułów, co wykonujemy komendą:

```
# module-assistant auto-install ndiswrapper
```

Po kompilacji powinniśmy otrzymać automatycznie zainstalowane moduły dla aktualnie używanej wersji jądra.

Kolejnym krokiem jest ustalenie z jakim urządzeniem mamy do czynienia (**lspci** i **lspci -n**), oraz sprawdzenie na liście kart, jakiego drivera powinniśmy użyć, oraz skąd go ściągnąć. Zwykle dobre efekty uzyskamy korzystając ze sterowników umieszczonych na płycie dołączonej do karty - mamy przynajmniej pewność, że sterowniki będą pasować do zakupionej wersji karty. Jeżeli do wyboru są sterowniki dla różnych wersji Windows, warto wypróbować te dla Windows XP, 2000 lub NT (najlepiej w tej kolejności). Sterowniki mogą mieć postać archiwum zip (możemy wtedy użyć do rozpakowania programu **unzip**), czasami jednak są dostarczane w postaci pliku wykonywalnego. Jeżeli próba użycia na tym pliku narzędzi do rozpakowywania archiwów zip zawiedzie, możemy być zmuszeni do uruchomienia programu pod Windows, aby wydobyć interesujące nas pliki.

Te interesujące nas pliki to plik z rozszerzeniem *INF* (zawiera między innymi listę właściwych plików ze sterownikiem) oraz pliki z samym sterownikiem, zwykle o rozszerzeniach *SYS*, *BIN* lub *DAT*. Wszystkie te pliki powinniśmy umieścić w jednym katalogu, po czym wywołać komendę instalacji sterownika, podając jako argument plik *INF*, np.

```
# ndiswrapper -i RTL8180.INF
```

Możemy teraz sprawdzić czy sterownik zainstalował się poprawnie:

```
# ndiswrapper -l
Installed ndis drivers:
net8180 driver present, hardware present
#
```

Powinniśmy otrzymać informację, jak powyżej, o obecności zarówno sterownika, jak i obsługiwanego sprzętu.

Następny krok to załadowanie modułu:

```
# modprobe ndiswrapper
```

W logach sprawdzamy czy wszystko poszło poprawnie:

```
# tail /var/log/kern.log
Mar 22 12:04:50 btw7 kernel: ndiswrapper version 1.0rc2 loaded (preempt=yes,smp=no)
Mar 22 12:04:50 btw7 kernel: PCI: Found IRQ 9 for device 0000:00:05.0
Mar 22 12:04:50 btw7 kernel: ndiswrapper: using irq 9
Mar 22 12:04:50 btw7 kernel: wlan0: ndiswrapper ethernet device 00:50:fc:d1:05:a4 \
using driver net8180
```

```
Mar 22 12:04:50 btw7 kernel: wlan0: encryption modes supported: WEP, WPA with TKIP, \
WPA with AES/CCMP
Mar 22 12:04:50 btw7 kernel: ndiswrapper: driver net8180 (Realtek,10/07/2004, \
5.173.1007.2004) added
```

Przykładowe komunikaty donoszą o pomyślnym wykryciu i zainstalowaniu sterownika dla karty. Jeżeli na tym etapie napotkamy problemy, to mogą być spowodowane złą wersją sterowników (niedostosowaną do sprzętu), za starą wersją jądra lub po prostu oznaczać, że karta nie chce pracować pod kontrolą ndiswrappera. Jeżeli zaś wszystko wydaje się OK, to należy zapewnić, że moduł ndiswrapper będzie ładowany przy starcie systemu (w Debianie umieszczamy nazwę modułu w pliku `/etc/modules`).

Ostatni krok to konfiguracja interfejsu sieciowego (zwykle wlan0). Dokonujemy tego standardowo za pomocą narzędzi **ifconfig** i **iwconfig** lub w sposób specyficzny dla danej dystrybucji, np. w Debianie umieszczając w pliku `/etc/network/interfaces` następujące linie:

```
auto wlan0
iface wlan0 inet static
address 192.168.3.47
netmask 255.255.255.0
broadcast 192.168.3.255
gateway 192.168.3.1
wireless-essid IDENTYFIKATOR_SIECI
wireless-mode Managed
wireless-key restricted KLUCZ_WEP
```

Po podaniu komendy **ifup wlan0** powinniśmy mieć działającą sieć bezprzewodową. Sprawdzamy to za pomocą polecenia **iwconfig**, które powinno wyświetlić informacje w stylu:

```
wlan0      IEEE 802.11b  ESSID:"IDENTYFIKATOR SIECI"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 00:06:25:BC:88:3B
          Bit Rate:11 Mb/s   Tx-Power:20 dBm   Sensitivity=0/3
          RTS thr:2432 B   Fragment thr:2432 B
          Encryption key:off
          Power Management:off
          Link Quality:100/100  Signal level:-53 dBm  Noise level:-256 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

Uwaga - interfejs może nie być prawidłowo skonfigurowany dopóki nie znajdzie się w zasięgu punktu dostępowego.

### 3.2.15. Tworzenie połączeń sieciowych w oparciu o protokoły point-to-point

Rozdział niniejszy dotyczy zagadnień związanych łączeniem się przy pomocy protokołów point-to-point z użyciem protokołu PPP. Wykorzystywany tu program **pppd** (czyli „demon PPP”) jest najbardziej popularnym (czy nie jedynym?) narzędziem tego typu. Połączenia PPP stosowane są zarówno w klasycznych modemowych połączeniach telefonicznych (dial-up), jak i w przypadku innych metod dostępu (xDSL, ISDN, dostęp radiowy).

Rozdział nie jest jednolity: opisano tu różne podejścia do konfiguracji demona **pppd**. Ich zróżnicowanie wynika w głównej mierze z tego, że program jest cały czas rozwijany, zmieniają się jego opcje i

możliwości. Obecnie najwygodniejszą metodą zarządzania konfiguracją i zachowaniem programu **pppd** jest wywoływanie go w następujący sposób:

```
pppd call <konfiguracja>
```

Wywołanie takie spowoduje rozpoczęcie połączenia opisanego w pliku `/etc/ppp/peers/<konfiguracja>`. Zastosowanie tego mechanizmu pozwala wyodrębnić części wspólne oraz części specyficzne dla różnych połączeń (ma to dużą wagę szczególnie w dzisiejszych czasach, kiedy to jedna maszyna często wykorzystuje wiele różnych metod dostępu do sieci). Mechanizm „pppd call” wykorzystano w opisie konfiguracji połączenia z siecią przez Sferię (Sekcja 3.2.15.12) oraz opisu konfiguracji systemów GPRS (Sekcja 3.2.15.14). W systemie Debian standardowo obecna jest nakładka upraszczająca korzystanie z tego mechanizmu - skrypty **pon** i **poff**: pierwszy z nich rozpoczyna połączenie i wywołany jest następująco:

```
pon <konfiguracja>
```

Drugi ze skryptów połączenie zamyka.

### 3.2.15.1. Instalacja modemu wewnętrznego PCI z chipem Conexant

Jak to opisano w rozdziale Sekcja 2.1.13, zalecanym rozwiązaniem dla Linuksa jest modem zewnętrzny. Jego instalacja sprowadza się do podłączenia modemu do komputera. Sprawa komplikuje się poważnie jeżeli mamy do czynienia z modemem wewnętrznym, nie będącym "prawdziwym" modemem, a tzw. soft-modemem, wymagającym wsparcia oprogramowania działającego na komputerze. Większość tego typu urządzeń będzie działać tylko pod Windows, ale są chlubne wyjątki.

Należą do nich modemy oparte na układach kanadyjskiej firmy Conexant, takie jak np. Pentagram Hex2 56I. Modem ma postać karty PCI, którą należy włożyć w wolny slot na płycie głównej. Posiada dwa wyjścia na skrętkę telefoniczną, do których w typowy dla modemów sposób podłącza się kabel do gniazdka telefonicznego (wyjście oznaczone jako LINE) i do telefonu (oznaczenie PHONE).

Kolejnym krokiem jest instalacja sterowników. Sterowniki można znaleźć na stronie <http://www.linuxant.com/drivers>. Sterowniki tworzone są przez firmę Linuksant, nie mającą nic wspólnego z producentem chipsetu, czyli firmą Conexant. Stąd też Linuksant, aby zdobyć finanse na rozwój sterowników zmienił licencję na częściowo płatną. Wersja bezpłatna umożliwia łączenie z prędkościami do 14400 bps, nie obsługuje także funkcji głosowych. Aby wykorzystać pełne możliwości modemu, należy zakupić licencję w cenie około 15 \$ za jeden modem.

Dostępne są sterowniki dla dwóch typów modemów - HSF (software modems) i HCF (controlled modems). Pierwsze z nich są tańsze, należy do nich np. wspomniany modem Hex2, do drugiego typu należy np. Pentagram Diabło 56I. Jeżeli nie wiemy, jakiego typu mamy modem, nie należy się przejmować tylko spróbować instalować sterowniki po kolei dla obu typów - jeżeli źle trafimy, instalator poinformuje nas, że mamy modem innego typu. Typ modemu można także sprawdzić za pomocą dostępnego na stronach Linuksant skryptu. Większość współczesnych jąder Linuksa potrafi także poprawnie zidentyfikować typ modemu. Wywołanie komendy **lspci** dla modemu Hex2 może dać w wyniku np. linię postaci:

```
0000:00:08.0 Communication controller: Conexant HSF 56k HSF-i Modem (rev 01)
```

Najłatwiej jest wybrać do ściągnięcia wersję binarną sterownika - są one dostępne dla części popularniejszych dystrybucji Linuksa. Należy przy tym zwrócić uwagę na wybranie wersji odpowiedniej dla procesora, jaki jest zainstalowany w komputerze. W szczególności istnieje oddzielna wersja dla

procesorów Athlon - sterowniki wykorzystują specyficzne dla procesorów instrukcje MMX, SSE czy 3DNow!. Sterowniki muszą też być dopasowane do konkretnej wersji jądra (można ją sprawdzić przez wydanie komendy **uname -r**).

Po ściągnięciu odpowiedniego pakietu RPM (w przypadku RedHata) instalujemy go przez wydanie komendy **rpm -i <nazwapakietu>**. Jeżeli wszystko poszło dobrze, powinniśmy mieć skonfigurowany i działający modem. Link `/dev/modem` powinien wskazywać na urządzenie `/dev/ttySHSF0` lub `/dev/ttySHCF0` zależnie od typu modemu (HSF czy HCF).

Pod Debianem należy ściągnąć paczkę 'deb' ze źródłami. Musimy mieć zainstalowany pakiet z plikami nagłówkowymi jądra dla aktualnie używanej wersji jądra. Przykładowo, jeżeli komenda 'uname -r' wypisuje '2.6-11-1-686', to musimy zainstalować pakiet `kernel-headers-2.6.11-1-686` (od jądra 2.6.12 jest to pakiet o przykładowej nazwie `linux-headers-2.6.14-1-686`). Następnie za pomocą komendy 'dpkg -i' instalujemy ściągniętą paczkę ze sterownikami. Skrypt instalacyjny powinien sam wykryć katalog z konfiguracją jądra, w naszym przykładzie byłyby to `/lib/modules/2.6.11-1-686/build`. Akceptujemy zaproponowaną lokalizację, podajemy swój adres e-mail i klucz licencji (FREE - chyba, że zakupiliśmy wersję pełną).

Jeżeli modem nie chce odbierać połączeń przychodzących, powodem może być zły string inicjalizujący wysyłany przez mgetty. Prawidłowy string inicjalizacyjny można podać albo w pliku `/etc/mgetty+sendfax/mgetty.config` (w Debianie `/etc/mgetty/mgetty.config`) przez dodanie linii:

```
init-chat "" \dATQ0V1H0[0d] OK ATS0=0Q0&D2&C1 OK AT&K3 OK ATLO OK
```

albo przez podanie parametru do programu mgetty w pliku `/etc/inittab`. Wówczas odpowiednia linia może mieć postać:

```
m:345:respawn:/sbin/mgetty -x 0 -m '"" \dATQ0V1H0[0d] OK ATS0=0Q0&D2&C1 OK AT&K3 OK \
ATLO OK' /dev/modem
```

(w Debianie zamiast 'm:345' podajemy 'm:23', bo domyślny 'runlevel' to 2). Zobacz też rozdział o konfigurowaniu połączeń przychodzących - Sekcja 3.2.15.2.

**Notatka:** Starsze wersje sterowników Conexanta źle współpracowały z jądrami z serii 2.6. Nie można się było dodzwonić do komputera z modemem Conexant (np. sesja zawieszała się po podaniu nazwy użytkownika). Rozwiązaniem było uruchomienie na komputerze odbierającym połączenia jądra z serii 2.4. Nowsze wersje sterowników nie kompilują się w ogóle pod jądrem 2.4 (przynajmniej te z pakietu .deb), za to pod 2.6 działają poprawnie.

### 3.2.15.2. Konfiguracja modemu do przyjmowania połączeń przychodzących

Ten rozdział zawiera opis konfiguracji modemu do obsługi połączeń przychodzących. Konfiguracja serwera / klienta ppp znajduje się w następnych rozdziałach.

Co trzeba zrobić? Po kolei:

- podłączyć modem do komputera.
- Przygotować drugi komputer z działającym modemem i przetestowaną możliwością dzwonienia do ISP (np. do TPSA: 0w202122): w przypadku komputera z dos'em lub windows'ami powinniśmy mieć

zainstalowany program term95 (rozprowadzany razem z NC) lub inny odpowiedni program komunikacyjny. W przypadku komputera z Linuxem wystarczy nam minicom.

- Do obu komputerów (modemów) podłączyć centralkę telefoniczną (centralka ma 4 numery: 1,2,3,4).
- Zalogować się jako root.
- Zrobić linka (o ile jeszcze nie istnieje) z portu, do którego jest podłączony modem (/dev/ttyS0 = COM1, /dev/ttyS1 = COM2) do /dev/modem:

```
ln -s /dev/ttyS0 /dev/modem
```

- Teraz należy zainstalować program jakiś program z rodziny getty otwierający/czuwający na porcie szeregowym. Tu opisano dwa takie programy **mgetty** i **ugetty**. Najpopularniejszym i polecanym tutaj rozwiązaniem jest **mgetty**.

**mgetty**. Jest on prostszy w konfiguracji, powszechnie używany, umożliwia nawiązywanie połączeń wychodzących w trakcie czuwania (używa plików blokujących).

- W /etc/inittab na końcu dodać linię:

```
m:345:respawn:/sbin/mgetty -x 0 /dev/modem
lub pod Debianem (który działa domyślnie w runlevel'u 2):
```

```
m:23:respawn:/sbin/mgetty -x 0 /dev/modem
```

opcja -x 0 oznacza brak logowania; standardowo mgetty zapisuje logi do pliku /var/log/mgetty.log.modem; standardowo tych logów jest dużo, w przypadku wyłączenia zasilania modemu zewnętrznych są znacznie większe - ok. 100kB-300kB dziennie.

- Teraz uruchamiamy mgetty:

```
kill -1 1
ps ax | grep mgetty
```

i patrzymy czy jest linia z mgetty. Jak jest to znaczy że działa, jeśli nie działa to musieliśmy coś pomylić (mgetty na prawdę nie wymaga wielkiej filozofii!)

**ugetty**. Jest standardowo obecny w RedHacie, jeśli go nie ma to należy zainstalować go wraz z pakietem **getty\_ps**. Wymaga dodatkowych plików konfiguracyjnych, ale może być jeszcze elastyczniej konfigurowalny niż mgetty. Pozwala na bezpieczne nawiązywanie połączeń wychodzących. Podobnie jak mgetty stosuje standard blokowania UUCP, czyli pliki /var/lock/LCK.<nazwa urządzenia>, standard używany przez minicoma, kppp oraz pppd (gdy podamy opcję lock), itp. Co robimy:

- Tworzymy plik conf.ugetty.modem z zawartością:

```
#ALTLOCK=ttyS0
TIMEOUT=60
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
DELAY=1
#DEBUG=010
```

Pierwsza linia (tu zakomentowana) oznacza, że alternatywną nazwą dla naszego urządzenia będzie ttyS0. Jest to przydatne, gdy zdarza nam się odwoływać bezpośrednio do urządzeń /dev/ttySx a niekoniecznie przez linka /dev/modem.

- W /etc/inittab na końcu dodajemy linię:

```
uu:345:respawn:/sbin/ugetty modem
```

Uwaga! Jako parametr do ugetty podajemy nazwę urządzenia bez poprzedzającego katalogu /dev/. Jest to jedna z różnic między mgetty i ugetty.

- Teraz niech zadziała uugetty:

```
init q
ps ax | grep uugetty
```

i patrzemy czy jest linia z uugetty. Jak jest to znaczy że działa. Jeśli nie to mogliśmy zrobić błąd w pliku konfiguracyjnym albo jest jakiś inny problem. Przy usuwaniu problemów pomocny jest parametr DEBUG w pliku `/etc/conf.uugetty.<nazwa_urządzenia>`. Flaga 010 (jak w proponowanym pliku powyżej) wystarcza do znalezienia większości błędów. Najmocniejszą flagą jest 777. Logi generowane przez uugetty są przekazywane do sysloga.

- *Testujemy*

Na drugim komputerze uruchomiamy minicom (dla Linuksa) lub inny program (np. dla DOS-a może to być term95) i dzwoniemy do naszego komputera:

```
minicom
```

W przypadku niektórych modemów mogą występować problemy, jeśli na maszynie, na której uruchomiliśmy minicom'a, działa już mgetty. Wtedy przed próbą połączenia powinniśmy wydać w minicomie polecenie:

```
atz
```

W minicomie następnie wydajemy polecenie:

```
atdt "numer"
```

gdzie "numer" oznacza 1, 2, 3, lub 4 w zależności od tego, do którego numeru centralki podpięliśmy serwer.

Po krótkiej chwili (komutacja łączy, popiskiwanie modemów) powinniśmy po stronie klienckiej (dzwoniącej) dostać zgłoszenie się zdalnego systemu (login prompt): login: Teraz możemy się zalogować i od tej chwili powinniśmy być w stanie wydawać polecenia. Oznacza to, że wszystko jest O.K. Możemy już się rozłączyć (np. komendą exit).

### 3.2.15.3. Konfiguracja serwera ppp (połączenia przychodzące)

Ten rozdział opisuje sposób konfiguracji serwera ppp. Zakłada, że udało się poprawnie skonfigurować modem (np. według instrukcji z poprzedniego rozdziału). Zakłada również, że mamy możliwość przetestowania połączenia (na przykład przy pomocy centralki telefonicznej - tak jak w poprzednim rozdziale).

Co robimy? Po kolei:

- W pliku: `/etc/ppp/options` dla Red Hat'a wpisujemy:

```
115200
defaultroute
lock
noauth
/dev/modem
```

a dla Debiana:

```
asynmap 0
noauth
crtsets
lock
hide-password
```

```

modem
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
noipx
/dev/modem
defaultroute
replacedefaultroute

```

W systemie Red Hat 7.x dla prawidłowego działania konieczne może być usunięcie opcji `noauth`. Czasem także wymagane jest zmniejszenie prędkości portu z 115200 na 38400. Uwaga! Plik `/etc/ppp/options` jest podstawowym plikiem konfiguracyjnym dla `pppd`. Można tam ustawiać wiele różnych opcji, charakterystycznych dla poszczególnych systemów. Na przykład w przypadku modemu GSM Siemens M20 w sieci Era, opisanego w rozdziale Sekcja 3.2.15.6, należało dodać opcję:

```
lcp-max-configure 30
```

..ewentualnie można ją nawet zwiększyć.

- Tworzymy plik `/etc/ppp/options.server` i wpisujemy w nim:

```

-detach
netmask 255.255.254.0
proxyarp
192.168.8.1:192.168.8.8

```

Ostatnia linia oznacza numery IP przydzielane połączeniu, odpowiednio po stronie serwera i klienta. Można ustawić je na inne, niekolidujące z numeracją innych dostępnych sieci.

- Tworzymy plik `/etc/ppp/ppplogin` z zawartością:

```

#!/bin/bash
msg n
stty -echo
exec /usr/sbin/pppd file /etc/ppp/options.server -detach modem crtscts

```

- Zmieniamy prawa do pliku `/etc/ppp/ppplogin`

```
chmod +x /etc/ppp/ppplogin
```

- Dodajemy użytkownika `ppp` bez hasła:

```

adduser -d /tmp -s /etc/ppp/ppplogin ppp
passwd -d ppp

```

Pod Debianem komendy do wpisania są nieco inne:

```

adduser --home /tmp --shell /etc/ppp/ppplogin ppp
passwd -d ppp

```

Jeżeli system nie pozwala na logowanie bez hasła przez łączy szeregowy, należy oczywiście ustawić jakieś hasło, standardowo `'ppp'`.

- Zmieniamy prawa do pliku `/usr/sbin/pppd`:

```
chmod +s /usr/sbin/pppd
```

Ponadto należy zadbać, by użytkownik `ppp` posiadał prawo wykonania pliku `/usr/sbin/pppd`. Np. w Debianie można to osiągnąć dodając użytkownika `ppp` do grupy `dip`.

- *Testujemy*

- Łączymy się za pomocą `minicom` ze zdalnym komputerem (opis tej procedury znajduje się w poprzednim rozdziale). Logujemy się na użytkownika `ppp`.
- Wychodzimy z `minicom` bez resetu modemu: `Ctrl-A Q`
- Nadal na tym samym komputerze uruchamiamy:

```
/usr/sbin/pppd -detach lock noauth /dev/modem
```

komputery powinny się połączyć i podać ip oraz remote-ip.

- W przypadku testu przez Internet powinniśmy dodać do pliku `/etc/resolv.conf` serwery nazw:

```
nameserver 194.204.159.1
nameserver 194.204.152.34
```

- Na innej konsoli należy sprawdzić programem **ifconfig** czy jest zarejestrowany interfejs `ppp0`. Jeśli połączenie PPP zostało nawiązane, to na wyjściu z programu **ifconfig** powinna się pojawić sekcja jak niżej (z dokładnością do niektórych wartości liczbowych):

```
ppp0      Link encap:Point-to-Point Protocol
inet addr:192.168.8.8  P-t-P:192.168.8.1  Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
RX packets:5 errors:1 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:30
RX bytes:72 (72.0 b)  TX bytes:78 (78.0 b)
```

Uwaga! Zauważmy, że adresem IP przydzielonym naszemu hostowi jest ten oznaczony jako „inet addr” (na przykładzie powyżej jest to 192.168.8.8), natomiast adres serwera to „P-t-P” (tu 192.168.8.1). Naturalnie, jeśli chcemy się zalogować na hosta „po drugiej stronie” to powinniśmy wpisać adres „P-t-P”.

- Możemy dodatkowo przetestować czy poprawnie działają programy oparte o protokół TCP:

```
ping <ip-drugiego-komputera>
telnet <ip-drugiego-komputera>
```

#### 3.2.15.4. Klient ppp (połączenia wychodzące) przy użyciu **kppp**

To jest opis konfiguracji ppp po stronie klienta tak, abyśmy mogli np. łączyć się z internetem. (Warto przeczytać także rozdziały Sekcja 3.2.15.2 i Sekcja 3.2.15.3).

- Upewnij się, że modem jest dobrze podłączony i skonfigurowany. Najlepiej sprawdź to na tym samym komputerze pod innym systemem operacyjnym (dos, windows). Jeśli na twoim komputerze nie ma innego systemu niż linux, pominiemy ten punkt.

- W Linuksie porty szeregowy nazywają się inaczej niż w dosie i windowsach:

```
com1 = /dev/ttyS0
com2 = /dev/ttyS1
com3 = /dev/ttyS2
com4 = /dev/ttyS3
itd.
```

- Dalej zakładam, że modem jest podłączony do `/dev/ttyS1`. Jeśli tak nie jest, zmień wszystkie wystąpienia `ttyS1` na odpowiednie inne.

- Wersja `kppp`, której ja używam, wymaga znajomości przez użytkownika hasła `root'a`. Jest to niewygodne (w rzeczywistości dzwonić może tylko administrator systemu), ale bezpieczne rozwiązanie. Starsze wersje nie miały tego ograniczenia. Jeśli twoja wersja `kppp` pyta cię o hasło `root'a` przed rozpoczęciem działania, i jest to dla ciebie uciążliwe lub niedopuszczalne rozwiązanie (bo np. chciałbyś dać możliwość dzwonięcia innym, ale nie dawać im hasła `root'a`), spróbuj użyć starszej wersji `kppp` lub innych podobnych programów, które najprawdopodobniej znajdują się na twoim

komputerze: gnome-ppp, rp3. Ja osobiście używam rp3. Konfiguracja tych programów jest podobna do kppp i nie będę jej opisywał.

- Jeśli używasz starszej wersji kppp i chcesz, aby mogli z niej korzystać użytkownicy inni niż root, zmień prawa (zalogowany jako root):

```
chmod a+r+w /dev/modem
chmod a+r+w /dev/ttyS1
chmod +s /usr/sbin/pppd
chmod +s /usr/bin/kppp
```

- Test modemu: napisz:

```
minicom
```

Powinien uruchomić się program, napisać: Initializing Modem, i na końcu ok. Sprawdźmy czy komunikujemy się z modemem, napisz:

```
at
```

powinieneś dostać odpowiedź ok. Jeśli jest inaczej, to albo modem jest źle podłączony, albo źle skonfigurowany. Z programu minicom zadzwonimy do TP SA.

- Napisz:

```
atdt0w202122
```

- Powinieneś usłyszeć wybieranie numeru i charakterystyczne pischczenie (oczywiście numer może być zajęty i wtedy operację należałoby powtórzyć).
- Powinieneś otrzymać prompt Username:
- Rozłącz się: Ctrl + A , X (pomoc Ctrl + A, Z).
- kppp jest dostępne w KDE z menu Internet->Kppp. Uruchom je.
- Skonfiguruj kppp:
  - w kppp wybierz Opcje... pojawi się nowe okno: Konfiguracja kppp.
  - W oknie Konfiguracja kppp wybierz zakładkę Połączenia i dodaj nowe połączenie: Nowe... pojawi się nowe okno: Nowe połączenie.
  - W oknie Nowe połączenie ustaw:
    - zakładka Dzwonienie:
 

```
nazwa: tpsa
numer telefonu: 0w202122
identyfikacja: oparta na terminalu
zapamiętaj hasło: tak
Zmiana opcji pppd: w polu Opcja wpisać: noauth i nacisnąć Dodaj
pozostałe puste.
```
    - Zakładka Adres IP:
 

```
zmienny adres: tak
pozostałe puste.
```
    - Zakładka DNS:
 

```
lista serwerów: 194.204.159.1
194.204.152.34
pozostałe puste.
```
    - Zakładka Router:

```
domyślny router: tak
ustaw IP routera: nie
przypisz domyślny routing: tak
```

- **Zakładka Skrypt logowania:**

```
Expect Password:ppp
Expect name:
Send ppp
Expect word:
Send ppp
```

- **zakładka koszty - można nie wypełniać.**

- **W oknie Konfiguracja kppp ustaw:**

- **zakładka Port:**

```
port szeregowy: /dev/modem
kontrola przepływu: CRTSCTS
znak końca linii: CR/LF
szybkość portu: 115200 (najczęściej, jeśli masz stary
komputer podziel tą wartość przez 2 lub 4)
używaj pliku blokującego: tak
limit czasu modemu: 60
```

- **zakładka Modem:**

```
czekaj, gdy zajęty: 0
reszty można użyć do testowania komunikacji z modemem
zakładka PPP:
limit czasu pppd: 30
pokaż na panelu po połączeniu: tak
dzwon ponownie po rozłączeniu: tak
pokaż czas połączenia na pasku zadań: tak
rozłącz przy zamknięciu X-serwera: tak
zakończ przy rozłączeniu: nie
minimalizuj okno w czasie połączenia: tak
```

- **Pozostałe zakładki:**

wg. uznania.

Wybierz ok, aby zachować ustawienia.

- **W głównym oknie kppp wybierz:**

```
połącz z: tpsa
użytkownik: ppp
hasło: ppp (wyświetlane jako gwiazdki)
pokaż podgląd: tak
```

- **Wybierz: Połącz**

Powinieneś połączyć się teraz z serwerem TP SA i mieć dostęp do internetu.

- **Jeśli coś nie zadziało, warto przeczytać pomoc do kppp ( w głównym oknie naciśnij przycisk Pomoc), lub PPP-HOWTO (plik: /usr/doc/HOWTO/PPP-HOWTO, to jest dość stare HOWTO i opisuje sposób łączenia się z internetem za pomocą skryptów. Czasami takie skrypty są bardzo wygodne i warto wiedzieć jak działają).**

### 3.2.15.5. Klient ppp (połączenia wychodzące) oparty o skrypty

Jest to sposób na uruchomienie ppp po stronie klienta równoważny opisanemu w rozdziale Sekcja 3.2.15.4. Skonfigurowanie skryptów w opisany tu sposób „nie gryzie” się z konfiguracją kppp, mogą być stosowane wymiennie. Zakłada się, że modem jest poprawnie skonfigurowany jako /dev/modem.

Skrypty są mniej wygodne i trudniejsze do konfiguracji niż klienci X/KDE, lecz mogą być stosowane w sytuacji padniętych X-ów, lub niezainstalowanych klientów X-owych (np. kppp, gppp).

Żeby uruchomić dialupa ppp, dzwoniącego za tępsę, należy stworzyć 3 pliki (istniejące pliki o tych nazwach można wyrzucić lub zbackupować):

- **Plik:** /etc/ppp/ppp-on o następującej treści:

```
#!/bin/sh

TELEPHONE=0202122 # The telephone number for the connection
ACCOUNT=ppp # The account name for logon (as in 'George Burns')
PASSWORD=ppp # The password for this account (and 'Gracie Allen')
LOCAL_IP=0.0.0.0 # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP=0.0.0.0 # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.255.0 # The proper netmask if needed

export TELEPHONE ACCOUNT PASSWORD

DIALER_SCRIPT=/etc/ppp/redialer

exec /usr/sbin/pppd noauth debug lock modem crtscts /dev/modem 57600 \
asynmap 20A0000 escape FF kdebug 24 $LOCAL_IP:$REMOTE_IP \
noipdefault netmask $NETMASK defaultroute connect $DIALER_SCRIPT
```

- **Plik:** /etc/ppp/ppp-off o następującej treści:

```
#!/bin/sh

if [ "$1" = "" ]; then
DEVICE=ppp0
else
DEVICE=$1
fi

if [ -r /var/run/$DEVICE.pid ]; then
kill -INT `cat /var/run/$DEVICE.pid`

if [ ! "$?" = "0" ]; then
rm -f /var/run/$DEVICE.pid
echo "ERROR: Removed stale pid file"
exit 1
fi

echo "PPP link to $DEVICE terminated."
exit 0
fi

echo "ERROR: PPP link is not active on $DEVICE"
exit 1
```

- **Oraz plik:** /etc/ppp/redialer o następującej treści:

```

#!/bin/sh

MAX_ATTEMPTS=20
SLEEP_DELAY=10s

function initialize
{
    chat -v TIMEOUT 3 " AT 'OK-+++c-OK'
    return
}

function callnumber
{
    chat -v \
    TIMEOUT 3 \
    ABORT '\nBUSY\r' \
    ABORT '\nNO ANSWER\r' \
    ABORT '\nRINGING\r\n\r\nRINGING\r' \
    " ATl1 \
    " ATM1 \
    " \rAT \
    'OK-+++c-OK' ATH0 \
    TIMEOUT 250 \
    OK ATDT$1 \
    CONNECT " \
    sername: $ACCOUNT \
    assword: $PASSWORD

    if [ "$?" = "0" ]; then
        exit 0
    fi

    return
}

initialize
if [ ! "$?" = "0" ]; then
    exit 1
fi

attempt=0
while : ; do
    attempt=`expr $attempt + 1`
    callnumber $TELEPHONE
    if [ "$attempt" = "$MAX_ATTEMPTS" ]; then
        exit 1
    fi
    sleep "$SLEEP_DELAY"
done

```

**Pliki** /etc/ppp/ppp-on, /etc/ppp/ppp-off /etc/ppp/redialer powinny mieć ustawione flagi wykonywalności:

```
chmod +x /etc/ppp/ppp-o* /etc/ppp/redialer
```

Od tej pory można dzwonić poprzez uruchamianie skryptu `/etc/ppp/ppp-on` a rozłączenie następuje po wywołaniu skryptu `/etc/ppp/ppp-off`. Plikiem konfiguracyjnym dla `pppd` jest plik `/etc/ppp/options`. Powinien mieć on zawartość taką jak opisana w rozdziale Sekcja 3.2.15.3, czyli:

```
defaultroute
lock
noauth
/dev/modem
```

W systemie Red Hat 7.x dla prawidłowego działania konieczne może być usunięcie opcji `noauth`. Można modyfikować konfigurację poprzez zmiany w skryptach `/etc/ppp/ppp-on`, `/etc/ppp/ppp-off` i `/etc/ppp/redialer` (np. zmiana ISP przez odpowiednie ustawienie `TELEPHONE`, `ACCOUNT` i `PASSWORD`). Plik `/etc/ppp/redialer` odpowiada za dzwonienie „do skutku”, lub do przekroczenia maksymalnej ilości prób (np. w przypadku zajętości linii). Ta ilość prób jest ustalana w zmiennej `MAX_ATTEMPTS` w pliku `/etc/ppp/redialer`.

### 3.2.15.6. Konfiguracja klienta i serwera ppp w przypadku modemu GSM (model Siemens M20 + sieć Era)

Jeśli chodzi o fizyczne podłączenie modemu, to jego instalacja jest analogiczna do zwykłego modemu - tworzymy link `/dev/modem` wskazujący na port szeregowy, do którego jest podłączony modem. Komunikacja następuje z prędkością 19200 kbps (8N1), jest kontrola przepływu RTS/CTS.

Problem komplikuje się nieco, kiedy chcemy ustawić odpowiadanie modemu na połączenie przy użyciu programu `mgetty`. Otóż zanim można wykonywać jakiegokolwiek operacje, modem musi się zalogować do sieci GSM, ale do tego potrzebny jest kod PIN. Do wprowadzenia kodu PIN służy komenda `AT+CPIN="nnnn"`. Po wprowadzeniu prawidłowego PIN dioda na modemie przestaje migać i zapala się na stałe. Zatem program `mgetty` przy inicjalizacji modemu musi wprowadzać prawidłowy kod PIN. Można tego dokonać za pomocą opcji `-m`. Do `/etc/inittab` należy dopisać następującą linijkę (xxxx oznacza oczywiście kod PIN):

```
m:345:respawn:/sbin/mgetty -m '"" AT&F&C1&D2S7=55 OK ATS0=1 OK
AT+CPIN="xxxx"' -s 19200 /dev/modem
```

Po prawidłowym skonfigurowaniu `mgetty` komputer będzie automatycznie podawał kod PIN. Pora teraz na skonfigurowanie połączeń z siecią Internet. Sieć EraGSM ma numer dostępowy 0604010101, pod którym znajduje się protokół PPP z autoryzacją PAP. Do katalogu `/etc/ppp` należy skopiować następujące pliki (wersja dla Era GSM):

Plik `pap-secrets`:

```
# Secrets for authentication using PAP
# client server secret IP addresses

erantet "*" erantet
```

Plik `ppp-on-dialer`:

```
#!/bin/sh
#
exec chat -v \
```

```

TIMEOUT 3 \
ABORT '\nBUSY\r' \
ABORT '\nNO ANSWER\r' \
ABORT '\nRINGING\r\n\r\nRINGING\r' \
" \rAT \
'OK-+++\c-OK' ATH0 \
TIMEOUT 30 \
OK ATDT$TELEPHONE \
CONNECT '\n'

```

**Plik ppp-on:**

```

#!/bin/sh

TELEPHONE=604010101 # The telephone number for the connection
LOCAL_IP=0.0.0.0 # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP=0.0.0.0 # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.255.0 # The proper netmask if needed
ACCOUNT=eranet

export TELEPHONE ACCOUNT

DIALER_SCRIPT=/etc/ppp/ppp-on-dialer

exec /usr/sbin/pppd lock crtscts /dev/ttyS1 19200 \
defaultroute connect $DIALER_SCRIPT user $ACCOUNT

```

**Plik ppp-off:**

```

#!/bin/sh
#####
#
# Determine the device to be terminated.
#
if [ "$1" = "" ]; then
DEVICE=ppp0
else
DEVICE=$1
fi

#####
#
# If the ppp0 pid file is present then the program is running. Stop it.
if [ -r /var/run/$DEVICE.pid ]; then
    kill -INT `cat /var/run/$DEVICE.pid`
#
# If the kill did not work then there is no process running for this
# pid. It may also mean that the lock file will be left. You may wish
# to delete the lock file at the same time.
    if [ ! "$?" = "0" ]; then
        rm -f /var/run/$DEVICE.pid
        echo "ERROR: Removed stale pid file"
        exit 1
    fi
fi

#

```

```
# Success. Let pppd clean up its own junk.
    echo "PPP link to $DEVICE terminated."
    exit 0
fi
#
# The ppp process is not running for ppp0
echo "ERROR: PPP link is not active on $DEVICE"
exit 1
```

Ważna jest opcja `user` programu `pppd`, która mówi jaki użytkownik ma się łączyć z siecią Internet. Dla sieci Era jest to `erantet`.

### 3.2.15.7. Instalacja i konfiguracja SDI

Ten rozdział opisuje sposób instalacji SDI (Szybkiego Dostępu do Internetu). Opis podzielono na 3 fazy.

- Przygotowania

Ponieważ komputer z SDI jest dostępny z całego Internetu, musimy go zabezpieczyć przed atakami z zewnątrz.

Co robimy?

- Korzystając z narzędzia `ntsystv` usuwamy niepotrzebne serwisy w ten sposób, żeby pozostały tylko serwisy: **atd**, **crond**, **gpm**, **identd**, **(x)inet**, **network**, **random**, **sshd**, **syslog**, **xfs**, **xntpd**
- Można wyrzucić także **xfs**, ale wówczas należy zmodyfikować `/etc/X11/XF86Config` wpisując tam jako kolejne sekcje **FontPath** wszystkie ścieżki podane w pliku `/etc/X11/fs/config` w sekcji **catalogue**.

Można też wyrzucić **xntpd**, o ile dany komputer nie pełni roli serwera ntp.

- Część usług jest uruchamianych przy użyciu superdemona `(x)inetd`. Należy więc jego konfigurację delikatnie zmodyfikować (w przypadku `inetd` będzie to modyfikacja pliku `/etc/inetd.conf`). Wykomentować należy nieużywane i niepotrzebne serwisy; w standardowym RH 6.2 nie stwierdzono żadnych potrzebnych serwisów, więc należy wykomentować *wszystko*.

- Konfiguracja urządzenia

- Podłączamy modem do komputera oraz do przygotowanej wcześniej linii abonenckiej.
- Tworzymy linka o nazwie `/dev/modem` wskazującego na odpowiednie urządzenie znakowe w katalogu `/dev`.
- Przystępujemy do właściwej konfiguracji komputera. Tworzymy (ew. modyfikujemy) odpowiednie pliki:

`/etc/ppp/options` o treści:

```
115200
modem
defaultroute
lock
noauth
```

```

user <nazwa użytkownika>
/dev/modem
persist

/etc/ppp/pap-secrets o treści:
<nazwa użytkownika> * <hasło tegoż użytkownika> <numer IP uzyskany od TPSA>

/etc/rc.d/init.d/his o treści:
#!/bin/bash
. /etc/rc.d/init.d/functions

[ -f /usr/sbin/pppd ] || exit 0
[ -f /etc/ppp/options ] || exit 0
RETVAL=0
case "$1" in
  start)
    echo -n 'Starting HIS connection... '
    daemon /usr/sbin/pppd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/pppd
    ;;
  stop)
    echo -n 'Shutting HIS connection... '
    killproc pppd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/pppd
    echo
    ;;
  reload|restart)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
  *)
    echo "Usage: /etc/rc.d/init.d/his {start|stop|reload|restart}"
    exit 1
esac
exit $RETVAL

```

Uwaga! Temu skryptowi należy dodać atrybuty wykonywalności (**chmod +x /etc/rc.d/init.d/his**).  
Należy go również dodać do skryptów startowych w pożądanym runlevelu.

Plik `/etc/ppp/chap-secrets` powinien być pusty.

- Udostępnianie połączenia innym użytkownikom sieci lokalnej

Jak to zrobić opisuje Sekcja 3.2.18.

### 3.2.15.7.1. Konfiguracja SDI pod Debianem

Można użyć sposobu opisanego powyżej, ale bardziej elegancki i zgodny z duchem Debiana będzie następujący:

- Tworzymy plik `/etc/ppp/peers/sdi` o zawartości:

```
115200
modem
defaultroute
lock
noauth
user NAZWA
/dev/modem
persist
```

Zamiast NAZWA wpisujemy przydzieloną przez TP S.A. nazwę użytkownika.

- Do pliku `/etc/ppp/pap-secrets` dodajemy linię postaci:

```
NAZWA * HASŁO IP
```

Oczywiście zamiast NAZWA, HASŁO i IP podajemy otrzymane od TP S.A. parametry - nazwę użytkownika, hasło i przydzielony numer IP.

- Do pliku `/etc/network/interfaces` dodajemy linie:

```
auto ppp0
iface ppp0 inet ppp
    provider sdi
```

- Ta elegancka konfiguracja ma jednak jedną wadę - mimo opcji `'persist'` nie zawsze demon `pppd` będzie podtrzymywał połączenie. Jako zabezpieczenie tworzymy więc plik `/etc/cron.d/ppp.cron` o zawartości:

```
#
# Every 10 minutes check if ppp0 is up.

9,19,29,39,49,59 * * * *      root    /sbin/ifconfig ppp0 || /sbin/ifup ppp0
```

Podane tu wartości są przykładowe - tu sprawdzanie odbywa się na minutę przed każdymi pełnymi 10 minutami.

Jeżeli nasz komputer ma być użyty jako pośrednik dostępu do Internetu dla innych komputerów w sieci, powinniśmy użyć mechanizmu NAT/Masquarading, który opisuje Sekcja 3.2.18.

### 3.2.15.8. Instalacja usługi Neostrada z wykorzystaniem modemu USB

Poniżej przedstawiono opis instalacji i konfiguracji usługi Neostrada TP z modemem USB Sagem F@st 800 lub Thomson SpeedTouch 330 pod systemem Debian. Dla osób, które używają innych systemów operacyjnych lub/i innego typu modemu polecamy stronę <http://neostrada.info/instalacja.php>, gdzie zebrano teksty dotyczące różnych sposobów instalacji Neostrady TP.

Uruchomienie Neostrady (zwłaszcza w przypadku modemu Sagem) wymaga zainstalowania kilku dodatkowych pakietów. Najłatwiej jest to zrobić przy użyciu narzędzia **apt-get**, które automatycznie rozwiąże zależności między pakietami. Jeśli to możliwe, należy zapewnić komputerowi, na którym odbywa się instalacja, tymczasowy dostęp do Internetu. Pozwoli to na użycie repozytoriów z najnowszymi wersjami pakietów oraz ściąganie plików nagłówkowych jądra (instalacja z modemem

Sagem). Jeśli tymczasowy dostęp do sieci nie jest możliwy, potrzebne pakiety wraz z zależnościami należy pobrać w innym miejscu i dostarczyć na przykład na płycie CD. Niezależnie od typu modemu będziemy potrzebowali pakietów *hotplug*, przyda się też *usbutils*.

*Uwaga!* Należy dopilnować, aby pakiet *hotplug* był w wersji co najmniej 0.0.20040329-22.

Wcześniejsze wersje uniemożliwiają wznawianie utraconego połączenia mimo zastosowania opcji „persist”. Szczegółowy opis problemu można znaleźć na stronie:

<http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=302709>.

Uruchomienie neostrady wymaga rejestracji - dokonujemy jej po wstępnym skonfigurowaniu połączenia z domyślnym użytkownikiem *rejestracja@neostrada.pl* i hasłem *rejestracja* na stronie:

[rejestracja.neostrada.pl](http://rejestracja.neostrada.pl) (<http://rejestracja.neostrada.pl>). Potrzebne będą dane takie, jak numer zgłoszenia (taki sam jak pierwsze litery i 6 pierwszych cyf numeru umowy), numer linii telefonicznej, na której jest instalowana usługa, numer PESEL (w przypadku osoby fizycznej) lub NIP nabywcy (w przypadku firmy). Po ich podaniu otrzymujemy identyfikator i numer PIN, podając je z kolei otrzymujemy nazwę użytkownika i hasło. Poniższy opis zakłada, że rejestracji dokonujemy na tym samym komputerze, na którym instalujemy usługę. Wymaga to więc późniejszego przekonfigurowania połączenia z otrzymaną w procesie rejestracji nazwą użytkownika i hasłem. Rejestracji można także dokonać wcześniej, na innym komputerze. Wtedy na etapie konfiguracji powinniśmy od razu podawać prawidłową nazwę użytkownika i hasło. Nazwa użytkownika ma zawsze postać *identyfikator@neostrada.pl*, podajemy je zawsze w pełnej postaci (a nie sam *identyfikator*).

### 3.2.15.8.1. Przygotowanie linii telefonicznej

Do prawidłowego działania modemy ADSL wymagają, aby wszystkie telefony podłączone do tej samej linii co modem zostały odseparowane przy pomocy odpowiednich filtrów. Dwa takie filtry dołączone są do zestawu instalacyjnego. W wypadku gdy do linii, z której ma korzystać modem, podłączone są więcej niż dwa aparaty telefoniczne można zainstalować filtr przed gniazdkiem (puszką), do której doprowadzona jest linia zewnętrzna, a do modemu poprowadzić stamtąd dodatkowy przewód.

### 3.2.15.8.2. Modem Sagem F@st 800

Poniższy opis dotyczy jądra o wersji co najmniej 2.6.15. Opis dla starszych jąder znajduje się w kolejnej sekcji.

Modem F@st 800 obsługiwany jest przez moduł *ueagle-atm*. Jest on standardowo dostępny w jądrach Debiana. Do uruchomienia go konieczne jest pobranie i zainstalowanie firmware'u. Jest on dostępny na stronie: <http://eagle-usb.org/ueagle-atm/non-free/>

Za załadowanie firmware'u potrzebny jest jeszcze *udev* lub *hotplug*. Aby sprawdzić czy są zainstalowane, należy wykonać polecenia:

```
dpkg -l udev hotplug
```

O zainstalowaniu pakietu świadczy litera 'i' na początku wiersza z nazwą pakietu. Gdyby żaden z nich nie był zainstalowany, należy zainstalować np. *udev*

```
apt-get install udev
```

Po przygotowaniu firmware'u można podłączyć modem. Jeśli był podłączony wcześniej, należy odłączyć go i podłączyć ponownie. Po wywołaniu polecenia `dmesg` powinno pojawić się:

```
[ueagle-atm] pre-firmware device, uploading firmware
[ueagle-atm] loading firmware ueagle-atm/eagleII.fw
[ueagle-atm] firmware uploaded
```

Następnie należy przygotować plik `/etc/ppp/peers/neostrada` z następującą zawartością:

```
debug
kdebug 1
user rejestracja@neostrada.pl
asynmap 0
lcp-echo-interval 2
lcp-echo-failure 7
plugin pppoa.so
0.35
noauth
usepeerdns
noipdefault
defaultroute
holdoff 4
persist
maxfail 25
```

Jeżeli rejestracja została już wykonana, należy podać otrzymaną nazwę użytkownika w miejsce *rejestracja@neostrada.pl*.

Do plików `/etc/ppp/pap-secrets` i `/etc/ppp/chap-secrets` należy wpisać nazwę użytkownika i hasło. Dla rejestracji będzie to:

```
rejestracja@neostrada.pl *   rejestracja *
```

Kolejną czynnością jest edycja pliku `/etc/network/interfaces`. Należy dodać do niego sekcję:

```
auto pppoa
iface pppoa inet ppp
    provider neostrada
```

Jeżeli istnieje plik `/etc/ppp/resolv.conf`, to należy wykonać kopię zapasową pliku `/etc/resolv.conf` a na jego miejsce podlinkować `/etc/ppp/resolv.conf`:

```
mv /etc/resolv.conf /etc/resolv.conf.old
ln -s /etc/ppp/resolv.conf /etc/resolv.conf
```

Po wykonaniu powyższych czynności można nawiązać połączenie. W tym celu wykonujemy polecenie:

```
ifup pppoa
```

Jeśli wszystko zastało wykonane prawidłowo po wykonaniu polecenia **ifconfig** powinny pojawić się komunikaty tej postaci:

```
ppp0      Link encap:Point-to-Point Protocol
          inet addr:83.27.227.217  P-t-P:213.25.2.37  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:9178  Metric:1
          RX packets:175922 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:144799 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:3
RX bytes:18388453 (17.5 MiB) TX bytes:19310312 (18.4 MiB)
```

### 3.2.15.8.3. Modem Sagem F@st 800 - jądra < 2.6.15

Modem Sagem F@st 800 w jądrach starszych niż 2.6.15 obsługiwany był przez moduł ueagle-usb. Poniżej zamieszczamy opis uruchomienia neostrady z wykorzystaniem tego modułu.

Zanim rozpoczniemy właściwą instalację, musimy przygotować pakiety z nagłówkami do używanej wersji jądra oraz programy narzędziowe. Dodatkowo, użyjemy pakietu **module-assistant**, żeby zautomatyzować proces konfiguracji i instalacji sterowników modemu. Wydajemy zatem komendy:

```
apt-get install module-assistant
apt-get install eagle-usb-modules-source
```

**Notatka:** Uwaga! Jeśli nie mamy dostępu do sieci, pakiety muszą być zainstalowane na komputerze ręcznie, co jest dosyć kłopotliwe, bo np. pakiet **eagle-usb-modules-source** ma kilka zależności, które należy również ściągnąć i zainstalować.

Gdy mamy zainstalowane oba pakiety, należy przygotować moduły dla naszego systemu. Wywołujemy następującą komendę:

```
module-assistant -t auto-install eagle-usb
```

Następnie podłączamy modem do portu USB i instalujemy programy narzędziowe do modemu:

```
apt-get install eagle-usb-utils
```

Po zainstalowaniu uruchomi się konfigurator:

- W pierwszym oknie wybieramy opcję „Polska” (jeżeli mamy odpowiednio nową wersję pakietu eagle-usb-utils, to od razu będziemy mogli wybrać neostradę TP S.A.).
- W kolejnym oknie musimy wpisać nazwę użytkownika podaną w procesie rejestracji do Neostrady (jeżeli logujemy się po raz pierwszy należy tu wpisać „rejestracja@neostrada.pl”).
- W następnym oknie podajemy hasło (przy pierwszym logowaniu się do sieci podajemy hasło „rejestracja”).
- Na pytanie „Does your ISP supports password encryption?” odpowiadamy „Yes”.
- Na pytanie „Start DSL connection automatically?” odpowiadamy „onplug”.
- Teraz modem będzie próbował stworzyć sesję ADSL, co może trochę potrwać. Jeżeli wszystko przebiegło OK, powinniśmy otrzymać komunikat

```
Konfiguracja zakończona z powodzeniem.
```

Możesz teraz wydać komendę "startadsl", aby nawiązać połączenie

- Teraz powinniśmy zmodyfikować plik `/etc/ppp/options`, najlepiej tak, żeby jego zawartość była identyczna z plikiem `/etc/eagle-usb/scripts/options`.
- Powinniśmy również sprawdzić czy w pliku `/etc/ppp/chap-secrets` mamy poprawnie wpisany podany przez nas login i hasło.
- Wydajemy komendę **startadsl** i powinniśmy uzyskać połączenie z siecią.
- Połączenie można zakończyć poleceniem: **stopadsl**

Uwaga! Czasami zdarza się, że konfigurator nie ustawi odpowiednio konfiguracji (skrypt **startadsl** zawiesza się wówczas na dłuższy czas po czym informuje, że nie może nawiązać połączenia). Należy wtedy ręcznie uruchomić skrypt **eagleconfig** (jest to konsolowy konfigurator modemu) i skonfigurować połączenie powtórnie.

Jeżeli logujemy się po raz pierwszy (używamy loginu/hasła: `rejestracja@neostrada.pl/rejestracja`), to po uzyskaniu połączenia będziemy mogli wejść tylko na stronę `http://rejestracja.neostrada.pl`. Należy wówczas przejść przez kilka kroków wg. instrukcji ze strony. Otrzymamy tu nasze nowe, indywidualne hasło i nazwę użytkownika. Wtedy należy zakończyć połączenie i powtórnie uruchomić skrypt **eagleconfig**, żeby wprowadzić nowo otrzymaną nazwę użytkownika i hasło.

#### 3.2.15.8.4. Modem Thomson SpeedTouch 330

Uruchomienie Neostrady przy użyciu modemu SpeedTouch wymaga zainstalowania pakietu `hotplug`. Instalujemy go (jeżeli nie jest zainstalowany) przez wydanie polecenia:

```
apt-get install hotplug
```

Sterownik do modemu (moduł `speedtouch`) we wszystkich nowszych jądrach powinien być już obecny, trzeba natomiast zadbać o ładowanie firmware'u modemu. Firmware najłatwiej skopiować z płytki instalacyjnej dostarczonej przez TP S.A. wraz z zestawie instalacyjnym. Kopiujemy z niej np. do katalogu `/tmp` plik `neostrada_tp.tgz` i rozpakowujemy go:

```
mount /media/cdrom
cp /media/cdrom/Linux/ST330/neostrada_tp.tgz /tmp
cd /tmp
tar xzf neostrada_tp.tgz
```

Wchodzimy do powstałego po rozpakowaniu katalogu `neostrada_tp` i rozpakowujemy w nim archiwum `ST330_firmware_3012.zip` (potrzebny jest do tego program **unzip** - jeśli nie jest zainstalowany to oczywiście wydajemy komendę **apt-get install unzip**):

```
cd neostrada_tp
unzip ST330_firmware_3012.zip
```

Pliki z firmware są dwa do wyboru. Wszystkie obecnie sprzedawane modele modemu wymagają pliku `ZZZL_03.012`. Pliku `KQD6_3.012` używamy, jeżeli mamy starszą wersję modemu, co można rozpoznać przez wydanie komendy (potrzebna jest program **lsusb** z pakietu `usbutils`):

```
lsusb -d 0x06b9:0x4061 -v | grep bcdDevice
```

W przypadku starszych modemów powinien zostać wypisany numer wersji 2.00, w przypadku nowszych otrzymujemy 4.00. Można bezpiecznie założyć, że mamy nowszą wersję i sprawdzać tylko jeżeli coś nie działa.

Objawem wskazującym, że próbujemy załadować złą wersję firmware'u, jest następujący komunikat pojawiający się w pliku `/var/log/syslog`.

```
... speedtch_upload_firmware: read BLOCK2 from modem failed (-110)!
```

Dalsze postępowanie zależy od używanej przez nas wersji jądra (sprawdzamy poleceniem: **uname -r**):

- Jeżeli korzystamy z wersji wcześniejszej niż 2.6.10, to konieczny będzie pakiet *usbutils*, a także musimy zainstalować pakiet *speedtouch* (uwaga - obecnie znajduje się on w repozytorium *stable*):

```
apt-get install speedtouch
```

Następnie tworzymy katalog `/usr/local/lib/speedtouch` i kopiujemy do niego plik z firmware'em:

```
mkdir /usr/local/lib/speedtouch
cp ZZZL_3.012 /usr/local/lib/speedtouch
```

- Jeżeli korzystamy z jądra 2.6.10 lub nowszego, to jądro samo ładuje dla nas firmware, musimy go tylko odpowiednio przygotować, czyli podzielić na dwa pliki - kod ładujący i właściwy. W tym celu do naszego tymczasowego katalogu ściągamy plik spod adresu:

<http://prdownloads.sourceforge.net/speedtouch/speedtouch-1.3.1.tar.gz?download>. Rozpakowujemy go a następnie rozpakowujemy kolejny plik umieszczony wewnątrz ściągniętego archiwum:

```
tar xzf speedtouchconf-06-03-2006.tar.gz
cd speedtouchconf-06-03-2005
tar xzf speedtouch-1.3-sgp.tar.gz
speedtouch-1.3-sgp
```

Następnie kompilujemy mały programik służący do dzielenia pliku z firmware na dwa:

```
cd src
gcc -o firmware firmware.c -DSTANDALONE_EXTRACTER
```

Powinniśmy otrzymać w aktualnym katalogu program `firmware`. Uruchamiamy go, podając jako argument oryginalny plik z firmware:

```
./firmware ../../../../ZZZL_3.012
```

Powinniśmy otrzymać dwa pliki wyjściowe - `boot.bin` i `firmware.bin`. Zmieniamy im nazwy i kopiujemy do odpowiedniego katalogu:

```
mv boot.bin /usr/local/lib/firmware/speedtch-1.bin
mv firmware.bin /usr/local/lib/firmware/speedtch-2.bin
```

Oczywiście można też skopiować te pliki np. z innego komputera z zainstalowanym modemem SpeedTouch.

Po wykonaniu powyższych czynności możemy podłączyć modem. W pliku `/var/log/syslog` powinny pojawić się informacje o zarejestrowaniu sterownika:

```
... kernel: usbcore: registered new driver speedtch
```

W przypadku jądra 2.6.10 lub nowszego powinniśmy też zobaczyć komunikaty o pozytywnym załadowaniu firmware:

```
... kernel: usb 2-3: found stage 1 firmware speedtch-1.bin
... kernel: usb 2-3: found stage 2 firmware speedtch-2.bin
```

Następnie należy przygotować plik `/etc/ppp/peers/neostrada` z następującą zawartością:

```

debug
kdebug 1
user rejestracja@neostrada.pl
asyncmap 0
lcp-echo-interval 2
lcp-echo-failure 7
plugin pppoa.so
0.35
noauth
usepeerdns
noipdefault
defaultroute
holdoff 4
persist
maxfail 25

```

Możemy także dodać opcję *replacedefaultroute* jeżeli chcemy, aby uruchomienie połączenia przez neostradę powodowało zastąpienie ewentualnie używanej innej bramy domyślnej. Należy także zwrócić uwagę na ewentualne opcje zawarte w pliku `/etc/ppp/options` - najlepiej, aby był on pusty, żeby opcje przeznaczone np. dla modemu telefonicznego nie psuły nam konfiguracji.

Do plików `/etc/ppp/pap-secrets` i `/etc/ppp/chap-secrets` dodajemy linię:

```
rejestracja@neostrada.pl *   rejestracja *
```

Kolejną czynnością jest edycja pliku `/etc/network/interfaces`. Należy dodać do niego sekcję:

```

auto pppoa
iface pppoa inet ppp
    provider neostrada

```

Jeżeli istnieje plik `/etc/ppp/resolv.conf`, to należy wykonać kopię zapasową pliku `/etc/resolv.conf` a na jego miejsce podlinkować `/etc/ppp/resolv.conf`:

```

mv /etc/resolv.conf /etc/resolv.conf.old
ln -s /etc/ppp/resolv.conf /etc/resolv.conf

```

Po wykonaniu powyższych czynności można wykonać pierwsze połączenie w celu zarejestrowania się i otrzymania identyfikatora i hasła. W tym celu wykonujemy polecenie:

```
ifup pppoa
```

Jeśli wszystko zastało wykonane prawidłowo po wykonaniu polecenia **ifconfig** powinny pojawić się komunikaty tej postaci:

```

ppp0      Link encap:Point-to-Point Protocol
          inet addr:83.27.227.217  P-t-P:213.25.2.37  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:9178  Metric:1
          RX packets:175922 errors:0 dropped:0 overruns:0 frame:0
          TX packets:144799 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:18388453 (17.5 MiB)  TX bytes:19310312 (18.4 MiB)

```

Należy teraz przeprowadzić rejestrację zgodnie z instrukcją dostarczoną w zestawie instalacyjnym. Uzyskany w jej trakcie identyfikator użytkownika i hasło należy zapisać.

Następnie należy wyedytować pliki `/etc/ppp/pap-secrets` i `/etc/ppp/chap-secrets` zastępując **rejestracja@neotrada.pl** przez pełną nazwę użytkownika a **rejestracja** przez hasło. Plik po edycji powinien wyglądać w następujący sposób:

```
login@neotrada.pl * haslo *
```

Uzyskaną nazwą użytkownika należy także zastąpić napis „rejestracja” w pliku `/etc/ppp/peers/neotrada`.

Kolejnym krokiem jest wyłączenie i ponowne włączenie interfejsu pppoa:

```
ifdown pppoa
ifup pppoa
```

W tym momencie Neotrada powinna być w pełni skonfigurowana i gotowa do użycia. Można to sprawdzić wydając ponownie polecenie: **ifconfig**, które powinno, jak poprzednio, pokazać informacje o skonfigurowanym interfejsie ppp0, lub wydając np. polecenie:

```
ping google.pl
```

które powinno wyświetlić komunikaty:

```
PING google.pl (216.239.39.99) 56(84) bytes of data.
64 bytes from 216.239.39.99: icmp_seq=1 ttl=236 time=130 ms
64 bytes from 216.239.39.99: icmp_seq=2 ttl=236 time=131 ms
64 bytes from 216.239.39.99: icmp_seq=3 ttl=236 time=132 ms
```

Aby przerwać wyświetlanie należy nacisnąć Crtl-C.

Ze względu na długi czas synchronizacji modemu, interfejs może nie zdążyć skonfigurować się podczas startu systemu. Można temu zapobiec zmieniając plik `/etc/network/interfaces` na przykład w następujący sposób:

```
auto pppoa
iface pppoa inet ppp
    provider neotrada
    pre-up while ! grep 'Line up' '/proc/net/atm/speedtch:0' \
        &> /dev/null ; do sleep 1; done
    post-up while ! ifconfig ppp0 &> /dev/null ; do sleep 1; done
```

### 3.2.15.9. Instalacja usługi Internet DSL

Usługa „Internet DSL” jest usługą oferowaną przez Telekomunikację Polską. Podobnie jak usługi SDI i Neotrada (Plus) technologią dostępową jest tu xDSL (a dokładnie ADSL). W zależności od wersji, użytkownik otrzymuje od jednego do czterech numerów IP z klasy publicznej oraz modem ADSL z wyjściem Ethernet. Dzięki interfejsowi Ethernet użytkownik musi skonfigurować komputery w sieci korzystając ze standardowych ustawień sieci w sieci Ethernet (por. opis dla systemów pochodnych od RedHata, opis dla systemu Debian). Od dostawcy usługi powinniśmy dostać wszystkie potrzebne dane na temat sieci: numer IP lub pulę numerów IP, adres maski podsieci i adres rozgłoszeniowy. Dostajemy też

adres IP interfejsu Ethernet modemu ADSL. Ten oto adres należy ustawić jako adres bramki (gateway). Dodatkowo powinniśmy ustawić adres serwera DNS w pliku `/etc/resolv.conf`, np. dodając linię:

```
nameserver 194.204.152.34
```

Po skonfigurowaniu sieci należy sprawdzić czy nasz komputer widzi interfejs w modemie („bramkę”). W kolejnym kroku możemy spróbować już „spingować” jakiegoś hosta w Internecie.

Uwaga! Usługę DSL często zakłada się w miejscach, gdzie istnieje działająca sieć lokalna. Jeśli nie mamy do dyspozycji dwóch kart sieciowych, możemy spróbować podłączyć „bramkę” jak i hosta do huba w naszej sieci lokalnej a na hoście dodać nowy adres internetowy jako tzw. alias IP. Procedurę jak to należy zrobić opisuje sekcja Sekcja 3.2.13.

Dodatkowo, jeżeli nasz komputer ma być użyty jako pośrednik dostępu do Internetu dla innych komputerów w sieci lokalnej, powinniśmy użyć mechanizmu NAT/Masquarading, który opisuje Sekcja 3.2.18.

### 3.2.15.10. Instalacja i konfiguracja CDI

CDI jest to dostęp do Internetu oferowany przez firmę Dialog (szczegóły ([http://www.dialog.pl/dialoginternet\\_glowna.php](http://www.dialog.pl/dialoginternet_glowna.php))). Wykorzystywana jest tutaj technologia ISDN (a nie xDSL jak w przypadku SDI). Adres IP przydzielany jest w tym przypadku dynamicznie (przy każdym zestawieniu połączenia użytkownik może otrzymać inny adres IP). Poniżej przykład konfiguracji CDI z Dialogu z urządzeniem końcowym Aethra NT-IP 1197. Zakładamy, że operator przydzielił nam następujące parametry:

```
identyfikator (uzytkownik): 2384012483D
haslo: SHDKWP
numer dostepowy: 2401383
```

Należy utworzyć następujące pliki:

`/etc/ppp/dialog` (w tym pliku za linią `'ATZ'` można wstawiać dodatkowe komendy: dla zakończenia Aethra NT-IP: `'ATB41CL2048', 'ATB40&J3', 'ATB40&38;H3&J3JA1JS1KAS1KSS30CE1'`; dla zakończenia Elcon NT-IP NT1+2ab+V24: `'ATB41CL2048', albo: 'ATZB40&J3!X22=0', 'ATS7=60S30=0&H3'`):

```
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
" 'ATZ'
'OK' 'ATD2401383'
'CONNECT' "
```

`/etc/ppp/options` (poniższy ustawia automatycznie DNSy Dialogu: 217.30.129.149 i 217.30.137.200; domena to: `dialog.net.pl`):

```
115200
```

```

debug
defaultroute
name 2384012483D
/dev/modem
lock
crtscts
receive-all
connect '/usr/sbin/chat -f /etc/ppp/dialog'
usepeerdns
persist

```

```
/etc/ppp/chap-secrets:
```

```

# Secrets for authentication using CHAP
# client server secret IP addresses
2384012483D * SHDKWP

```

Jeżeli powyższa konfiguracja nie będzie działała, można spróbować zamienić 'name' na 'user' a także utworzyć plik `/etc/ppp/pap-secrets` o takiej samej zawartości jak `/etc/ppp/chap-secrets`.

Dla automatycznego startu `pppd` należy utworzyć skrypt `/etc/rc.d/init.d/his` i dodać go do skryptów startowych, jak to opisano w rozdziale o konfiguracji SDI, gdzie opisano też udostępnianie połączenia innym użytkownikom.

Można też łączyć się z wykorzystaniem `wvdial`, który pozwala na bardzo szybko działającą konfigurację, która jednak ma tę wadę, że przy BODASie każdorazowy zrzut danych na serwer `praterm.com.pl` powoduje utworzenie nowej sesji `pppd` i przydziału innego adresu IP (przeciętny czas życia jednej sesji to ok. 4 minuty). Nie jest możliwy więc zdalny dostęp do takiego systemu. Przy konfiguracji `pppd` opisanej powyżej, adres IP będzie zmienny za każdym uruchomieniem `pppd` (np. po zawiśnięciu po stronie Dialoga), czyli stosunkowo rzadko (w ideale tylko przy restarcie komputera).

Aby skonfigurować `wvdial` (służący do tego celu `wvdialconf` sam nie wykryje zakończenia CDI), należy utworzyć plik `/etc/wvdial.conf`:

```

[Dialer Defaults]
Modem = /dev/modem
Baud = 115200
Init1 = ATZ
# Ponizsze dwie linie sa opcjonalne dla zakonczenia Aethra NT-IP 1197
# moze ich nie byc
Init2 = ATB41CL2048
Init3 = ATB40&J3
# Ponizsza linia jest opcjonalna dla zakonczenia Elcon NT-IP NT1+2ab+V24
# moze jej nie byc
# Init2 = ATB41CL2048
# Ponizej jeszcze dwie linie opcjonalne
# Init2 = ATZB40&J3!X22=0
# Init3 = ATS7=60S30=0&H3
Phone = 2401383
Username = 2384012483D
Password = SHDKWP

```

i na koniec `/etc/rc.d/rc.local` dodać linijkę:

```
/usr/bin/wvdial &
```

wvdial przy zestawieniu każdego połączenia ustawia odpowiednio `/etc/ppp/pap-secrets`:

```
UZYTKOWNIK * HASLO
```

Dodatkowo trzeba ręcznie ustawić DNS Dialogu w `/etc/resolv.conf`.

### 3.2.15.11. Konfiguracja połączenia przy wykorzystaniu ISDN

ISDN to system telefonii cyfrowej dostępny już na rynku od dłuższego czasu. Urządzenia do jego obsługi mogą występować w kilku formach:

- terminale podłączane przez port szeregowy (rs232) albo ethernet
- terminale w formie kart rozszerzeń (ISA/PCI)

Tutaj zajmiemy się przypadkiem drugim - kart rozszerzeń, oraz konfiguracją do działania zarówno jako serwer i klient dla połączeń przy wykorzystaniu sieci ISDN.

#### 3.2.15.11.1. Konfiguracja karty

Pierwszym etapem jest konfiguracja samej karty ISDN. Obecnie w jądrze Linuksa istnieją dwa podsystemy do obsługi urządzeń ISDN: `isdn4linux` oraz `mISDN` (modular ISDN). Konfigurację oprzemy na pierwszym z nich ponieważ jest standardowo dostępny w jądrach Debiana, natomiast `mISDN` wymaga kompilacji dodatkowych modułów.

Musimy odnaleźć moduł odpowiedni dla danej karty. Większość kart jest wspieranych przez moduł `HiSax`. Na podstawie pliku `/usr/src/linux/Documentation/isdn/README.HiSax` lub strony <http://www.linuxpowered.com/html/editorials/isdn/hisaxreadme.html> musimy odnaleźć odpowiedni identyfikator typu dla posiadanej karty (np. dla karty opartej na układzie HFC 2BDS0 PCI będzie to 35).

Lista przetestowanych kart: Sekcja 2.1.14.

W Debianie należy utworzyć plik w katalogu `/etc/modprobe.d/` o nazwie np. `isdn` z zawartością:

```
options hisax protocol=2 type=35
```

gdzie w miejsce 35 wpisujemy znaleziony typ. Po utworzeniu pliku można załadować moduł poleceniem:

```
modprobe hisax
```

W RedHacie tworzymy plik `/etc/sysconfig/isdncard` o zawartości:

```
MODULE=hisax
RESOURCES="protocol=2 type=35"
```

po czym restartujemy serwis `isdn`:

```
service isdn restart
```

(musimy także pamiętać o tym, by serwis ten był ładowany w czasie uruchamiania komputera - możemy to zrobić np. przy użyciu narzędzia `ntsysv`).

Po tych operacjach karta terminala ISDN jest już gotowa do dalszej konfiguracji.

### 3.2.15.11.2. Konfiguracja systemu Debian

Należy zadbać o to, by były zainstalowane następujące pakiety:

```
isdnutils-base
isdnlog
ippd
```

(są one dostarczone z naszą dystrybucją).

Poniżej zamieszczamy zalecane odpowiedzi na pytania zadawane w trakcie instalacji tych pakietów:

- *isdnutils* Na pytanie o ładowanie firmware'u - należy wybrać odpowiedź 'none'.
- *isdnlog* Należy wybrać 'other' a następnie wpisać kod *PL* oraz wpisać numer kierunkowy.
- *ippd* Instalator pyta o numer telefonu, z którym będziemy się łączyć. Należy pozostawić wartość domyślną *manual* i ręcznie skonfigurować połączenie.

Po zakończeniu instalacji konieczne jest również poprawienie pliku `/etc/isdn/isdn.conf` ponieważ kod kraju ustawiany jest nieprawidłowo. Należy wpisać:

```
COUNTRYCODE = 48
```

Pliki konfiguracyjne ISDN znajdują się w katalogu `/etc/isdn/`. Z każdą kartą powiązane są dwa interfejsy `ipp`, które mogą być wykorzystywane niezależnie. Dla każdego interfejsu potrzebne są dwa pliki konfiguracyjne `device.ippX` i `ippd.ippX`. Pierwszy z nich ustawia parametry urządzenia, drugi zawiera opcje dla demona `ippd`. Poniżej zamieszczamy przykładową konfigurację. Pliki te można również utworzyć przy pomocy narzędzia **isdnconfig**. Tworzy ono przykładową konfigurację, którą należy dostosować do własnych potrzeb.

Najpierw omówiona zostanie konfiguracja połączeń wychodzących. Zwykle sprawia ona mniej problemów może więc posłużyć do przetestowania karty.

#### 3.2.15.11.2.1. Połączenia wychodzące

Konfiguracja urządzenia `device.ippX`

```
#!/bin/sh

set -e

# Adresy IP.
# Jeżeli są przydzielane dynamicznie należy pozostawić wartości
# domyślne. W wypadku gdy są statyczne należy je wpisać.
LOCALIP=169.255.255.169 # IP klienta
```

```

REMOTEIP=10.0.0.2 # IP serwera

# Numery telefonow
LOCALMSN=203123456 # numer lokalny
REMOTEMSN=221345789 # numer serwera z kierunkowym, bez 0
# moze byc lista rozdzielona spacjami
LEADINGZERO=0 # dodawane do numeru serwera

# Tryb pracy urzadzenia, nie zmieniac.
DIALMODE=auto

# Czy ustawiac default route na to urzadzenie?
DEFROUTE=1

# urzadzenie - ippX
devnums=`basename $0`; devnums=${devnums##device.ippX}
device=ippX$devnums

dowhat="$1"

masterslave="$2"
case "x$masterslave" in
    xmaster) master=true; slave=false;;
    xslave) master=false; slave=true; masterdevice="$3";;
    x) master=false; slave=false;;
    *) echo "Error, $0 called with unknown 2nd arg '$masterslave', aborting!"
       exit 1;;
esac

# Configuration (start)

case "$dowhat" in
start)
    [ -f /proc/sys/net/ipv4/ip_dynaddr ] && echo 5 > /proc/sys/net/ipv4/ip_dynaddr

    # Utworzenie interfejsu
    if $slave
    then isdnctrl addslave ${masterdevice} ${device}
    else isdnctrl addif ${device}
    fi

    # Ustawienie MSN
    isdnctrl eaz ${device} $LOCALMSN

    # Ustawienie numeru, pod ktory dzwonimy
    if [ ! -z "$REMOTEMSN" ]; then
for MSN in $REMOTEMSN; do
    isdnctrl addphone ${device} out $LEADINGZERO$MSN
done
    fi

    # Rozlaczenie po X sekund nieaktywnosci, 0 wylaczone
    isdnctrl huptimeout ${device} 0

    # Enkapsulacja

```

```

isdnctrl encap ${device} syncppp

# Protokoly
isdnctrl l2_prot ${device} hdlc
isdnctrl l3_prot ${device} trans

# Gadatliwosc isdn, UWAGA dotyczy wszystkich urzadzen
isdnctrl verbose 2

# zwiazanie z deamonem ippdp
bindnum=`expr $device : 'ipp\(.*\)'` || true
if [ ! -z "$bindnum" ]; then
isdnctrl pppbind ${device} $bindnum
fi

# KONFIGURACJA SIECI

if ! $slave; then
ifconfig ${device} $LOCALIP pointopoint $REMOTEIP netmask 255.255.255.255
set +e # ignore errors from here on

route del -host $REMOTEIP ${device} 2>/dev/null
route add -host $REMOTEIP ${device}

# default route
if [ "$DEFROUTE" = 1 ]; then
route del default 2>/dev/null
route add default netmask 0 ${device}
fi

fi # not slave

if $slave
then isdnctrl dialmode $device auto >/dev/null 2>&1
else isdnctrl dialmode $device $DIALMODE >/dev/null 2>&1
fi
;;

# Usuwanie interfejsu
stop)
set +e

isdnctrl dialmode $device off >/dev/null 2>&1

if ! $slave; then
route del $REMOTEIP $device 2> /dev/null

# usuwanie default route
if [ "$DEFROUTE" = 1 ]; then
route del default netmask 0 2>/dev/null
fi
ifconfig $device down 2> /dev/null
isdnctrl delif $device 2> /dev/null
fi
;;

```

```

*)
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac

exit 0

```

W docelowym pliku należy ustawić:

- numery telefonów w zmiennych LOCALMSN i REMOTEMSN - odpowiednio lokalny i zdalny,
- adresy ip - jeśli są znane, w przeciwnym wypadku pozostawić domyślne,
- czy urządzenie ma być wykorzystywane jak domyślna brama (zmienna DEFROUTE).

### Konfiguracja daemona ipppd ipppd.ipppX

```

# Komunikaty daemona
debug # włącz dodatkowe komunikaty
kdebug 1 # poziom logowanych komunikatów

lcp-restart 1 # Set timeout for LCP

# Autoryzacja
user ppp # nazwa użytkownika
noauth #

# Kompresja wyłączona
noccp
nolzs

# Adresy IP
noipdefault
netmask 255.255.255.255

# Routing
nohostroute # nie ustawiaj routingu

#usepeerdns # użyj serwerów dns podanych przez interfejs

mru 1524 # maksymalny rozmiar odbieranych pakietów
mtu 1500 # maksymalny rozmiar wysyłanych pakietów

ipcp-accept-local # Accept peer's address for us
ipcp-accept-remote # Accept peer's address for it

```

Przedstawione ustawienia powinny umożliwić nawiązanie połączenia. Konieczne jest ustawienie nazwy użytkownika (user) oraz dodatnie wpisu z hasłem do pliku `/etc/ppp/pap-secrets` w postaci:

```
nazwa_użytkownika * hasło *
```

na przykład:

```
ppp * ppp *
```

Nie należy dodawać opcji defaultroute nawet gdy połączenie ISDN będzie podstawowym łączem. Routing ustawiany jest przez skrypt device.ippX. Do bardziej zaawansowanej konfiguracji routingu wykorzystuje się skrypty:

```
/etc/ppp/ip-up.d/00-ippd
/etc/ppp/ip-down.d/99-ippd
```

### 3.2.15.11.2.2. Połączenie przychodzące

Konfiguracja urządzenia device.ippX

```
#!/bin/sh

set -e

# Adresy IP.
LOCALIP=192.168.8.1 # IP serwera
REMOTEIP=192.168.8.8 # IP klienta

# Numery telefonow
LOCALMSN=203123456 # numer serwera bez 0
REMOTEMSN='*' # akceptowane numery klientow, * - wszystkie

LEADINGZERO=0 # dodawane do numeru serwera

# Tryb pracy urzadzenia, nie zmieniac.
DIALMODE=auto

# Czy ustawiac default route na to urzadzenie?
DEFROUTE=0

# urzadzenie - ippX
devnum=`basename $0`; devnums=${devnum##device.ipp}
device=ipp$devnum

dowhat="$1"

masterslave="$2"
case "x$masterslave" in
  xmaster) master=true; slave=false;;
  xslave) master=false; slave=true; masterdevice="$3";;
  x) master=false; slave=false;;
  *) echo "Error, $0 called with unknown 2nd arg '$masterslave', aborting!"
     exit 1;;
esac

# Configuration (start)

case "$dowhat" in
start)
```

```

[ -f /proc/sys/net/ipv4/ip_dynaddr ] && echo 5 > /proc/sys/net/ipv4/ip_dynaddr

# Utworzenie interfejsu
if $slave
then isdnctrl addslave ${masterdevice} ${device}
else isdnctrl addif ${device}
fi

# Ustawienie MSN
isdnctrl eaz ${device} $LOCALMSN

# Ustawienie numeru, pod który dzwonimy
if [ ! -z "$REMOTEMSN" ]; then
for MSN in $REMOTEMSN; do
isdnctrl addphone ${device} out $LEADINGZERO$MSN
done
fi

# Rozłączanie po X sekund nieaktywności, 0 wylaczone
isdnctrl huptimeout ${device} 0

# Enkapsulacja
isdnctrl encap ${device} syncppp

# Protokoly
isdnctrl l2_prot ${device} hdlc
isdnctrl l3_prot ${device} trans

# Gadatliwosc isdn, UWAGA dotyczy wszystkich urzadzen
isdnctrl verbose 2

# związanie z deamonem ipppd
bindnum=`expr $device : 'ippp\(.*\)'` || true
if [ ! -z "$bindnum" ]; then
isdnctrl pppbind ${device} $bindnum
fi

# KONFIGURACJA SIECI

if ! $slave; then
ifconfig ${device} $LOCALIP pointopoint $REMOTEIP netmask 255.255.255.255
set +e # ignore errors from here on

route del -host $REMOTEIP ${device} 2>/dev/null
route add -host $REMOTEIP ${device}

# default route
if [ "$DEFROUTE" = 1 ]; then
route del default 2>/dev/null
route add default netmask 0 ${device}
fi

fi # not slave

if $slave
then isdnctrl dialmode $device auto >/dev/null 2>&1

```

```

        else isdnctrl dialmode $device $DIALMODE >/dev/null 2>&1
        fi
        ;;

# Usuwanie interfejsu
stop)
    set +e

    isdnctrl dialmode $device off >/dev/null 2>&1

    if ! $slave; then
route del $REMOTEIP $device 2> /dev/null

# usuwanie default route
if [ "$DEFROUTE" = 1 ]; then
    route del default netmask 0 2>/dev/null
fi
ifconfig $device down 2> /dev/null
isdnctrl delif $device 2> /dev/null
    fi
    ;;

*)
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac

exit 0

```

Podobnie jak przy połączeniach wychodzących należy ustawić prawidłowy numer telefonu linii, do której podłączony jest modem, oraz adresy IP. Zmienna REMOTEMSNUM odpowiada za numery, z których połączenia będą akceptowane. \* oznacza, że przyjmowane będą wszystkie połączenia.

#### *Uwaga!* Komunikaty

```

kernel: isdn_net: ipp0: No phone number, signalling dst_link_failure
...
kernel: isdn_tty: Incoming call without CPN, assuming '0'
kernel: isdn_tty: call from 846385042 -> 0 ignored

```

oznaczają, że modem odbierający połączenie nie otrzymuje informacji pod jaki numer dzwoni modem nawiązujący połączenie. W tej sytuacji należy ustawić LOCALMSNUM na 0.

#### Konfiguracja daemona ippd.ippdX

```

# Komunikaty daemona
debug # włącz dodatkowe komunikaty
kdebug 1 # poziom logowanych komunikatów

lcp-restart 1 # Set timeout for LCP

# Autoryzacja
auth
+pap

# Kompresja wyłączona

```

```

noccp
nolzs

# Adresy IP
192.168.8.1:192.168.8.8 # serwer:klient
netmask 255.255.255.255

# Routing
nohostroute # nie ustawiaj routingu

#usepeerdns # uzyj serwerow dns podanych przez interfejs

mru 1524 # maksymalny rozmiar odbieranych pakietow
mtu 1500 # maksymalny rozmiar wysylanych pakietow

```

Przed uruchomieniem należy sprawdzić czy adres IP w pliku `device.ippX` jest zgodny z `ippd.ippX`.

### 3.2.15.11.3. Konfiguracja systemu RedHat

Poniżej omówiona zostanie konfiguracja dla systemu RedHat 7.3. Ten sposób konfiguracji może być również wykorzystany jako alternatywa dla Debiana w wypadku, gdy podany wyżej opis nie zadziała.

Mając skonfigurowany terminal, możemy spróbować połączyć się z czymś na zewnątrz. Do połączenia się z naszym ulubionym dostawcą (TPSA - numer 0-202422) można wykorzystać następujący skrypt (`/opt/szarp/bin/isdn-up-out.sh`):

```

#!/bin/sh

if [ x$1 = x ]; then
    devicenum=0
    device=ipp0
else
    devicenum=$1
    device=ipp$1
fi

#numer telefonu
number=0202422

#konfiguracja dla dial-out
isdnctrl addif $device
isdnctrl addphone $device out $number
isdnctrl huptimeout $device 0
isdnctrl l2_prot $device hdlc
isdnctrl l3_prot $device trans
isdnctrl encap $device syncppp
/usr/sbin/ippd \
    user ppp \
    ipcp-accept-local ipcp-accept-remote noipdefault lock \
    mru 1500 mtu 1500 usepeerdns useifip \
    /dev/$device
isdnctrl pppbind $device $devicenum
ifconfig $device up

```

i możemy go wywołać np. tak:

```
/opt/szarp/bin/isdn-up-out.sh 0
```

Musimy jeszcze skonfigurować hasło dla użytkownika ppp. Robimy to poprzez dodanie do pliku `/etc/ppp/pap-secrets` linijki:

```
ppp "*" ppp
```

W tym momencie terminal już jest skonfigurowany do połączeń z tpsa. W celu dokonania właściwego połączenia wywołujemy komendę:

```
isdnctrl dial ipp0
```

Możemy jeszcze ustawić właściwy routing:

```
route add default ipp0
```

Jeśli będziemy chcieli się rozłączyć, to wystarczy wywołać coś takiego:

```
isdnctrl hangup ipp0
```

Uruchomienie połączeń przychodzących wymaga trochę innego skryptu:

(`/opt/szarp/bin/isdn-up-in.sh`):

```
#!/bin/sh
```

```
if [ x$1 = x ]; then
    devicenum=0
    device=ipp0
else
    devicenum=$1
    device=ipp0$1
fi
```

```
#konfiguracja dla dial-in
isdnctrl addif $device
isdnctrl eaz $device 19
isdnctrl addphone $device in "*"
isdnctrl hupertimeout $device 0
isdnctrl l2_prot $device hdlc
isdnctrl l3_prot $device trans
isdnctrl encap $device syncppp
/usr/sbin/ipp0d \
    auth +pap \
    proxyarp netmask 255.255.224.0 192.168.8.1:192.168.8.8 \
    mru 1500 mtu 1500 usepeerdns useifip \
    /dev/$device
isdnctrl pppbind $device $devicenum
ifconfig $device up
```

Wywołujemy go podobnie jak poprzedni:

```
/opt/szarp/bin/isdn-up-in.sh 0
```

Musimy jeszcze skonfigurować hasło dla użytkownika ppp. Robimy to poprzez dodanie do pliku `/etc/ppp/pap-secrets` linijki:

```
ppp "*" ppp
```

W tym momencie terminal już jest skonfigurowany do przyjmowania połączeń przychodzących.

Może się także zdarzyć, iż chcemy, by na danej maszynie istniały obydwie konfiguracje. Wtedy wystarczy, iż wywołamy coś takiego:

```
/opt/szarp/bin/isdn-up-out.sh 0
/opt/szarp/bin/isdn-up-in.sh 1
```

W tym momencie powinniśmy już mieć możliwość jednoczesnego dzwonienia i przyjmowania połączeń.

Jeśli chcemy, by połączenia były konfigurowane przy starcie systemu musimy dodać wywołania skryptów `isdn-up-*.sh` do `/etc/rc.local` (RedHat). Możliwe jest jednoczesne używanie połączeń przychodzących i wychodzących. Należy tylko zadbać aby numer urządzeń w wywołaniach skryptów były różne (np. 0 - wychodzące, 1 - przychodzące).

W Debianie uruchomienie połączeń przy starcie systemu można uzyskać tworząc skrypt `/etc/isdn/device.ippX` o następującej zawartości:

```
#!/bin/sh

#prefix - skąd skrypty do isdn (kopiowane z palca)
prefix=
#dir - kierunek (in/out)
dir=out
#id - nr device'a (0/1 - ipp0/ipp1)
id=0

case "$1" in
  start)
    /opt/szarp/$prefix/isdn-up-$dir.sh $id
    ;;
  stop)
    /opt/szarp/bin/isdn-down.sh $id
    ;;
  *)
    echo "$0: (start|stop)"
    exit 1
    ;;
esac
```

W miejsce X i do zmiennej `id` przypisujemy numer urządzenia. By przetestować konfigurację wywołujemy:

```
invoke-rc.d isdn restart
```

### 3.2.15.12. Konfiguracja połączeń z Internetem przez Sferię

Ten rozdział opisuje jak łączyć się z Internetem za pomocą Sferii. Wszystkie ustawienia pokazano dla terminala Dowltel WS-2100 (przyłączany do komputera PC przez łącze szeregowo). Dla innych typów terminali konfiguracja powinna różnić się tylko nieznacznie.

Połączenia z Internetem przy pomocy Sferii (<http://www.sferia.pl>) przypominają „zwykłe”, wdzwaniane połączenia dial-up. Nie jest to więc stałe połączenie z Internetem, przed skorzystaniem z Internetu należy połączyć się za pomocą programu **kppp**, skryptu **szppp-on** itd. Każdy użytkownik otrzymuje od operatora własny login/hasło (poniżej stosujemy przykładowy login „uzytkow111” oraz hasło „haslo111”).

Żeby skonfigurować **pppd** do tworzenia połączeń internetowych poprzez Sferię, należy stworzyć następujące pliki:

- Plik `/etc/ppp/peers/sferia` o zawartości:

```
connect "/usr/sbin/chat -v -f /etc/chatscripts/sferia-connect-chat "
disconnect "/usr/sbin/chat -v -f /etc/chatscripts/sferia-disconnect-chat "

115200
updetach
/dev/modem
crtscts
local
ipcp-accept-local
ipcp-accept-remote
defaultroute
usepeerdns
noccp

user "uzytkow111"
password "haslo111"
```

Uwaga! Dwie ostatnie linie w pliku powyżej należy zmodyfikować odpowiednio do podanej w umowie nazwy użytkownika i hasła.

- Plik `/etc/chatscripts/sferia-connect-chat` o zawartości:

```
TIMEOUT 3
" 'AT&C2+CMUX=C,2;AT+CRM=150'
ABORT '\nBUSY\r'
ABORT '\nNO ANSWER\r'
ABORT '\nRINGING\r\n\r\nRINGING\r'
'OK-+++c-OK' \AT
'OK-+++c-OK' ATH0
TIMEOUT 120
OK ATDT#777
CONNECT ""
TIMEOUT 5
```

- Plik `/etc/chatscripts/sferia-disconnect-chat` o zawartości:

```
ABORT "BUSY"
ABORT "ERROR"
ABORT "NO DIALTONE"
SAY "\nReset modemu\n"
"" "\K"
"" "+++ATH"
```

Warto również dopisać skrypty ustawiające zawartość pliku `/etc/resolv.conf` (czyli serwery DNS) w taki sposób, że zawartość pliku będzie zawsze aktualizowana na czas połączenia. W tym celu tworzymy 2 skrypty (Uwaga! muszą one mieć ustawioną flagę wykonywalności):

- Plik `/etc/ppp/ip-up.local` o zawartości:

```
mv /etc/resolv.conf /etc/resolv.conf-pppbackup
cp -f /etc/ppp/resolv.conf /etc/resolv.conf
chmod a+r /etc/resolv.conf
```

- Plik `/etc/ppp/ip-down.local` o zawartości:

```
mv /etc/resolv.conf-pppbackup /etc/resolv.conf
```

Ewentualnie można ustawić ręcznie plik `/etc/resolv.conf` jak niżej:

```
nameserver 193.41.112.18
```

Połączenia rozpoczynane są w następujący sposób:

```
pppd call sferia
```

a kończone jak poniżej:

```
killall pppd
```

### 3.2.15.13. Konfiguracja połączenia PLIP

Czasem istnieje konieczność zestawienia prostej sieci dla dwóch maszyn. Nie zawsze niestety mamy dostępne karty sieciowe - wtedy istnieje kilka metod, takich jak:

- poprzez port szeregowy (rs232)
- poprzez port równoległy (drukarkowy)
- poprzez port usb

Tutaj opiszemy metodę wykorzystującą do tego port równoległy i odpowiednio spreparowany w tym celu kabel. Prędkość transmisji uzyskanej w ten sposób wynosi średnio ok. 60kB/s.

Do połączenia należy użyć kabla zakończonego z obydwu stron 25-pin wtyczką męską, takiego samego jak do np. połączenia kablowego w Windows, czy Windows Commander, Norton Commander. Połączenia między poszczególnymi pinami kabla (są symetryczne, więc obojętne z której strony się patrzy):

*Kabel drukarkowy do połączenia między dwoma komputerami*

```
2 - 15
3 - 13
4 - 12
5 - 10
6 - 11
10 - 5
11 - 6
12 - 4
13 - 3
```

15 - 2  
 16 - 16  
 17 - 17  
 25 - 25

Po złączeniu komputerów wystarczy wykonać kilka poleceń. Wspólne dla obu stron:

```
insmod parport
insmod parport_pc io=0x378 irq=7 #wartości z ustawień BIOSu
insmod plip
```

następnie po jednej stronie:

```
ifconfig plip0 172.20.0.1 pointopoint 172.20.0.2 netmask 255.255.255.255 up
```

(uwaga na pojedyncze „t” w słowie „pointopoint”), zaś po drugiej:

```
ifconfig plip0 172.20.0.2 pointopoint 172.20.0.1 netmask 255.255.255.255 up
```

Po tych operacjach łącze powinno już działać. Zostaje nam tylko przetestowanie przy użyciu np. polecenia ping.

### 3.2.15.14. Łączenie się z Internetem przez GPRS

Ten rozdział opisuje jak połączyć się z Internetem za pomocą telefonu komórkowego wykorzystując system GPRS. *General Packet Radio System* (GPRS) jest metodą dostępu do Internetu w sieciach GSM, która, w przeciwieństwie do zwykłego połączenia komutowanego (Seksja 3.2.15.6) oraz połączenia HSCSD, charakteryzuje się pakietową transmisją danych. Trzy cechy odróżniają system GPRS od pozostałych metod dostępu: po pierwsze, użytkownik płaci tylko za ilość przesłanych danych a nie za czas połączenia. Po drugie, GPRS jest szybszy od zwykłego połączenia dial-up z siecią przez telefon komórkowy (9,6 kbps), bo pozwala na łączenie kilku kanałów i osiągnięcie przepływu do abonenta (*downstream*) do 115 kbps (na razie ani sieci, ani telefony nie wspierają jeszcze tak dużych prędkości transmisji). Po trzecie, system GPRS pozwala odbierać połączenia i SMS-y w trakcie trwania sesji (połączenia) GPRS. Sesja jest wówczas na czas połączenia zawieszana, a po jego zakończeniu - wznawiana. Dodatkowym atutem jest czas nawiązywania połączenia - jest kilkakrotnie krótszy niż w przypadku zwykłych modemów dial-up.

Usługa GPRS musi być aktywowana u operatora telefonii komórkowej (w niniejszym opisie, dla uproszczenia wszystkie przykłady i konfiguracja będą przedstawione dla sieci Era). W Erze aktywacja usługi jest za darmo, standardowo nie ma również żadnej opłaty miesięcznej w abonamencie. Koszt: 49 gr (+VAT) za 100 kB wysłanych i (oddzielnie) odebranych danych (na poziomie protokołu IP). „Kwantem rozliczeniowym” jest właśnie 100 kB. Oznacza to, że minimalna opłata za sesję GPRS wynosi 49 gr (dane odebrane) + 49 gr (dane wysłane) = 98 gr (kilkaset bajtów musi być zawsze wysłanych i odebranych w ramach nawiązywania połączenia). Sesja GPRS przerwana połączeniem przychodzącym i (automatycznie) wznawiana, jest traktowana jak pojedyncza sesja (choćby trwała nawet cały dzień :-)). GPRS jest bardzo użyteczny przy połączeniach shellowych. Połączenie tekstowe/konsolowe z serwerem jest relatywnie tanie (choć do każdego znaku należy doliczyć nagłówek IP) i nie ma ciągłej presji czasu jak w przypadku zwykłego połączenia dial-up przez telefon komórkowy.

Naturalnie, system GPRS musi być wspierany przez telefon. Aktualnie coraz więcej telefonów ma wbudowanego GPRS-a, są to m.in.: Nokia 6310, 6310i, 7650, 8310; Ericsson R520, T65i, T39, T68; Siemens ME45, M50, S55 i wiele innych.

Sesje GPRS mogą być uruchamiane na dwa sposoby:

- Z poziomu telefonu, wówczas telefon pełni funkcję terminala:

```
Telefon ---- GPRS ---- sieć GSM ---- Internet
```

Możliwe zastosowania to m.in. WAP over GPRS (tu opłata jest z reguły wyższa niż przy dostępie do Internetu), SMS over GPRS (z rzadka wspierany), dostęp do Internetu przez MIDlety (programy w Javie na telefony komórkowe). Niewielkie możliwości terminalowe telefonów (przede wszystkim mały wyświetlacz) ograniczają możliwości zastosowań.

- Z poziomu komputera (wówczas telefon spełnia funkcję modemu):

```
Komputer PC ---- RS-232 ---- Telefon ---- GPRS ---- sieć GSM ---- Internet
```

```
Komputer PC ---- IrDA ---- Telefon ---- GPRS ---- sieć GSM ---- Internet
```

```
Komputer PC ---- Bluetooth ---- Telefon ---- GPRS ---- sieć GSM ---- Internet
```

Klasyczną już metodą połączenie telefonu do komputera specjalnym kablem (RS-232), który jest przyłączany do portu szeregowego (czyli COM-a). Ceny takich kabli są z reguły dosyć wysokie (ok. 150 zł), poza tym każdy producent ma własne standardy (Nokia ma ich kilka!), więc nie jest to rozwiązanie przenośne. Ostatnio rosnącą popularnością cieszą się połączenia bezprzewodowe - IrDA, oraz, wciąż jeszcze mało popularny - bo drogi - Bluetooth. Wybranie IrD-y jako medium komunikacyjnego z telefonem ma w przypadku GPRS-u pewien mankament: otóż gdy odbieramy rozmowę telefoniczną w trakcie połączenia GPRS to z reguły musimy podnieść telefon i tym samym zerwać komunikację w podczerwieni. Nawet, gdy po zakończeniu rozmowy odłożymy telefon na miejsce, to sesja GPRS nie może być wznowiona. Tego problemu nie ma w przypadku zastosowania kabla ani (najprawdopodobniej) w przypadku systemu Bluetooth.

### 3.2.15.14.1. IrDA

W podrozdziale tym opisany jest sposób instalacji i konfiguracji systemu IrDA na Linuksie z jednym z nowszych systemów RedHat (rodzina 7.x, 8.x, 9.x). Starsze systemy RedHat wymagały łat na jądro (teraz IrDA jest już standardowo wspierana) oraz doinstalowania pakietu **irda-utils** (obecnego w nowszych wersjach systemu).

IrDA jest obecna we wszystkich obecnie produkowanych laptopach a wyposażenie komputera stacjonarnego w system IrDA (dioda podczerwieni podłączana do płyty głównej) to kwestia kilkudziesięciu złotych.

Zanim zaczniemy konfigurować IrD-ę na Linuksie, powinniśmy dokładnie dowiedzieć się jakie są ustawienie w naszym systemie (sprawdzić w BIOS-ie). Często należy uaktywnić system IrDA, ponieważ może być standardowo wyłączony. Należy sprawdzić pod którym portem komunikacyjnym będzie widoczna IrDA. W przypadku laptopów będzie to zwykle COM2, czyli /dev/ttyS1. Co ciekawe, niektóre systemy biurkowe, wyposażone w dwa porty COM, podpinają IrD-ę pod COM2 - prawdopodobnie wyłączając standardowy, „kablony” port szeregowy. Warto także sprawdzić numer portu i przerwanie przydzielone przez BIOS.

Po włączeniu IrD-y w BIOS-ie i odczytaniu jej parametrów uruchamiany Linuksa. Następnie:

- Za pomocą komendy **dmesg | grep tty** sprawdzamy, które przerwanie przydzieliło interesującemu nas portowi komunikacyjnemu jądro Linuksa. Otrzymamy informację typu:

```
ttyS00 at 0x03f8 (irq = 4) is a 16550A
```

```
ttyS01 at 0x02f8 (irq = 3) is a 16550A
```

W tym przypadku port został skonfigurowany do używania portu 0x02f8 i przerwania 3. Jeżeli nie zgadza się to z ustawieniami w BIOS'ie, należy zmienić ustawienia portu za pomocą komendy **setserial**, np. jeżeli chcemy zmienić przerwanie na 10, możemy to uczynić komendą:

```
# setserial /dev/ttyS1 port 0x02f8 irq 10
Sprawdzamy czy wszystko poszło tak jak zamierzaliśmy:
```

```
# setserial -g /dev/ttyS1
/dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 10
```

Niestety, w większości przypadków jeżeli jądro automatycznie nie przydzieli portowi dobrego przerwania, odpowiednie wywołania **setserial** trzeba wykonywać przy każdym starcie systemu, przed uruchomieniem IrD-y (najlepiej dodać je do skryptów startowych systemu).

- Dodajemy po pliku `/etc/modules.conf` wpis:

```
alias tty-ldisc-11 irtty
alias char-major-161 ircomm-tty
```

- Modyfikujemy plik `/etc/sysconfig/irda` tak, żeby wyglądał jak poniżej:

```
IRDA=yes
DEVICE=/dev/ttyS1
DISCOVERY=yes
```

Jeżeli u nas IrDA jest na innym porcie szeregowym niż `/dev/ttyS1`, to należy odpowiednio zmodyfikować zmienną `DEVICE`.

- Uruchamiamy IrD-ę:

```
service irda start
```

.. i sprawdzamy czy został uruchomiony proces **irattach**:

```
ps auxw | grep irattach
```

Jeśli tak - to wszystko w porządku. Jeśli nie - to problem. Najczęstszą przyczyną jest zły port szeregowy lub złe ustawienia tego portu (np. przerwanie). Komunikat błędu można odczytać z `/var/log/messages`.

- Dodajemy usługę **irda** do usług uruchamianych przy starcie przez system RedHat (np. przy pomocy narzędzia **ntsysv**).

Teraz sprawdzamy czy nasz telefon zostanie wykryty przez system: włączamy IrD-ę w telefonie i stawiamy naprzeciwko nadajnika (ang. *transceiver*) IrD-y. Wywołujemy komendę:

```
cat /proc/net/irda/discovery
```

Jeśli telefon został wykryty, to program **cat** powinien na wyjściu na wyjściu zwrócić informację jak poniżej:

```
IrLMP: Discovery log:
```

```
nickname: Nokia 6310i, hint: 0xb125, saddr: 0xb4bd12f0, daddr: 0x0000b624
```

Wartości `nickname`, `hint`, `saddr` i `daddr` mogą się różnić w zależności od modelu telefonu - ważny jest fakt pojawienia się tej całej linii. W przypadku niewykrycia urządzenia zostanie zwrócony pusty log:

```
rLMP: Discovery log:
```

Powodem jest zwykle niewłączenie IrD-y w telefonie lub niewidzenie się czytnika IrD-y w telefonie i w komputerze (niektóre telefony mają podczerwień na czubku, a inne - z boku obudowy).

Jeśli telefon został wykryty, możemy spróbować się połączyć z telefonem za pomocą programu **minicom** (tu drobna dygresja: **minicom** jest programem do komunikacji przez porty/łącza szeregowy. W nomenklaturze komputerowej człon „com”, „COM” lub „comm”, pochodzący zapewne od słowa „communication”, oznacza komunikację szeregową (ang. *serial communications*), kojarzoną zwykle z portami szeregowymi, tzw. COM-ami w pececie i interfejsem RS-232. Jak się ma IrDA to komunikacji szeregowy? Otóż wiele standardów, w tym także bezprzewodowych, wspiera tryb komunikacji, który jest identyczny do tego znanego z RS-232. W przypadku IrD-y tryb ten zwie się *IrCOMM (Infared COMM)* a w przypadku Bluetootha *RFCOMM (Radio Frequency COMM)*. Dzięki temu istnieje prosta możliwość przeniesienia setek programów wykorzystujących komunikację szeregową do nowych zastosowań oraz możliwość tworzenia nowych aplikacji, w oparciu o tradycyjny i opanowany sposób komunikacji. Program **minicom** nie wie i nie musi wiedzieć, że porozumiewa się poprzez łącze bezprzewodowe. Koniec dygresji). Tu uwaga! Urządzeniem, które należy wskazać programowi **minicom** jest `/dev/ircomm0` a nie `/dev/ttyS1`. Sprawdzamy czy odpowiada na komendy Hayes'a (AT). Odpowiada? Więc IrDA została zainstalowana.

### 3.2.15.14.2. GPRS - Konfiguracja **pppd**

Opisany tu sposób konfiguracji połączenia przez GPRS został zaczerpnięty z dokumentu Mikko Rapeliiego pt.: „Of Linux, GPRS Phones, Serial Cable, IrDA, Bluetooth and USB” ([http://www.iki.fi/mikko.rapeli/linux\\_gprs.html](http://www.iki.fi/mikko.rapeli/linux_gprs.html)) i zmodyfikowany na potrzeby połączenia z siecią Era przez telefon Nokia 6310i. W plikach pozostawiono angielskie komentarze autora, żeby ułatwić ich dostosowanie do innych konfiguracji (np. Plus GSM, telefony Siemens'a itd.). Thank you, Mikko. Good job.

Żeby skonfigurować program **pppd** w celu obsługi systemu GPRS, należy stworzyć 4 pliki:

- Plik `/etc/ppp/peers/gprs` o zawartości:

```
# File: gprs
#
# Description:
# Serial cable, IrDA, Bluetooth and USB pppd options for GPRS phones.

# Keep pppd attached to the terminal:
# Comment this to get daemon mode pppd
#nodetach
updetach
# Opcja updetach oznacza, że pppd odłączy się od terminala dopiero
# gdy zostanie nawiązane połączenie.

# Debug info from pppd:
# Comment this off, if you don't need more info
debug

# Show password in debug messages
show-password

# Connect/Disconnect scripts:

connect "/usr/sbin/chat -v -f /etc/chatscripts/gprs-connect-chat"

disconnect "/usr/sbin/chat -v -f /etc/chatscripts/gprs-disconnect-chat"

# Serial device to which the GPRS phone is connected:
```

```

# /dev/ttyS0 for serial port (COM1 in Windows),
/dev/ircomm0
# /dev/ircomm0 for IrDA,
# /dev/ttyUB0 for Bluetooth (Bluez with rfcomm running) and
# /dev/ttyUSB0 for USB
#/dev/ttyS0 # serial port one
#/dev/ttyS1 # serial port two
#/dev/ircomm0 # IrDA serial port one
#/dev/rfcomm0 # Bluetooth serial port one
#/dev/ttyUSB0 # USB serial device, for example Orange SPV

# Serial port line speed
#115200 # fast enough
57600 # perhaps usefull with IrDA

# Hardware flow control:
# Use hardware flow control with cable, Bluetooth and USB but not with IrDA.
#crtscts # serial cable, Bluetooth and USB
nocrtscts # IrDA

# Ignore carrier detect signal from the modem:
local

# IP addresses:
# - accept peers idea of our local address and set address peer as 10.0.0.1
# (any address would do, since IPCP gives 0.0.0.0 to it)
# - if you use the 10. network at home or something and pppd rejects it,
# change the address to something else
:10.0.0.1

# pppd must not propose any IP address to the peer!
noipdefault

# Accept peers idea of our local address
ipcp-accept-local

# Add the ppp interface as default route to the IP routing table
defaultroute

# DNS servers from the phone:
# some phones support this, some don't.
usepeerdns

novj
nobsdcomp
novjccomp
nopcomp
noaccomp

# The phone is not required to authenticate:
noauth

# Username and password:
# If username and password are required by the APN, put here the username
# and put the username-password combination to the secrets file:
# /etc/ppp/pap-secrets for PAP and /etc/ppp/chap-secrets for CHAP

```

```
# authentication. See pppd man pages for details.
# Example, Radiolinja operator pap-secrets:
# "rlnet" * "internet" *
user "erainternet"
```

```
# Asyncmap:
# some phones may require this option.
#asyncmap 0xa0000
```

```
# No magic:
# some phones may require this option.
#nomagic
```

```
# Require PAP authentication:
# some phones may require this option.
#require-pap
```

- Plik /etc/chatscripts/gprs-connect-chat o następującej treści:

**Uwaga! Pod treścią skryptu jest kilka słów wyjaśnienia. Przeczytać koniecznie!**

**Zawartość pliku:**

```
# File: gprs-connect-chat
#
# Description:
#
# Set PDP context CID=1, protocol=IP, APN=internet:
# AT+CGDCONT=1,"IP","internet","",0,0
#
# Set CID=1 QoS requirements from the network, not supported by Nokia:
# AT+CGQREQ=1,0,0,0,0,0
#
# Set CID=1 minimum acceptable QoS parameters, not supported by Nokia:
# AT+CGQMIN=1,0,0,0,0,0
#
# 'Call' CID=1 (activate PDP context one, perform GPRS attach):
# ATD*99***1#
#
# Some phones like the Orange SPV (yes, the Microsoft Smartphone) use this
# dial string to start GPRS connection:
# ATD*99#
#
# The actual chat script:

TIMEOUT 5
ECHO ON
ABORT '\nBUSY\r'
ABORT '\nERROR\r'
ABORT '\nNO ANSWER\r'
ABORT '\nNO CARRIER\r'
ABORT '\nNO DIALTONE\r'
ABORT '\nRINGING\r\n\r\nRINGING\r'
" \rAT
TIMEOUT 12
SAY "Press CTRL-C to close the connection at any stage!"
SAY "\ndefining PDP context...\n"
OK ATH
```

```

OK ATE1
OK 'AT+CGDCONT=1,"IP","erainternet"'
OK ATD*99***1#
TIMEOUT 22
SAY "\nwaiting for connect...\n"
CONNECT ""
SAY "\nConnected."
SAY "\nIf the following ppp negotiations fail,\n"
SAY "try restarting the phone.\n"

```

Uwaga! W skrypcie powyżej występuje następująca linia:

```
OK 'AT+CGDCONT=1,"IP","erainternet"'
```

Jest ona specyficzna tylko dla operatora ERA (nazwa APN to „erainternet”). Należy ją zmienić w przypadku innych operatorów. Ustawienia dla polskich operatorów są następujące:

- Plus APN: „www.plusgsm.pl” (użytkownik: brak, hasło: brak).
- Idea APN: „www.idea.pl” (użytkownik: ppp, hasło: ppp).
- Era APN: „erainternet” (użytkownik: erainternet, hasło: erainternet).

Poniższa linia to wywołanie odpowiedniego numeru GPRS:

```
OK ATD*99***1#
```

Ogólna składnia polecenia wygląda następująco:

```
ATD*99[*APN[*Protocol[*CID]]]#
```

gdzie zarówno „APN”, „Protocol” jak i „CID” są opcjonalne (przyjmują wówczas wartości domyślne). Wartości „Protocol” nie należy ustawiać nigdy - domyślne „PPP” jest jak najbardziej OK. Składnia minimalna wygląda następująco:

```
ATD*99#
```

Jest to jednak składnia niezalecana przez niektórych producentów telefonów, szczególnie tych, które wspierają możliwość wpisania więcej niż jednego aktywnego punktu dostępu (APN-a). Każdy taki punkt jest wpisany w telefonie pod określoną pozycją (1, 2, ..) - ta pozycja to jest właśnie CID (connection ID). Niektórzy operatorzy wpisują swoje ustawienia w sprzedawanych telefonach, zwykle trzeba to jednak zrobić ręcznie. Najbezpieczniejszą składnią jest taka, jak zaproponowano w pliku powyżej:

```
OK ATD*99***1#
```

czyli domyślny (lub ustawiony wcześniej komendą AT+CGDCONT) APN, protokół PPP i CID o numerze 1. Czasami jednak CID=1 może być zarezerwowany dla np. WAP-u po GPRS-ie (a nie pełnego, modemowego dostępu po Internetu). Wówczas może być konieczne ustawienie CID=2 lub CID=3, np.

```
OK ATD*99***2#
```

- Plik /etc/chatscripts/gprs-disconnect-chat o zawartości:

```

# File: gprs-disconnect-chat
#
# send break
exec /usr/sbin/chat -V -s -S
ABORT "BUSY"
ABORT "ERROR"
ABORT "NO DIALTONE"
SAY "\nSending break to the modem\n"
"" "\K"

```

```
" "+++ATH"
SAY "\nPDP context detached\n"
```

- Plik `/etc/ppp/chap-secrets` o zawartości:

```
# Secrets for authentication using PAP
# client server secret IP addresses
"erainternet" * "erainternet" *
```

Warto również stworzyć 2 skrypty (z flagą wykonywalności), które automatycznie ustawią nam przekazane przez Ere adresy DNS-ów:

- Plik `/etc/ppp/ip-up.local` o zawartości:

```
mv /etc/resolv.conf /etc/resolv.conf-pppbackup
cp -f /etc/ppp/resolv.conf /etc/resolv.conf
chmod a+r /etc/resolv.conf
```

- Plik `/etc/ppp/ip-down.local` o zawartości:

```
mv /etc/resolv.conf-pppbackup /etc/resolv.conf
```

Uwaga! W przypadku telefonów Nokia 6310/6310i (może też innych?) stwierdziliśmy objaw zrywania połączenia GPRS po ok. dwóch minutach. Żeby się go pozbyć, należy otworzyć plik `/etc/ppp/options` i sprawdzić czy w pliku jest następująca linia (lub podobna):

```
lcp-echo-failure 4
```

Jeśli jest - należy ją usunąć albo wykomentować. Należy się również upewnić czy opcji tej nie ma w innych plikach odczytywanych przy połączeniu, można to sprawdzić np. programem **grep**:

```
grep -r lcp-echo-failure /etc/ppp
```

Teraz czas na przetestowanie połączenia. Wywołanie komendy:

```
pppd call gprs
```

spowoduje odczytanie przez **pppd** pliku `/etc/ppp/peers/gprs` i uruchomienie zawartych tam instrukcji. Na wyjściu powinniśmy otrzymać komunikaty drukujące kolejne fazy połączenia: komendy AT, wybieranie numeru i otrzymanie adresu IP. Powinny nam zostać również przesłane adresy IP serwerów DNS (ostatni komunikat przed odłączeniem się od terminala). Za obsługą opcja adresów IP serwerów DNS stoi opcja **usepeerdns**, dzięki której adresy IP zostaną umieszczone w pliku `/etc/ppp/resolv.conf`, a następnie przekopiowane przez nasz skrypt `/etc/ppp/ip-up.local` do `/etc/resolv.conf`. Zakończenie połączenia następuje przez wywołanie:

```
killall pppd
```

Ta barbarzyńska z pozoru metoda zakończenia połączenia sprawuje się wyśmienicie, w tym wywołuje skrypty `/etc/ppp/if-down` i `/etc/ppp/if-down.local` (czyli zamyka połączenie w sposób, jaki byśmy oczekiwali).

W przypadku połączenia GPRS istotna jest ilość wysłanych i odebranych bajtów. Informację tę można zdobyć wywołując komendę **pppstats** (przed zakończeniem połączenia!).

### 3.2.15.14.3. Instalacja modemów Sony Ericsson GC75, GC75e, GC85, GC89

W tym podrozdziale zostanie opisana instalacja modemów GPRS Sony Ericsson GC75 i GC75e oraz konfiguracja pppd dla połączenia GPRS IDEA.

Obie karty PCMCIA są standardowo wykrywane. Jeśli nasz modem nie jest wykrywany, powinniśmy się upewnić czy nasze jądro jest odpowiednio skonfigurowane. Przykład poprawnej konfiguracji jądra.

I tak dla:

- Jąder serii 2.4 :

```
General setup --->
[*] Support for hot-pluggable devices
PCMCIA/CardBus support --->
[M] PCMCIA/CardBus support
[*] CardBus support
[*] Databook TCIC host bridge support
[ ] i82092 compatible bridge support
[*] i82365 compatible bridge support
Character devices --->
PCMCIA character devices --->
[M] PCMCIA serial device support
[ ] SyncLink PC Card support
```

- Jąder serii 2.6 :

```
General setup --->
[*] Support for hot-pluggable devices
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
PCCARD (PCMCIA/CardBus) support --->
[M] PCCard (PCMCIA/CardBus) support
[ ] Enable PCCARD debugging
[ ] Enable obsolete PCCARD code
[M] 16-bit PCMCIA support
[*] 32-bit CardBus support
--- PC-card bridges
[M] CardBus yenta-compatible bridge support
[ ] Cirrus PD6729 compatible bridge support
[M] i82092 compatible bridge support
[M] i82365 compatible bridge support
[ ] Databook TCIC host bridge support
Device Drivers --->
Character devices --->
Serial drivers --->
[*] 8250/16550 and compatible serial support
[*] Console on 8250/16550 and compatible serial port
[M] 8250/16550 PCMCIA device support
[ ] 8250/16550 device discovery via ACPI namespace
```

Przy tak skonfigurowanym jądrze po wsunięciu karty PCMCIA do slotu powinniśmy zobaczyć komunikat wyglądający mniej więcej tak:

```
ttyS0 at I/O 0x3f8 (irq = 3) is a 16550A
```

Jeśli korzystamy z X'ów możemy skorzystać z polecenia **dmesg**. Jeśli nasz modem zostanie wykryty np. jako ttyS14, a w /dev nie ma takiego urządzenia należy je utworzyć np.

```
# mknod /dev/ttyS14 c 4 78
```

WAŻNE: W przypadku gdy nasz modem to GC89, musimy wykonać następujące czynności.

Musimy zainstalować setserial, w przypadku Debiana możemy posłużyć się apt-get.

```
# apt-get install setserial
```

Nasza karta jest "dość wybredna" jeśli chodzi o prędkość z jaką będzie chciała z nami rozmawiać. Mamy dwie możliwości:

- 

```
# setserial /dev/ttyS14 spd_shi
# setserial /dev/ttyS14 baud_base 460800
```

Teraz nasz port szeregowy powinien rozmawiać z prędkością 38400 (standardowo ustawiona w minicomie).

- 

```
# setserial /dev/ttyS14 baud_base 460800
```

Teraz nasz port szeregowy powinien rozmawiać z prędkością 230400 (działa w większości przypadków.)

Pamiętajmy o wpisaniu poprawnej prędkości do plików konfiguracyjnych pppd.

Aby uruchomić moduł wlan, musimy skorzystać z ndiswrapper'a. Dokładny opis instalacji ndiswrappera znajdziemy w Sekcja 3.2.14.2. Nasze drivery znajdziemy na dołączonej płycie do SE GC89 (Wlan-2k/semw15.inf).

W tym momencie nasz modem PCMCIA jest gotowy do pracy. Została nam jeszcze konfiguracja pppd.

Jeśli jesteśmy użytkownikami Debiana, możemy pominąć następne rozdziały i zainstalować paczkę edge-gprs udostępnianą przez PRATERM S.A. Konfiguracja jest bardzo prosta i na pewno każdy sobie poradzi, wystarczy odpowiedzieć na kilka pytań.

#### 3.2.15.14.4. Instalacja OptionGlobeTrotter Fusion+

Modem korzysta ze sterowników nozomi, które są obecne w nowszych jądrach (na pewno 2.6.26). Jeśli mamy starszy modem, musimy ściągnąć sterowniki ze strony

[http://www.pharscape.org/3G/nozomi\\_060209.tar.gz](http://www.pharscape.org/3G/nozomi_060209.tar.gz), a następnie je skompilować. Aby tego dokonać, powinniśmy zaopatrzyć się w źródła/nagłówki jądra.

Następnie standardowo **make** i powinniśmy otrzymać plik `nozomi.ko`. W przypadku problemów ze znalezieniem katalogu ze źródłami, możemy użyć opcji **-I/path/to/kernel**. Po załadowaniu modułu nasz modem jest dostępny jako `/dev/noz0 ... /dev/noz3`. Jeśli urządzenia nie zostały automatycznie stworzone, musimy zrobić to ręcznie:

```
mknod /dev/noz0 c 241 0
mknod /dev/noz1 c 241 1
mknod /dev/noz2 c 241 2
mknod /dev/noz3 c 241 3
```

Transmisja danych odbywa się na urządzeniu `/dev/noz0`, jednak w tym samym czasie możemy komunikować się z modemem w `/dev/noz2`. Jest to pewne udogodnienie, ponieważ teoretycznie pozwala na badanie siły sygnału podczas transmisji (czego nie umożliwiają modemy Sony Ericsson GC8X) oraz np. dostęp do skrzynki sms, spisu telefonów itp.

Ostatnim etapem, opcjonalnym jest umieszczenie `nozomi.ko` w katalogu z modułami. Robimy to w zależności od posiadanej przez nas dystrybucji, zazwyczaj `/lib/modules/2.X.X/...`

Jeśli jesteśmy użytkownikami Debiana, możemy pominąć następne rozdziały i zainstalować paczkę `edge-gprs` udostępnianą przez PRATERM S.A. Konfiguracja jest bardzo prosta i na pewno każdy sobie poradzi, wystarczy odpowiedzieć na kilka pytań.

### 3.2.15.14.5. Konfiguracja modemu Huawei E160

Modem Huawei E160 jest automatycznie rozpoznawany i obsługiwany przez współczesne wersja jądra Linuksa. Po podłączeniu modem powinien być widziany jako urządzenie `/dev/ttyUSB0` (lub kolejne, jeśli mamy podłączoną np. przejściówkę USB - RS232). Działająca konfiguracja dla tego modemu podana jest jako przykład w rozdziale Sekcja 3.2.15.14.13.

### 3.2.15.14.6. Konfiguracja modemu ICON225

Modem ICON 225 jest automatycznie rozpoznawany i obsługiwany przez współczesne wersja jądra Linuksa. Po podłączeniu modem powinien być widziany jako urządzenia `/dev/ttyHS0` do `/dev/ttyHS3`, przy czym do właściwej komunikacji używamy tego ostatniego. Poniżej znajduje się przykładowa konfiguracja modemu, zgodnie z konwencją opisaną w rozdziale Sekcja 3.2.15.14.13.

Plik `/etc/ppp/peers/icon225:`

```
crtsets
updetach
lcp-echo-interval 65536
lcp-echo-failure 4
connect "/etc/ppp/icon225-chat"
disconnect "/etc/ppp/icon225-disconnect"
lock
local
defaultroute
usepeerdns
/dev/ttyHS3
noauth
```

Plik `/etc/ppp/icon225-chat` (przykład dla sieci Orange, zobacz też Sekcja 3.2.15.14.11):

```
OP=www.idea.pl
PIN=1111
SLEEP=3

/usr/sbin/chat -v \
TIMEOUT 5 \
ABORT '\nERROR\r' \
ECHO ON \
```

```

SAY      "\r\nSetting PIN - this can fail" \
"        '\rAT' \
OK        'ATZ' \
'OK'      "AT+CPIN=$PIN" \
'OK'      " \
SAY      "\r\nWaiting $SLEEP seconds" && sleep $SLEEP

```

```

/usr/sbin/chat -v \
TIMEOUT  20 \
ECHO      ON \
ABORT     '\nBUSY\r' \
ABORT     '\nERROR\r' \
ABORT     '\nNO ANSWER\r' \
ABORT     '\nNO CARRIER\r' \
ABORT     '\nNO DIALTONE\r' \
ABORT     '\nRINGING\r\n\r\nRINGING\r' \
TIMEOUT  30 \
"         '\rAT' \
OK        'AT+IFC=0,0' \
SAY      'PRE\r' \
OK        "AT+CGDCONT=1,\"IP\", \"$OP\", \"\", 0, 0" \
SAY      'Dialing... ' \
OK        'ATD*99***1#' \
SAY      "\nConnected.\n" \

```

Plik /etc/ppp/icon225-disconnect:

```

/usr/sbin/chat -v \
TIMEOUT  5 \
ECHO      ON \
""        "\K" \
""        "+++ATH0"

```

### 3.2.15.14.7. Konfiguracja telefonu LG GT505 jako modemu

Telefon LG GT505 może być w łatwy sposób wykorzystany jako modem. Po podpięciu telefonu do komputera za pomocą kabla USB, wybieramy tryb PC Suite. Powinno to spowodować widoczność telefonu jako modemu - w logu dmesg pojawi się komunikat:

```

[838134.116082] usb 1-2: new high speed USB device using ehci_hcd and address 38
[838134.263708] usb 1-2: New USB device found, idVendor=1004, idProduct=6091
[838134.263722] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[838134.263728] usb 1-2: Product: GT500
[838134.263733] usb 1-2: Manufacturer: LG Electronics
[838134.263738] usb 1-2: SerialNumber: 3546950362635220
[838134.264660] usb 1-2: configuration #1 chosen from 1 choice
[838134.267204] cdc_acm 1-2:1.1: ttyACM0: USB ACM device
[838134.283571] cdc_wdm 1-2:1.5: cdc-wdm-176: USB WDM device

```

Przygotowujemy skrypty konfiguracyjne - /etc/ppp/peers/lg o treści:

```

debug
230400
nocrtscts
nodeflate
lcp-echo-interval 0
connect "/etc/ppp/lg-chat"
:20.0.0.1
noipdefault
usepeerdns
novj
novjccomp
nobsdcomp
defaultroute
/dev/ttyACM0
noauth

```

oraz /etc/ppp/lg-chat:

```

OP=internet
SLEEP=3
/usr/sbin/chat \
TIMEOUT 20 \
ECHO ON \
ABORT '\nBUSY\r' \
ABORT '\nERROR\r' \
ABORT '\nNO ANSWER\r' \
ABORT '\nNO CARRIER\r' \
ABORT '\nNO DIALTONE\r' \
ABORT '\nRINGING\r\n\r\nRINGING\r' \
TIMEOUT 30 \
" '\rAT' \
OK 'ATE1' \
OK 'AT+IFC=0,0' \
OK "AT+CGDCONT=1,\"IP\", \"$OP\", \"\", 0, 0" \
OK 'ATD*99***1#' \
TIMEOUT 30 \
CONNECT "" \
SAY "\nConnected.\n" \

```

W pliku /etc/network/interfaces (pod Debianem) dodajemy sekcję:

```

iface ppp0 inet ppp
provider lg

```

Połączenie nawiązujemy wydając komendę **ifup ppp0**, wyłączamy przez **ifdown ppp0**. Jeżeli po zerwaniu połączenia nie da się nawiązać go ponownie (w logach są informacje o niemożności otwarcia urządzenia /dev/ttyACM0), pomaga odłączenie telefonu od komputera i ponowne połączenie.

### 3.2.15.14.8. Resetowanie modemów GPRS (GCXX i OGTFusion+)

W przypadku modemów Sony Ericsson GCXX sprawa jest bardzo prosta. Po zaopatrzeniu się w pakiet pcmcia-cs wydajemy po prostu polecenie: **cardctl eject** a następnie **cardctl insert**.

Gdy posiadamy OptionGlobeTrotter Fusion+, sprawa nieco się komplikuje. Gdybyśmy użyli metody z GCXX, nastąpiłby kernel panic. Poprawnym rozwiązaniem jest najpierw zabicie pppd, a następnie usunięcie w odpowiedniej kolejności modułów. I tak:

```
killall pppd
rmmod ppp_async
rmmod ppp_generic
rmmod nozomi
cardctl eject
cardctl insert
modprobe nozomi
```

### 3.2.15.14.9. Logowanie się do sieci GPRS

Aby zalogować się do sieci gprs, musimy w w przypadku Sony Ericsson GCXX podczas komunikacji z modemem wpisać:

```
AT+CFUN=1,1
AT+CPIN=TWÓJ_PIN
```

W przypadku OptionGlobeTrotter Fusion+:

```
AT+CPIN=TWÓJ_PIN
```

### 3.2.15.14.10. Ogólna konfiguracja pppd dla modemów GCXX i OGTFusion+.

Poniżej jest zamieszczony najogólniejszy zestaw plików ppp, który powinien działać zarówno dla GCXX jak i OGTFusion+.

```
/etc/ppp/peers/edge-gprs

debug

# Max speed
230400

# Do not use hardware flow control
nocrtscts

# Don't keep pppd attached to the terminal:
updetach

# Connection options
nodeflate
```

```

lcp-echo-interval 0
# Path to chat script connect/disconnect
connect "/usr/lib/edge-gprs/edge-gprs-connect-chat"

# IP address configuration
:20.0.0.1
noipdefault
usepeerdns

novj
novjccomp
nobsdcomp
defaultroute
# Enter below your path to modem
/dev/ttyS1

/usr/lib/edge-gprs/edge-gprs-connect-chat

/usr/sbin/chat \
TIMEOUT 20 \
ECHO ON \
ABORT '\nBUSY\r' \
ABORT '\nERROR\r' \
ABORT '\nNO ANSWER\r' \
ABORT '\nNO CARRIER\r' \
ABORT '\nNO DIALTONE\r' \
ABORT '\nRINGING\r\n\r\nRINGING\r' \
SAY "Press CTRL-C to close the connection at any stage!" \
TIMEOUT 60 \
" '\rAT' \
OK 'ATE1' \
OK 'AT+CGDCONT=1,"IP","AP_OPERATOR"' \
OK 'ATD*99***1#' \
TIMEOUT 30 \
CONNECT "" \
SAY "\nConnected."

```

Teraz tylko pppd call edge-gprs i powinniśmy uzyskać połączenie.

### 3.2.15.14.11. Operatorzy GPRS.

- ERA - AP\_OPERATORA to „erainternet”. Dodatkowo wpisujemy do /etc/ppp/pap-secrets „erainternet \* erainternet” oraz do /etc/ppp/peers/edge-gprs „user erainternet” oraz „auth”
- IDEA (ORANGE) - AP\_OPERATORA to „www.idea.pl”. Dodatkowo do /etc/ppp/peers/edge-gprs dopisujemy „noauth”.
- PLUS - AP\_OPERATORA to „www.plusgsm.pl”. Dodatkowo do /etc/ppp/peers/edge-gprs dopisujemy „noauth”.
- PLAY - AP\_OPERATORA to „internet”. Dodatkowo do /etc/ppp/peers/edge-gprs dopisujemy „noauth”.

### 3.2.15.14.12. Wymuszanie GPRS/UMTS

W przypadku np. słabego zasięgu możemy chcieć wymusić korzystanie z wolniejszej technologii GPRS zamiast szybszego UMTS. W takich sytuacjach do skryptu chat połączenia musimy dodać (np. przed komendą AT+CGDCONT) jedną z następujących komend:

- Aby wymusić korzystanie z GPRS:  
OK 'AT\_OPSYS=0,2' \
- Aby wymusić korzystanie z UMTS:  
OK 'AT\_OPSYS=1,2' \
- Aby ustawić preferowanie GPRS przed UMTS:  
OK 'AT\_OPSYS=2,2' \
- Aby ustawić preferowanie UMTS przed GPRS:  
OK 'AT\_OPSYS=3,2' \

### 3.2.15.14.13. Konfiguracja połączenia GPRS w systemach Debian

Jeśli mamy w miarę aktualną dystrybucję Debian/Ubuntu, konfiguracja połączenia GPRS powinna być dość prosta i dobrze integrować się z mechanizmem zarządzania połączeniami sieciowymi. Modem powinien być wykryty automatycznie (co można sprawdzić komendą **dmesg**), w razie problemów należy tylko upewnić się że korzystamy z w miarę nowego jądra.

W pliku `/etc/network/interfaces` umieszczamy wpis typu (podana niżej jest przykładowa konfiguracja dla Era GSM i modemu Huawei E160):

```
iface ppp0 inet ppp
provider e160
```

Nazwa `e160` może być dowolnie wybrana, później tylko należy użyć jej konsekwentnie w nazwach skryptów z opcjami PPP i skryptu chat.

Zamiast zmieniać globalne opcje PPP, tworzymy plik `/etc/ppp/peers/e160` o zawartości:

```
# Debug info from pppd:
# You can comment this off, if you don't need more info
debug

# Max speed
#230400

# Use hardware flow control
nocrtscts

# Don't keep pppd attached to the terminal:
updetach

# Connection options
nodeflate

lcp-echo-interval 0
```

```

# Path to chat script connect/disconnect
connect "/etc/ppp/peers/e160-chat"
# we don't need special script for that modem
#disconnect "/etc/ppp/peers/e160-disconnect"

# IP address configuration
:20.0.0.1
noipdefault
usepeerdns

novj
novjccomp
nobsdcomp
defaultroute
/dev/ttyUSB0
user erainternet
noauth

```

Na koniec potrzebujemy jeszcze skryptu chat - /etc/ppp/peers/e160-chat o zawartości:

```

OP=erainternet
PIN=9999
SLEEP=5

/usr/sbin/chat \
TIMEOUT 20 \
ABORT '\nERROR\r' \
ECHO ON \
SAY "\r\nSetting PIN - this can fail" \
" '\rAT' \
OK 'ATZ' \
'OK' "AT+CPIN=$PIN" \
'OK' " \
SAY "\r\nWaiting $SLEEP seconds" && sleep $SLEEP

/usr/sbin/chat \
TIMEOUT 20 \
ECHO ON \
ABORT '\nBUSY\r' \
ABORT '\nERROR\r' \
ABORT '\nNO ANSWER\r' \
ABORT '\nNO CARRIER\r' \
ABORT '\nNO DIALTONE\r' \
ABORT '\nRINGING\r\n\r\nRINGING\r' \
TIMEOUT 30 \
" '\rAT' \
OK 'ATE1' \
OK 'AT+IFC=0,0' \
OK "AT+CGDCONT=1,\"IP\", \"$OP\", \"\", 0, 0" \
OK 'ATD*99***1#' \
TIMEOUT 30 \
CONNECT "" \
SAY "\nConnected.\n" \

```

Przy takiej konfiguracji włączanie i wyłączenie połączenia możemy robić standardowymi komendami **ifup ppp0** oraz **ifdown ppp0**.

#### 3.2.15.14.14. Modemy GPRS w komputerach UNO

Jeśli zauważymy, że nasz modem działa bardzo wolno, albo że nie działa prawie w ogóle (szczególnie w przypadku komputerów UNO), najprawdopodobniej jest to wina przerwania. W przypadku komputerów UNO, musimy w Bios'ie zmienić ustawienia przerywania IRQ 9 z ISA/PNP na PCI/Legacy. Modem powinien zacząć działać z normalną prędkością. Dotyczy to zarówno modemów GCXX, jak i OGTFusion+.

#### 3.2.15.15. Diagnozowanie problemów z pppd

Czasami pojawia się potrzeba dokładnego śledzenia działań wykonywanych przez demona pppd. Jeżeli korzystamy z własnych skryptów 'chat', większość komunikacji z modemem znajdzie się w logach systemowych. Nieco inaczej jest np. w przypadku konfigurowania połączenia SDI.

Aby zmusić demona pppd do wypisywania szczegółów komunikacji z modemem, należy:

- Upewnić się że w pliku `/etc/ppp/options` jest opcja `debug` - jeśli jej nie ma, trzeba dopisać linię ze słowem `debug`.
- Zmodyfikować ewentualnie plik konfiguracyjny demona `syslog` - `/etc/syslog.conf` tak, aby przekierować do logów także komunikaty o poziomie ważności `debug`. Szczegóły zależą od dystrybucji - pod Debianem zwykle komunikaty te domyślnie lądują w `/var/log/debug`. Pod RedHatem i pochodnymi należy dopisać raczej gdzieś z tyłu pliku linię postaci:
 

```
*.debug /var/log/messages
```

 Należy pamiętać, aby po usunięciu problemu przywrócić poprzednią postać pliku.
- Aby zmiany zostały zauważone, należy zrestartować demona `syslogd` lub wysłać do działającego procesu sygnał `SIGHUP`. Jeżeli nie chcemy restartować działającego demona `pppd`, można, zamiast dodawać opcję `debug` do pliku `/etc/ppp/options`, wysłać do działającego procesu sygnał `SIGUSR1` (1). Wysłanie sygnału działa jak przełącznik - włącza i wyłącza opcję `debug`.

#### 3.2.15.15.1. Lista niekompatybilnego sprzętu

- Sony Ericsson GC89 + Przejściówka PCI->PCMCIA + Płyta Główna Intel'a oparta na i915.

### 3.2.16. Konfigurowanie ssh (klient, serwer, tunelowanie)

Program `ssh` (*Secure Shell*) służy do logowania się na zdalne komputery (przez sieć) i zdalnego wykonywania komend. W przeciwieństwie do programów typu `rsh` czy `telnet` oferuje bezpieczną, szyfrowaną komunikację, uniemożliwia więc ewentualne podsłuchiwanie transmisji. Autoryzacja

użytkownika może się odbywać na kilka sposobów. Najprostszym jest autoryzacja przy pomocy podawanego przez użytkownika hasła. Po uruchomieniu komendy:

```
ssh login@host
```

program pyta się o hasło. *login* oznacza tu nazwę użytkownika, a *host* nazwę komputera, np. *kazik@mail.praterm.com.pl*. Często przydatne jest, aby **ssh** nie pytał się o hasło - umożliwia to między innymi wygodne używanie **ssh** z programów nieinterakcyjnych (np. przez system BODAS).

Skonfigurowanie **ssh** wymaga kilku prostych kroków:

- W systemie muszą być zainstalowane odpowiednie pakiety, zawierające między innymi program **ssh** (pod Debianem są to pakiety `openssh-client` dla programu `ssh` i `openssh-server` dla serwera `ssh`). Praktycznie wszystkie współczesne dystrybucje Linuksa zawierają domyślnie te pakiety.
- Następnie można skonfigurować **ssh** tak, żeby nie było konieczne podawanie hasła.

Założmy, że np. użytkownik *janosik* na komputerze *alpha* chce się logować na komputer *beta* jako użytkownik *ondraszek*. W tym celu *janosik* musi sprawdzić, czy w jego katalogu domowym, w podkatalogu `.ssh` znajdują się pliki `id_rsa` i `id_rsa.pub`. Pliki te zawierają klucz prywatny i publiczny. Jeżeli ich nie mamy, to możemy je stworzyć przez wywołanie:

```
ssh-keygen -t rsa
```

Na zadawane przez program pytania należy odpowiadać naciskając po prostu *Enter*, czyli wybierając wartości domyślne.

Plik `.ssh/id_rsa` jest naszym kluczem publicznym - należy go chronić i nikomu nie udostępniać. Natomiast plik `.ssh/id_rsa.pub` kopiujemy na konto użytkownika *ondraszek* na komputerze *beta* i dodajemy do pliku `.ssh/authorized_keys2` - jeżeli tego pliku nie ma, musimy go stworzyć. Sekwencja komend do wykonania mogłaby wyglądać tak:

```
scp .ssh/id_rsa.pub ondraszek@beta:
ssh ondraszek@beta
mkdir -p .ssh
cat id_rsa.pub >> .ssh/authorized_keys2
```

Ważne jest też ustawienie odpowiednich praw zapisu - katalog `.ssh` powinien mieć tylko prawa odczytu i zapisu dla właściciela, a plik `authorized_keys2` dodatkowo prawa odczytu dla innych. Jeżeli uprawnienia będą nieprawidłowe, `ssh` odmówi użycia klucza. Właściwe uprawnienia użytkownik *ondraszek* może ustawić przez wykonanie komend:

```
chmod 0700 ~/.ssh
chmod 0644 ~/.ssh/authorized_keys2
```

Teraz, po wpisaniu przez użytkownika *janosik* na komputerze *alpha* polecenia

```
ssh ondraszek@beta
```

powinno nastąpić zalogowanie na komputer *beta*, bez pytania o hasło.

W razie wystąpienia problemów z autoryzacją trzeba uruchomić `ssh` w trybie `verbose` (`ssh -v`). Program wyświetli wtedy błędy, co znacznie ułatwia rozpoznanie problemu.

Powyższy opis odnosi się do wersji drugiej protokołu SSH. W starszych dystrybucjach Linuksa domyślnie używana była wersja 1 protokołu. Dla wersji tej zamiast pliku `authorized_keys2` używany był plik `authorized_keys`, generowany był klucz typu RSA1 (czyli do generacji klucza wywołanie `ssh-key -t rsa1`), klucz domyślnie umieszczany był w plikach `identity` i `identity.pub`. Używaną wersję protokołu można wymusić przez wpis *Protocol 2,1* w pliku `/etc/ssh/ssh_config` (ważna jest kolejność liczb, można też zostawić tylko jedną liczbę) lub przez

opcję w linii poleceń ssh, np. *-1*. Obecnie używanie wersji 1 protokołu nie jest zalecane, ze względu na mniejsze bezpieczeństwo.

Plikiem konfiguracyjnym dla klienta ssh jest `/etc/ssh/ssh_config`. Oprócz wspomnianej wyżej wersji protokołu możemy ustawić globalnie kilka innych parametrów, między innymi.:

- `StrictHostKeyChecking` - ustawienie, czy klient ssh może automatycznie dodawać hosty do pliku `~/.ssh/known_hosts`. Przyjmuje jedną z trzech wartości: `'yes'`, `'no'` lub `'ask'` (wartość domyślna). Jeżeli ustawimy na `'yes'`, to wpisy będą dodawane automatycznie, `'no'` oznacza, że będziemy musieli wpisy dodawać ręcznie, natomiast `'ask'` oznacza, że ssh sam będzie dodawał wpisy po potwierdzeniu użytkownika.

Uwaga - jeśli dodamy klucz jakiegoś komputera do pliku `known_hosts` i będziemy próbowali połączyć się z innym komputerem (o innym kluczu) używającym z jakiegoś powodu tego samego IP (częste, jeśli np. lokalne adresy IP w różnych sieciach, z których korzystamy się pokrywają), ssh podniesie alarm, podejrzewając że ktoś „podmienił” komputer, na który chcemy się zalogować. W szczególności nie pozwoli na zalogowanie się na taki komputer z użyciem hasła (logowanie po wymianie kluczy mimo ostrzeżenia będzie możliwe). Aby zalogować się za pomocą hasła, musimy ręcznie w edytorze tekstowym usunąć odpowiednią linię z pliku `.ssh/known_hosts`. Każdy klucz trzymany jest w osobnej linii, ssh poda w ostrzeżeniu numer linii, o którą chodzi.

Wartości opcji z pliku konfiguracyjnego można także ustawiać bezpośrednio w linii poleceń, przy wywołaniu program ssh, używając następującej składni:

```
ssh -o StrictHostKeyChecking=no ondraszek@beta
```

Plikiem konfiguracyjnym dla serwera SSH jest plik `/etc/ssh/sshd_config`.

### 3.2.16.1. Tunel zwrotny SSH

Często jest potrzeba do dostania się maszyny „schowanej” za bramką internetową. W przypadku rozwiązań typu NAT lub Masquarading, bezpośrednie połączenie się ze schowanym komputerem z zewnątrz nie jest możliwe. Możliwa jest natomiast inicjalizacja połączenia ssh z docelowego komputera i zalogowanie się na niego przez tak zwany tunel zwrotny.

Tunel ten tworzymy wywołując na docelowym komputerze następujące polecenie:

```
ssh user@server -R 7777:localhost:22
```

Podany numer portu - `7777` - jest przykładowy. Może być to dowolny numer portu nieużywany na komputerze *server*. Takie polecenie spowoduje utworzenie tunelu zwrotnego - na docelowy komputer można się zalogować z komputera *server* przez wydanie polecenia:

```
ssh localhost -p 7777
```

Jeżeli tunel ma być aktywny przez dłuższy czas, warto uruchamiając tunel podać dodatkowe polecenie, którego aktywność zabezpieczy przez zerwaniem połączenia uznanego za nieużywane. Przykładowo:

```
ssh user@server -R 7777:localhost:22 "/bin/sh -c 'while true; do sleep 5; \
echo +; done' "
```

Jeszcze lepszym rozwiązaniem jest użycie programu **autossh** (w Debianie paczka o takiej samej nazwie). Program ten pozwala na uruchomienie tunelu z wykorzystaniem dodatkowego portu monitorującego - zerwane lub zablokowane połączenia są automatycznie wznowiane. Przykładowe wywołanie autossh mogłoby wyglądać następująco:

```
autossh -fN -M 7778 -R 7777:localhost:22 user@server
```

Opcja -f powoduje przejście autossh w tło, opcja -N jest przekazywana do ssh - zakazuje wykonywania komend - tylko przekierowywanie portów (zwiększa to bezpieczeństwo). Numer portu do monitorowania, tu przykładowo 7778, może być również dowolny, ale unikalny.

Oczywiście pozostaje problem, jak na docelowym komputerze, który jest niedostępny, uruchomić tunel. Możemy podyktować odpowiednie polecenie przez telefon, ale na dłuższą metę warto dodać albo odpowiednie wywołanie ssh np. do crontaba, albo uruchomienie autossh przez skrypty startowe systemu.

**Notatka:** Oprogramowanie SZARP zawiera zarówno gotowy skrypt **ssh\_tunnel.sh**, jak i paczkę **szarp-tunnel**, wykorzystującą autossh do uruchomienia tunelu zwrotnego.

### 3.2.16.2. Tunelowanie połączeń X-Windows

Programu ssh można użyć w celu zdalnego uruchamiania programów „iksowych”. W tym celu zarówno klient, jak i serwer muszą mieć włączoną opcję *X11Forwarding* (ustawioną na *yes*). Na serwerze musimy to zrobić w pliku `/etc/ssh/sshd_config` (po zmianie opcji konieczny jest restart sshd), po stronie klienta albo w pliku `/etc/ssh/ssh_config`, albo przez opcję `-X` w linii poleceń:

```
ssh -X user@server
```

Opcja `-x` wyłącza przekierowywanie połączeń X-Windows - może to być przydatne, np. vim w trybie tekstowym uruchamia się szybciej.

Jeżeli logujemy się przez kilka komputerów, wszystkie połączenia „po drodze” muszą mieć włączone przekierowywanie połączeń X-Windows. Najprościej pierwsze połączenie rozpocząć z terminala uruchomionego w trybie graficznym - dzięki temu ssh odpowiednio ustawi nam zmienną środowiskową `DISPLAY`. Wtedy po prostu gdy uruchamiamy aplikację graficzną, jej okno pojawi się na naszym lokalnym ekranie.

Należy zwrócić uwagę, że protokół X11 zaprojektowany był do wykorzystywania w sieci lokalnej i praca przez wolne łącze jest bardzo nieefektywna, czasami wręcz niemożliwa. Warto więc rozważyć wykorzystanie protokołu VNC. Opis tunelowania VNC przez ssh znajduje się w Sekcja 3.2.16.3.

### 3.2.16.3. Tunelowanie połączeń VNC

Program ssh można także wykorzystać do tunelowania protokołu VNC. Jeśli na docelowym komputerze serwer VNC wykorzystuje port 5900, to logujemy się na niego podając do ssh opcję `-L 5900:localhost:5900`:

```
ssh -L 5900:localhost:5900 user@host
```

Jeśli musimy logować się „na kilka razy” (przez kilka komputerów pośrednich), opcję tą podajemy za każdym razem. Mamy teraz uruchomiony tunel dla połączeń VNC.

Na docelowym komputerze uruchamiamy serwer VNC. Przykładowo, aby podłączyć się do działającej sesji użytkownika *user*, musimy mieć na zdalnym komputerze zainstalowany pakiet *x11vnc* i wydać polecenie:

```
x11vnc -localhost -auth /home/user/.Xauthority
```

Opcja *-auth* ze ścieżką do pliku *.Xauthority* użytkownika jest potrzebna tylko jeżeli wydajemy polecenie jako inny użytkownik niż ten, do którego sesji chcemy się podłączyć. Ma to sens głównie dla użytkownika *root*, gdyż inni i tak nie będą mieli praw odczytu do pliku *.Xauthority*.

Jeżeli chcemy się podłączyć do sesji menedżera logowania (np. *kdm*), czyli w sytuacji jeśli żaden użytkownik nie jest zalogowany, parametr do opcji *-auth* powinien wskazywać na plik „Cookie” menedżera logowania, np. */var/run/xauth/XXXX*, gdzie zamiast *XXXX* występuje ciąg losowych znaków. Lokalizację pliku można często zobaczyć przez wydanie polecenia:

```
ps axw | grep auth
```

Ostatnim krokiem jest uruchomienie lokalnie klienta VNC, podając adres lokalny i numer portu 5900 - *localhost:5900*. Na przykład w przypadku programu **krdc** adres ten musimy wpisać po uruchomieniu programu, inni klienci VNC mogą wymagać podania go w linii poleceń.

### 3.2.17. Przekierowywanie portów

Czasem, w przypadku gdy serwer SZARP’a znajduje się za bramką dokonującą operacji maskowania adresów, nie posiada on numeru IP dającego możliwość podłączenia się do niego z zewnątrz (tzn. z internetu). Wtedy, w przypadku potrzeby dostępu do jego usług ze świata (np. *netpard*), może okazać się konieczne wykorzystać mechanizm przekierowania portów.

#### 3.2.17.1. Założenia wstępne

Założmy, że mamy następujący układ:

```
serwer SZARPA>-----<bramka>-----[INTERNET]-----<zdalny klient
```

i chcemy mieć dostęp do *netpard*’a na serwerze SZARP z naszego zdalnego klienta. Dane:

- Adres maszyny z SZARP’em: 192.168.100.24
- Numer portu *netpard*’a: 2390
- Bramka jest maszyną działającą pod kontrolą systemu unix-podobnego (np. *linux*)

### 3.2.17.2. Instalacja i uruchomienie

Uwaga: wszystkie te operacje wykonujemy na bramce.

Przykładowym programikiem mogącym dokonywać zadawanej operacji jest `redir`. Jego źródła można ściągnąć np. z `ftp://sunsite.icm.edu.pl/pub/linux/debian/pool/main/r/redir/redir_2.1.orig.tar.gz` rozpakowujemy i kompilujemy:

```
tar xzf redir_2.1.orig.tar.gz
cd redir-2.1
make
```

jeśli posiadamy uprawnienia administratora (choć nie jest to wymagane) wykonujemy jeszcze:

```
make install
```

Teraz wystarczy tylko uruchomić `redir'a`:

```
redir\
  --lport=<port, na którym będzie dana usługa dostępna na zewnątrz>\
  --cport=<port na serwerze wewnętrznym, na którym dostępna jest potrzebna usługa>\
  --caddr=<adres serwera w sieci lokalnej, którego port udostępniamy na zewnątrz>\
  --debug
```

Przykładowo dla danych w naszym przykładzie będzie to:

```
redir --lport=2930 --cport=2930 --caddr=192.168.100.24 --debug
```

Jeśli chcemy zapewnić automatyczne uruchamianie w tle `redir'a`, to możemy skorzystać z następującej sztuczki: Edytujemy `crontaba`:

```
crontab -e
```

i dodajemy następującą linię:

```
*/15 * * * * screen -mdS redir <ścieżka do redir'a>redir --lport=2930 --cport=2930 \
--caddr=192.168.100.24 --debug
```

## 3.2.18. Udostępnianie Internetu komputerom w sieci lokalnej: NAT/IP Masquerading

Często komputer ze skonfigurowanym dostępem do Internetu jest używany przez komputery w jego sieci lokalnej jako pośrednik w dostępie do Internetu. Jest to rozwiązanie bardzo popularne, gdyż pozwala korzystać wielu osobom z tego samego publicznego adresu IP. Mechanizm, który pozwala korzystać z tej funkcjonalności, jest nazywany NAT-em (ang. *Network Address Translation*) lub maskowaniem IP (ang. *IP masquerading*). Schemat systemu wygląda jak poniżej:

```
Internet - Publiczny IP=Host=Prywatny IP - Komputery wewnątrz sieci
```

Jak widać, host-pośrednik ma 2 interfejsy (2 adresy IP): jeden do sieci publicznej (Internetu) a drugi do sieci prywatnej. Poniżej opisano sposoby konfiguracji NAT-a na hoście-pośredniku dla systemów Debian

i pochodnych od RedHata. Reszta komputerów w sieci powinna jako adres bramki ustawić prywatny adres IP hosta-pośrednika.

## Debian

Pod Debianem jest wiele sposobów konfigurowania systemu NAT. Najprostszy polega na zainstalowaniu pakietu o nazwie *ipmasq*. Najczęściej nie są potrzebne żadne modyfikacje konfiguracji - interfejs zewnętrzny ustalany jest na podstawie adresu bramki ustawionej na hoście-pośredniku. Automatycznie obsługiwane są sytuacje podnoszenia i wyłączenia interfejsów, należy tylko podczas instalacji wybrać opcję rekonfiguracji firewalla przy tworzeniu połączenia PPP. Więcej informacji można znaleźć między innymi na stronach manuala do **iptables** i **ipmasq**.

## RedHat 6.2

W RH 6.2 wykonujemy następujące komendy w celu inicjalizacji NAT-a (są one wykonywane ręcznie i jednorazowo):

```
ipchains -P forward DENY
ipchains -A forward -i ppp0 -j MASQ
ipchains-save > /etc/sysconfig/ipchains
```

Następnie do pliku `/etc/rc.d/rc.local` dodajemy na końcu następujące polecenia:

```
echo -n "1" > /proc/sys/net/ipv4/ip_forward
ipchains-restore < /etc/sysconfig/ipchains
```

## RedHat 7.x, 8.x, Fedora, Aurox itd

W systemach RH 7.x zarzucono używanie pakietu **ipchains**, zamiast niego wprowadzono nowszy mechanizm **iptables** (kernel 2.4.x). Należy więc zamiast powyższych działań stworzyć ręcznie plik `/etc/sysconfig/iptables` o zawartości (skrypt ten dodatkowo ustawia podstawową konfigurację filtra pakietów) (zakładamy, że `ppp0` jest interfejsem „na zewnątrz”, zaś `eth0` - interfejsem „na sieć lokalną”):

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o ppp0 -j MASQUERADE
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:allowed_in - [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i ! ppp0 -m state --state NEW -j ACCEPT
-A INPUT -m state --state NEW -j allowed_in
-A INPUT -i ! ppp0 -p udp --dport 177 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i ! ppp0 -m state --state NEW -j ACCEPT
-A FORWARD -p udp --dport 53 -j ACCEPT
-A FORWARD -p udp --dport 123 -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m state --state NEW -j ACCEPT
-A OUTPUT -p udp --dport 123 -j ACCEPT
```

```
-A allowed_in -p icmp --icmp-type echo-request -j ACCEPT
-A allowed_in -p tcp --dport 22 -j ACCEPT
-A allowed_in -p udp --dport 123 -j ACCEPT
COMMIT
```

Następnie do pliku `/etc/rc.d/rc.local` dodajemy na końcu następujące polecenia:

```
echo -n "1" > /proc/sys/net/ipv4/ip_forward
iptables-restore < /etc/sysconfig/iptables
```

### 3.2.19. Konfiguracja lokalnego serwera poczty (SMTP)

Operacja ta jest przydatna w przypadku, gdy w sieci lokalnej chcemy mieć możliwość wysyłania poczty na świat, a z jakiś powodów nie chcemy korzystać z zewnętrznych serwerów poczty wychodzącej. Poniższy opis instalacji oraz konfiguracji serwera dotyczy będzie exim'a v4.

#### 3.2.19.1. Instalacja

Musimy zaopatrzyć się w odpowiednie pakiety.

```
apt-get install exim4-base exim4-daemon-heavy exim4-config
```

Warto również zaopatrzyć się w filtr spam'u oraz program antywirusowy.

```
apt-get install spamassassin clamav clamav-base clamav-daemon clamav-freshclam
```

Pakiet `clamav-freshclam` jest odpowiedzialny za automatyczną aktualizację bazy sygnatur wirusów z internetu.

W trakcie instalacji pakietów exim'a mogą zostać zadane pytania, ich opis znajdziemy w następnym rozdziale.

#### 3.2.19.2. Konfiguracja serwera

Najlepiej po zainstalowaniu pakietów wydać polecenie:

```
dpkg-reconfigure exim4-config
```

W trakcie konfiguracji będziemy musieli odpowiedzieć na następujące pytania:

- Podzielić konfigurację na małe pliki? - Odpowiadamy tak. Taka konfiguracja jest łatwiejsza w utrzymaniu.
- Typ konfiguracji - Wybieramy ośrodek internetowy, chyba że cel jaki chcemy osiągnąć jest inny.
- Nazwa pocztowa systemu - np `example.com.pl`
- Adresy IP na których system będzie nasłuchiwał połączeń? - naciskamy enter, spowoduje to że system będzie nasłuchiwał na wszystkich skonfigurowanych adresach w systemie.

- Inne systemu docelowe dla których poczta jest akceptowana - np  
mail.example.com.pl:example.com.pl:poczta.exapmle.com.pl
- Domeny dla których dozwolone jest przekazywanie poczty - zostawiamy puste
- Systemy dla których przekazujesz pocztę - zostawiamy puste, chyba ze chcemy inaczej
- Utrzymywać ilość zapytań DNS na minimalnym poziomie? - odpowiadamy NIE

Pliki konfiguracyjne exim'a sa podzielone na następujące sekcje:

- main - podstawowe ustawienia
- acl - odrzucanie i akceptowanie poczty, filtrowanie (spam, wirusy etc...)
- routers - reguły kierowania poczty do odpowiednich transporterów (podobnie jak route)
- retry - ponawianie
- rewrite - reguły do przepisywania nagłówek
- auth - autoryzacja
- transports - sposoby doręczania poczty przekierowanej przez router'y

Następnie musimy włączyć filtrowanie poczty pod kątem spam'u i wirusów. W tym celu edytujemy najpierw plik `/etc/exim4/conf.d/main/02_exim4-config_options` i ustawiamy zmienną **av\_scanner** na

```
av_scanner = amavisd:/var/lib/amavis/amavisd-new-milter.sock
```

Kolejna zmienna **spamd\_address** powinna być ustawiona na

```
spamd_address = 127.0.0.1 783
```

To są wszystkie zmiany jakich musimy dokonać w tej sekcji

W sekcji AUTH, a dokładniej w pliku `/etc/exim4/conf.d/30_exim4-config_examples` mamy przykłady autoryzacji użytkowników tak by nasz serwer nie był "open relay". Możemy odszukać interesujący nas przykład lub wpisać :

```
plain_saslauthd_server:
    driver = plaintext
    public_name = PLAIN
    server_condition = ${if saslauthd{${2}}{${3}}{1}{0}}
    server_set_id = $2
    server_prompts = :
    .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
    server_advertise_condition = ${if eq{${tls_cipher}}{}}{*}}
    .endif

login_saslauthd_server:
    driver = plaintext
    public_name = LOGIN
    server_prompts = "Username:: : Password::"
    server_condition = ${if saslauthd{${1}}{${2}}{1}{0}}
```

```
server_set_id = $1
.ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
server_advertise_condition = ${if eq{$tls_cipher}{*}}{*}}
.endif
```

Podana konfiguracja wymaga zainstalowanych kilku pakietów:

```
apt-get install sasl2-bin libsasl2-modules
```

Same wpisy odnośnie spamassassina i clamav nie wystarczą. W sekcji ACL, dokładnie w pliku `/etc/exim4/conf.d/acl/40_exim4-config_check_data` musimy dopisać :

```
deny
  malware = *
  message = This message was detected as possible malware ($malware_name).

deny
  spam = nobody
  message = X-Spam_score: $spam_score
```

Po wykonaniu wszystkich zmian, aby zostały one wczytane musimy wykonać następujące polecenia

```
update-exim4.conf
/etc/init.d/exim restart
```

### 3.2.20. Konfiguracja Window Makera

Window Maker jest menedżerem okien mogącym zastąpić KDE. W porównaniu z nim charakteryzuje się dużo większą szybkością działania i mniejszym zapotrzebowaniem na zasoby komputera. Warto więc rozważyć jego stosowanie w instalacjach, gdzie pojawiają się problemy z wydajnością stosowanego sprzętu lub przepustowością sieci (zwłaszcza przy stosowaniu X terminali). Poza tym jest stabilniejszy od KDE w sytuacji, gdy korzysta z niego na raz wielu tak samo zalogowanych użytkowników (sytuacja taka prowadzi często do załamania stabilności KDE w wersji 3). Ma także logiczny, łatwy do edycji schemat plików konfiguracyjnych.

W porównaniu do przestarzałego już IceWm'a Window Maker nie wymaga specjalnych zasobów polonizacyjnych, oferuje wsparcie dla mechanizmów wykorzystywanych przez programy napisane dla KDE lub Gnome, poza tym dobrze integruje się z kdm. Wadą jest wygląd i zachowanie odbiegające nieco od KDE czy desktopu Windows. Po przyzwyczajeniu się może się jednak podobać i okazuje się całkiem wygodny.

WindowMaker jest dostępny standardowo w dystrybucjach RedHata od wielu wersji, nie wymaga więc oddzielnej instalacji.

Poniższy opis nie obejmuje wszystkich zagadnień związanych z konfiguracją WindowMakera, skupia się na elementach potrzebnych w jego współdziałaniu z systemem SZARP.

Aby umożliwić użytkownikom korzystanie z WindowMakera jako alternatywnego menedżera okien, należy wykonać kilka prostych kroków. Zakładam, że system uruchamiany jest w runlevelu 5 i kdm jest

skonfigurowany w sposób typowy dla instalacji systemu SZARP. Wystarczy zalogować się jako root, uruchomić **Centrum Sterowania**, po czym wybrać **System, Menadżer Logowania** i zakładkę **Sesje**. W polu obok napisu **Typ sesji** wpisujemy *windowmaker* lub *wmaker* (to musi być nazwa programu wykonywalnego menedżera okien, różna w różnych wersjach) i naciskamy przycisk **Dodaj**. Innym sposobem jest ręczna edycja pliku */etc/kde/kdm/kdmrc* i dodanie wpisu *windowmaker* lub *wmaker* do parametru *SesionsTypes*. Powyższy sposób działa dla RedHatów 7.X i późniejszych oraz Debiana. Dla wcześniejszych wersji należy zadbać, aby *windowmaker* był uruchamiany w pliku *.Xsessions* odpowiednim dla używanego menedżera logowania (np. *kdm* lub *xdm*).

Od tej pory użytkownicy przy logowaniu przez *kdm* będą mieli dostępny wśród typów sesji typ *windowmaker*. Należy zwrócić uwagę, że *kdm* zawsze loguje użytkownika domyślnie z ostatnio użytym typem sesji, a więc po jednokrotnym użyciu *WindowMakera* będzie on używany domyślnie, dopóki użytkownik przy starcie nie wybierze innego typu sesji.

Rzeczą którą możemy chcieć skonfigurować w *WindowMakerze* jest menu aplikacji pojawiające się po kliknięciu prawym przyciskiem myszki. Istnieje specjalny program do konfiguracji tego menu, ale lepiej i prościej jest chyba robić to bezpośrednio przez edycję pliku.

Jeżeli chcemy zmienić domyślnie instalowane dla każdego użytkownika menu należy wyedytować plik */usr/X11R6/share/WindowMaker/plmenu.pl* (położenie tego pliku może być inne dla wcześniejszych wersji). Format tego pliku jest bardzo prosty, składa się z zagnieżdżonych nawiasów opisujących kolejne poziomy menu (wtedy po nazwie menu i przecinku występuje nawias z zawartością podmenu) lub komendy do wykonania (wtedy po przecinku mamy słowo *Exec*, przecinek a potem w cudzysłowie komendę do wykonania).

Oto przykładowa zawartość pliku konfiguracyjnego:

```
("WindowMaker",
  ("Informacja",
    ("Informacja o WMaker...", INFO_PANEL),
    ("Legalność...", LEGAL_PANEL),
    ("Konsola Systemu", EXEC, "xconsole"),
    ("Obciążenie Systemu", EXEC, "xosview || xload"),
  ),
  ("Konfiguracja",
    ("Edycja menu", EXEC, "xterm -T 'Edycja menu' -e vi \
$HOME/GNUstep/Library/WindowMaker/menu"),
    ("Ustawienie fontów", EXEC, "xterm -T 'Ustawienie fontów' -e vi \
$HOME/GNUstep/Defaults/WMGLOBAL"),
  )
)
```

Przy pierwszym użyciu *WindowMakera* przez danego użytkownika w jego katalogu domowym tworzony jest podkatalog *GNUstep/Defaults*. W katalogu tym znajduje się plik *WMRootMenu* z konfiguracją menu w podanym powyżej formacie. Alternatywnie, menu w starszym, mniej eleganckim, ale bardziej elastycznym formacie, może być umieszczone w innym pliku. Ścieżkę do niego podajemy w cudzysłowie w pliku *WMRootMenu*.

W pakiecie z *WindowMakerem* znajduje się program *wmenukde.pl*, służący do konwersji istniejącego menu KDE do postaci używanej przez *WindowMakera*. Np. aby stworzyć plik zawierający menu systemu SZARP, należy wydać komendę **wkdemenu.pl -d /usr/share/applnk/SZARP** i przekierować jej wyjście do pliku, np. */etc/szarp/wmszarp.menu*. Następnie w katalogach domowych użytkowników należy w plikach *GNUstep/Default/WMRootMenu* umieścić w cudzysłowie ścieżkę do pliku, np. *"~/GNUstep/Library/WindowMaker/menu"*, w którym należy umieścić wywołanie podmenu w postaci:

```
("External Menu", OPEN_MENU, "/etc/szarp/wmszarp.menu"),
```

Wersja skryptu `wkdemenu` rozprowadzana w dystrybucjach do RedHata 7.3 włącznie (a może i później) posiada sporo błędów. Dlatego zaleca się używanie zmodyfikowanej wersji, dostępnej w katalogu `script` w dystrybucji SZARPa.

WindowMaker automatycznie wykrywa zmiany w swoich plikach konfiguracyjnych, także automatycznie te zmiany zapisuje. Może to powodować problemy w sytuacji, jeśli jest uruchomionych kilka kopii programu dla tego samego użytkownika (sytuacja taka jest częsta jeśli na kilku terminalach z Windows użytkownicy korzystają z tego samego użytkownika na serwerze). W takiej sytuacji kolejne kopie programu `wmaker` powinny być uruchamiane z parametrem `--static`. Rozwiązaniem jest skopiowanie pliku `/usr/bin/wmaker` do `/usr/bin/wmaker.bin` i utworzenie skryptu `/usr/bin/wmaker` o następującej zawartości:

```
#!/bin/bash

[ -d $HOME/GNUstep ] || wmaker.inst

if ps axw | grep 'wmaker.bin' ; then {
/usr/bin/wmaker.bin --static "$@"
} else {
/usr/bin/wmaker.bin "$@"
} fi
```

Instalację menu WindowMakera zawierającego odpowiednie pozycje dla systemu SZARP i zainstalowanie powyżej opisanego skryptu w systemie RedHat 7.3 może być wykonane także przez wywołanie komendy `make wmmenu` w podkatalogu `resources` w katalogu ze źródłami SZARPa.

Więcej informacji można znaleźć w plikach w katalogu `/usr/share/doc/WindowMaker` i `/usr/X11R6/share/WindowMaker` lub odpowiednich (dokładne ścieżki mogą się trochę różnić zależnie do wersji).

### 3.2.21. Synchronizacja czasu

Żeby skonfigurować (x)ntp na serwerze, należy mieć zainstalowany serwer ntp i stworzyć plik `/etc/ntp.conf` z zawartością:

```
server 62.233.142.85 prefer

server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10

driftfile /etc/ntp/drift
authenticate no
```

oraz plik `/etc/ntp/step-tickers` z zawartością:

```
62.233.142.85
```

..i uruchomić demona (x)ntpd. Jeśli występują jakiegokolwiek problemy, chcemy poeksperymentować lub ustawić niestandardowe parametry, to należy przeczytać wszystko co jest poniżej.

W przypadku dystrybucji Debian odpowiednie pakiety mają nazwy `ntp-server` i `ntpdate`, a pliki konfiguracyjne to `/etc/ntp.conf` i `/etc/default/ntpdate`. Domyślna konfiguracja korzysta z

serwerów dostępnych pod adresem *pool.ntp.org*, jeżeli chcemy korzystać z innego serwera, należy zastąpić podany adres adresem pożądanego serwera.

Produkowane obecnie komputery są wyposażone w podtrzymywane bateryjnie zegary czasu rzeczywistego (RTC - Real Time Clock). Zwykle jednak ich dokładność pozostawia wiele do życzenia (potrafią "rozjeżdżać" się w tempie sekundy na dobę). Ręczne ustawianie ich i pilnowanie jest uciążliwe i problematyczne (nie każdy przecież ma dokładnie chodzący zegarek).

Rozwiązaniem tego problemu jest wykorzystanie NTP - Network Time Protocol. Jest to protokół pozwalający na synchronizację zegarów komputerów podłączonych do Internetu. Jego działanie opiera się na istnieniu kilku "wzorcowych" komputerów, których zegary synchronizowane są z zegarami atomowymi, drogą radiową lub satelitarną. Pozostałe komputery synchronizują się od nich i od kolejnych pośredników, tworząc strukturę drzewiastą. Zegary komputerów nie tylko są przestawiane, ale także przyspieszane / opóźniane, aby same „trzymały dokładność”.

W Polsce znajduje się jeden pośrednik "drugiego poziomu" (tzw. server stratum 2). Jest to serwer [www.cyf-kr.edu.pl](http://www.cyf-kr.edu.pl) (IP 149.156.4.11). Przy synchronizowaniu kilku komputerów (np. firmowej sieci) zaleca się wydzielenie jednego komputera, który będzie synchronizował się z cyf-kr, pozostałe zaś komputery powinny się synchronizować już z tym wewnętrznym serwerem. Do dobrych obyczajów należy wysłanie przed rozpoczęciem synchronizacji z danym serwerem mail'a do jego administratora.

Typowa konfiguracja "na ciepłowni", to kilka komputerów (serwerów i terminali), z których jeden podłączony jest do Internetu przez łącze stałe lub modem. Ten komputer powinien się synchronizować z jakimś zewnętrznym źródłem - najlepiej z głównym serwerem Pratermu.

Ten wydzielony komputer z dostępem do Internetu nazwijmy wewnętrznym serwerem (dla odróżnienia od zewnętrznego, czyli np. [praterm.com.pl](http://praterm.com.pl)), zaś pozostałe komputery w sieci to klienci.

Zarówno na serwerze, jak i na klientach, musi być zainstalowany odpowiedni pakiet o nazwie albo `ntp`, albo `xntp`. Jeżeli go nie ma, trzeba go doinstalować. W dystrybucji RedHat znajduje się zwykle na płycie "contrib".

Teraz przystępujemy do konfiguracji klientów. Pierwszym krokiem jest edycja pliku `/etc/ntp.conf`. Powinien on mieć na wszystkich klientach następującą zawartość:

```
server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10

server X.X.X.X prefer

driftfile /etc/ntp/drift
authenticate no
```

X.X.X.X zastępujemy numerem IP naszego planowanego wewnętrznego serwera. Pozostałe linie należy wykasować lub lepiej wykomentować.

Następnie w katalogu `/etc/ntp` tworzymy plik o nazwie `step-tickers`, zawierający tylko IP wewnętrznego serwera. Będzie on wykorzystywany do ustawiania czasu przy starcie systemu.

Ostatnim krokiem jest upewnienie się, że w katalogu ze skryptami startowymi (zwykle `/etc/rc.d/rc5.d`) jest link na plik `/etc/rc.d/init.d/ntp` lub `/etc/rc.d/init.d/xntp`.

Teraz przystępujemy do konfigurowania naszego wewnętrznego serwera. Jeżeli ma on stały dostęp do Internetu, to konfiguracja wygląda tak samo jak na kliencie, z tym że zamiast swojego IP, zarówno w pliku `/etc/ntp.conf` jak i `/etc/ntp/step-tickers`, wstawiamy IP zewnętrznego serwera (np. [mail.praterm.com.pl](http://mail.praterm.com.pl)) i jest to koniec konfiguracji.

Sprawa jest bardziej skomplikowana, jeżeli serwer ma dostęp do internetu przez modem i PPP, a nie SDI. Wówczas nie tworzymy na nim pliku `/etc/ntp/step-tickers`, a z pliku `/etc/ntp.conf` usuwamy (wykomentowujemy) linijkę

```
server X.X.X.X prefer
```

Teraz mamy serwer, z którym inne mogą się synchronizować, ale on sam nie ma zewnętrznego źródła synchronizacji. Synchronizacja będzie robiona okresowo, po połączeniu z Internetem, za pomocą programu `ntpdate`.

Jego wywołanie można wstawić albo do pliku `/etc/ppp/ip-up.local`, albo zapisać w pliku konfiguracyjnym SZARP'a przy wywołaniu BODAS'a, po nawiązaniu połączenia PPP.

Ponieważ jednak program `ntpdate` odmówi działania, jeżeli na danym komputerze działa demon `ntp/xntp`, trzeba go najpierw wyłączyć a potem ponownie wyłączyć. Cała linia komend będzie więc miała następującą postać:

```
/etc/rc.d/init.d/ntpd stop; ntpdate -t 3 -s mail.praterm.com.pl; \  
/etc/rc.d/init.d/ntpd start
```

Jeżeli używamy demona `xntpd`, należy oczywiście zamienić wywołania `ntpd` na `xntpd`. Podana nazwa serwera jest przykładowa. Powyższą komendę wpisujemy albo do pliku `/etc/ppp/ip-up.local`, albo jako część parametru "command\_line" odpowiedniej sekcji BODASa w pliku `/etc/szarp/szarp.cfg`.

Uruchomienie całości zaczynamy od uruchomienia demona `ntpd/xntpd` na serwerze wewnętrznym. Jeżeli mamy do czynienia z połączeniem przez PPP, najpierw należy dokonać wstępnej synchronizacji. Jeśli wywołanie programu `ntpdate` wstawiliśmy do `/etc/ppp/ip-up.local`, to wystarczy połączyć się z Internetem. Jeżeli zaś chcemy wywoływać synchronizację za pomocą `meanera`, to po połączeniu trzeba uruchomić ręcznie `ntpdate`, na przykład przez komendę:

```
ntpdate -b mail.praterm.com.pl
```

Powinniśmy otrzymać komunikat typu:

```
ntpdate[1243]: step time server mail.praterm.com.pl offset 48.340123 sec
```

Kolejnym krokiem, już niezależnie od tego czy mamy stałe połączenie z siecią, jest uruchomienie demona `ntp/xntp` na wewnętrznym serwerze przez komendę:

```
/etc/rc.d/init.d/ntpd start
```

(ew. `xntpd` zamiast `ntpd`).

I ostatnie co pozostało, to uruchomienie demonów `ntpd/xntpd` na komputerach - klientach. Normalnie będzie się to odbywało przy starcie komputerów.

Prawidłowe działanie całego mechanizmu można śledzić przez wpisy w pliku `/var/log/messages`. Jest przy tym normalne, że przez parę minut od startu demona na wewnętrznym serwerze synchronizacja

nie będzie się udawała - demon musi się „rozgrzać”. Po tych paru minutach możemy się już cieszyć milisekundową precyzją zegarów komputerów w całej sieci.

Programy ntpd/xntpd i ntpdate nie mają swoich stron manuala, dokumentacja całego pakietu znajduje się w formacie HTML w katalogach z dokumentacją, zwykle `/usr/share/doc`.

## 3.2.22. Konfiguracja drukowania

Drukowanie pod Linuxem jest obecnie obsługiwane przez system o nazwie *CUPS* - Common Unix Printing System. Niniejszy rozdział opisuje:

- Instalację CUPS'a Sekcja 3.2.22.1
- Konfigurowanie drukarki Sekcja 3.2.22.2
- Konfigurowanie drukarki za pomocą Menedżera Drukowania KDE Sekcja 3.2.22.3
- Udostępnianie drukarek dla komputerów z Windows Sekcja 3.2.22.4

### 3.2.22.1. Instalacja systemu CUPS

System CUPS jest dostępny prawdopodobnie we wszystkich obecnych dystrybucjach Linuksa. Poza samym demonem CUPS (w dystrybucji Debian jest to pakiet `cupsys`) warto zainstalować (w nawiasach nazwy pakietów Debiana):

- Programy klienckie - `lp`, `lpstat` itp. (`cupsys-client`).
- Programy klienckie „BSD” - `lpr`, `lpq`, `lprm` (`cupsys-bsd`).
- Sterowniki dla drukarek, w tym:
  - baza sterowników drukarek (`foomatic-db` `foomatic-filters` `foomatic-filters-ppd`),
  - sterowniki dla drukarek HP (`hpijs` `hplib-base` `hpoj` `foomatic-db-hpijs`),
  - sterowniki z projektu Gutenprint (dawniej Gimp-Print) (`cupsys-driver-gutenprint` `foomatic-db-gutenprint`),
  - sterownik dla drukarek GDI HP wykorzystujących protokół PPA (np. DeskJet 7XX, DeskJet 820, DeskJet 100) (`pnm2ppa`),
  - sterownik dla drukarek GDI wykorzystujących protokół ZjStream (np. modele Minolta magicolor 2300DL lub HP LaserJet 1000) (`foo2zjs`),
  - BlueZ - sterownik dla drukarek łączących się przez Bluetooth (`bluez-cups`).
- Dodatkowe narzędzia:
  - podsystem drukowania dla KDE (`kdeprint`),
  - `printconf` - narzędzie automatycznie rozpoznające i instalujące, drukarki podłączone bezpośrednio do komputera przez port USB lub równoległy (`printconf`),
  - Narzędzia dla drukarek Epson Stylus - czyszczenie głowic, sprawdzanie poziomu tuszu (`escputil`).

Większość potrzebnych pakietów/narzędzi jest obecna w dystrybucji Debian/SZARP pod wybraniem opcji *cups* w oknie *Dodatki* podczas instalacji. Możemy też zainstalować brakujące pakiety, np. za pomocą komendy **apt-get install nazwa\_pakietu**.

Po instalacji CUPS'a warto go uruchomić - np. komendą:

```
/etc/init.d/cupsys start
```

(pod Debianem) lub analogiczną dla innych dystrybucji.

### 3.2.22.2. Konfiguracja drukarki

System CUPS umożliwia skonfigurowanie zarówno drukarki lokalnej, sieciowej, jak i udostępnionej drukarki Windows - niniejszy rozdział opisuje pokrótce wszystkie te możliwości. Konfiguracji drukarki pod CUPS'em można dokonać na bardzo wiele sposobów - istnieje wiele narzędzi graficznych ułatwiających to zadanie. W tym rozdziale opisujemy sposób najbardziej uniwersalny i dostępny na każdym komputerze, także podczas pracy zdalnej. Wygodny interfejs do CUPS'a udostępniany przez środowisko KDE jest opisany w rozdziale Sekcja 3.2.22.3.

Pierwsza rzecz, którą musimy zrobić, to połączyć się z serwerem CUPS. Zakładamy, że CUPS jest zainstalowany i uruchomiony. Uruchamiamy więc dowolną przeglądarkę WWW i łączymy się z serwerem CUPS, wpisując w pasku adresu przeglądarki adres serwera z CUPS'em i podając po dwukropku numer portu - 631. Jeżeli przeglądarka jest uruchomiona na tym samym komputerze, na którym działa CUPS, to adres będzie miał postać `http://localhost:631`.

Jeżeli prawidłowo wpisaliśmy adres i CUPS jest uruchomiony, to po chwili zobaczymy ekran z opcjami administracyjnymi. W celu dodania drukarki wybieramy link *Manage Printers*. Jeżeli na komputerze są już skonfigurowane jakieś drukarki, to pojawi się ich lista z krótkim opisem, informacją o stanie, adresie, oraz przyciskami pozwalającymi wydrukować stronę testową (*Print test page*), zatrzymać drukarkę (*Stop printer*), usunąć wszystkie zadania wydruku (*Reject jobs*), zmodyfikować typ/adres drukarki (*Modify Printer*), zmienić opcje drukarki takie, jak jakość druku (*Configure printer*), wreszcie usunąć drukarkę z systemu (*Delete printer*) lub też ustawić ją jako domyślną (*Set As Default*).

Nas jednak interesuje umieszczony na dole przycisk *Add printer*. Po jego kliknięciu CUPS może poprosić nas o podanie nazwy użytkownika i hasła. Nazwa dotyczy użytkownika na komputerze na którym działa CUPS, może to być *root* lub dowolny inny użytkownik mający prawa do administrowania drukarkami. W dystrybucji Debian prawa te nadawane są przez dodanie użytkownika do grupy *lpadmin* (można to zrobić dopisując użytkownika do grupy w pliku `/etc/group`).

Jeżeli podamy prawidłową nazwę użytkownika i hasło, zostaniemy poproszeni o podanie nazwy drukarki (pole *Name*) oraz jej lokalizacji i opisu (*Location* i *Description*). Dwa ostatnie pola są tylko opisem dla użytkownika, w szczególności mogą zostać puste. Istotna jest natomiast nazwa. Tradycyjnie nazwa powinna składać się z małych liter i ewentualnie cyfr, np. „hp690” czy „minolta2300”. Najczęściej używanej drukarce warto nadać nazwę „lp” - niektóre stare programy mogą nie honorować ustawienia domyślnej drukarki i zawsze domyślnie drukować na drukarkę o właśnie takiej nazwie.

Po podaniu nazwy klikamy przycisk *Continue* i przechodzimy do wyboru typu drukarki, a właściwie sposobu dostępu do niej. Do wyboru mamy:

- *AppSocket/HP JetDirect* - wybieramy tą opcję w przypadku większości drukarek tzw. „sieciowych” (wyposażonych w kartę sieciową - nie mylić z drukarkami udostępnionymi). Wtedy na następnym ekranie proszeni jesteśmy o podanie adresu, który zwykle będzie w postaci `socket://XXX.XXX.XXX.XXX:9100`, gdzie zamiast XXX... należy podać adres IP drukarki. Jeżeli nie znamy adresu IP drukarki, możemy spróbować przeskanować sieć lokalną, korzystając np. z

programu **nmap**. Następująca komenda powinna pokazać nam wszystkie dostępne w sieci drukarki nasłuchujące na porcie 9100:

```
nmap -p 9100 192.168.1.*
```

Oczywiście zamiast „192.168.1.\*” należy podać maskę dla swojej sieci lokalnej. Program powinien nas poinformować, pod którym adresem znajduje się otwarty port 9100, np.:

```
Interesting ports on 192.168.1.49:
PORT      STATE SERVICE
9100/tcp  open  jetdirect
```

- *Internet Printing Protocol (http/ipp)* - wybieramy, jeżeli wiemy, że nasza drukarka obsługuje taki protokół lub jeżeli chcemy mieć dostęp do drukarki zainstalowanej na innym serwerze z CUPS.
- *LPD/LPR Host or Printer* - wybieramy, jeżeli chcemy drukować na drukarce skonfigurowanej na innym serwerze Linux/Unix z uruchomionym serwerem druku LPD (poprzednik CUPS'a).
- Port lokalny - USB, równoległy lub szeregowy. Jeżeli CUPS wykryje podłączone do portu urządzenie, jego nazwa pojawi się obok portu, ułatwiając nam wybór.
- *Windows Printer via SAMBA* - wybieramy jeżeli chcemy skorzystać z drukarki podłączonej do komputera z Windows obecnego w naszej sieci lokalnej.

Jeżeli ta opcja nam się nie pojawi, może to oznaczać, że nie mamy poprawnie zainstalowanego klienta Samby (pakiet `smbclient` w Debianie).

Na następnym ekranie musimy podać adres udziału Windows, pod którym widoczna jest dana drukarka. Adres ten najłatwiej sprawdzić na komputerze z Windows, do którego podłączona jest dana drukarka. Adres ma postać `smb://NAZWA_KOMPUTERA/NAZWA_DRUKARKI`. Nazwę komputera sprawdzamy klikając w Windows prawym przyciskiem myszy na ikonie *Mój komputer* i wybierając *Właściwości* a następnie zakładkę *Nazwa*. Nazwę drukarki można sprawdzić wybierając *Udostępnianie* we właściwościach drukarki. Inną metodą może być przeszukanie całej sieci Microsoft Network za pomocą polecenia `smbtree` (hasło podajemy puste) - powinny się wyświetlić wszystkie widoczne w sieci udziały, w tym drukarki.

Jeśli komunikacja z drukarką nie działa i dostajemy komunikat `NT_STATUS_ACCESS_DENIED`, to trzeba będzie w nazwie zasobu podać użytkownika i hasło. Sytuacja taka występowała przy starszych wersjach Samby i systemie Windows XP, obecnie raczej nie powinna mieć miejsca. Gdyby jednak się to zdarzyło, to najlepiej utworzyć w Windows specjalnego użytkownika tylko do drukowania korzystając z Panelu Sterowania, opcja *Zarządzanie Kontami Użytkowników* i upewnić się, że ma możliwość drukowania na udostępnionej drukarce. Można także ustawić hasło. W takiej sytuacji adres zasobu wpisywany przy konfiguracji drukarki powinien mieć postać `smb://user:haslo@NAZWA_KOMPUTERA/NAZWA_DRUKARKI`, gdzie *user* oznacza nazwę użytkownika, a *haslo* jest ustawionym hasłem.

Następnym krokiem, po wybraniu sposobu dostępu do drukarki, jest podanie jej typu/modelu. Na pierwszym ekranie wybieramy producenta drukarki, np. HP, Epson czy Minolta. Na kolejnym wybieramy konkretny model wraz z typem sterownika (czasami dla drukarki dostępny jest więcej niż jeden sterownik). Zwykle jeden ze sterowników będzie oznaczony jako *recommended* - i jest to zazwyczaj najlepszy wybór.

**Notatka:** Jeśli dostajemy komunikat, że KDE nie może przebudować bazy sterowników, prawdopodobnie oznacza to brak zainstalowanego pakietu `foomatic-db-engine` - należy go doinstalować.

Sterowniki różnią się ilością dostępnych opcji i jakością druku. Jeżeli nie będziemy zadowoleni z wybranego sterownika, zawsze możemy zmienić go uruchamiając ponownie konfigurację drukarki (przycisk *Modify Printer* na ekranie CUPS'a z listą drukarek).

**Notatka:** Jeżeli mamy drukarkę obsługującą PostScript, to zwykle najlepszym wyjściem będzie wybranie typu drukarki *Generic* i sterownika *Generic PostScript Printer*. Wyjątkiem mogą być drukarki atramentowe, gdzie czasem warto mieć dostęp do specyficznych dla drukarki ustawień dotyczących np. jakości druku. Podobne ogólne sterowniki istnieją dla drukarek korzystających z różnych wersji języka PCL firmy HP.

Jeżeli mamy problem z wyborem lub ze znalezieniem sterownika do naszej drukarki, bądź też zastanawiamy się jaki model wybrać przy zakupie, warto zajrzeć na stronę [www.linuxprinting.org](http://www.linuxprinting.org) (<http://www.linuxprinting.org>). Link *Database / Printer Listings* daje nam dostęp do opisów praktycznie wszystkich spotykanych na rynku drukarek, wraz z informacją o współpracy z Linuksem i o sugerowanym sterowniku.

Po zaakceptowaniu rodzaju sterownika CUPS poinformuje nas o utworzeniu nowej drukarki, możemy kliknąć na link z jej nazwą aby wyświetlić stronę opisującą stan drukarki i dającą dostęp do opcji drukarki. Na początek warto wydrukować stronę testową (*Print Test Page*). Jeżeli się to nie powiedzie, przyczyny błędu możemy szukać w linii z opisem stanu drukarki, choć zwykle trzeba będzie zajrzeć do logów (w katalogu `/var/log/cups`). Jeżeli problem dotyczy drukarki udostępnianej przez Sambę, warto też sprawdzić, czy możemy się „zalogować” na drukarkę za pomocą polecenia **smbclient** (jako argument podając nazwę udziału). Jeśli jesteśmy niezadowolony z jakości czy prędkości druku warto sprawdzić opcje udostępnianie po kliknięciu przycisku *Configure Printer*.

### 3.2.22.3. Konfiguracja drukarki z wykorzystaniem Menedżera Drukowania KDE

Jeżeli mamy dostęp do komputera ze środowiskiem KDE, to trochę mniej pracy będziemy musieli wykonać konfigurując drukarkę za pomocą narzędzia dostarczanego przez KDE - Menedżera Drukowania. Można go uruchomić z menu KDE (menu *Ustawienia*) - wtedy klikamy na ikonie *Dodaj* i wybieramy *Dodaj drukarkę/klasę* lub też klikając na ikonę z „różdżką” w oknie wydruku dowolnej aplikacji KDE (okno to można też uruchomić komendą **kdeprint**).

Powinniśmy zobaczyć śliczny kreator. Na ekranie startowym naciskamy **Następny** i wybieramy typ drukarki, podobnie jak przy standardowej konfiguracji za pomocą przeglądarki (zobacz Sekcja 3.2.22.2).

- W przypadku udostępnionej drukarki Windows na następnym ekranie możemy podać użytkownika Windows na którego będziemy się logować (konieczne dla Windows XP), a potem podać adres udziału. Ułatwieniem jest guzik **Przeszukaj**, pozwalający nam wybrać drukarkę klikając na jedną z drukarek udostępnionych w sieci.
- Możliwość przeszukania sieci mamy też wybierając drukarkę sieciową (TCP - większość drukarek sieciowych lub IPP). Na ekranie *Informacja o drukarce sieciowej* sprawdzamy, czy zaproponowana przez menedżera podsieć jest zgodna z adresem naszej sieci lokalnej - w szczególności nie powinna to być czasem proponowana sieć 127.0.0.1. W razie potrzeby zmieniamy sieć przyciskiem **Ustawienia** (możemy też wybrać inny numer portu niż zwykle używany przez drukarki 9100, a także ograniczyć przeszukiwane adresy IP). Następnie klikamy na **Szukaj** (jeżeli pojawi się ostrzeżenie o niezgodności adresu sieci, możemy je bezpiecznie zignorować). W głównym okienku kreatora powinna pojawić się lista znalezionych drukarek sieciowych.

- Jeżeli chcemy skorzystać z drukarki skonfigurowanej na innym komputerze z CUPS'em, wybieramy *Zdalny serwer CUPS*. Na kolejnym ekranie możemy podać użytkownika na którego chcemy się logować i hasło, a na jeszcze następnym wpisać adres komputera z serwerem CUPS. Powinniśmy zobaczyć listę dostępnych skonfigurowanych drukarek.
- Jeżeli mamy w sieci skonfigurowany serwer LPD, wybieramy *Zdalny serwer LPD* i podajemy jego adres oraz nazwę kolejki (drukarki), np. lp.
- Jeżeli konfigurujemy drukarkę lokalną (podłączoną przez port równoległy, szeregowy lub USB), podłączone drukarki powinny być od razu wyświetlone, nie będziemy musieli także zwykle samodzielnie wybierać producenta i sterownika.

Kolejnym krokiem jest wybór producenta i modelu/sterownika drukarki, możemy też zaznaczyć że chcemy korzystać z drukarki PostScriptowej. Ciekawą opcję udostępnia guzik **Inne**, pozwalający załadować plik PPD z opisem drukarki, ściągnięty np. z [www.linuxprinting.org](http://www.linuxprinting.org) (<http://www.linuxprinting.org>). Może to być przydatne jeżeli nasza drukarka nie jest obecna jeszcze w zainstalowanej w systemie wersji bazy sterowników drukarek.

Po wyborze typu drukarki możemy zmienić ustawienia drukarki takie, jak tryb druku, rozdzielczość, rodzaj papieru itp. klikając przycisk **Ustawienia** lub wydrukować stronę testową klikając **Test**.

Zostaje jeszcze kilka opcjonalnych ustawień takich, jak drukowanie winiet, ograniczenia na ilość stron dla użytkowników, zdefiniowanie użytkowników, którzy mają mieć dostęp do drukarki.

W ostatnim kroku podajemy nazwę dla drukarki, możemy podać także opis lokalizacji. Zatwierdzamy parametry nowej drukarki (możemy być poproszeni o podanie nazwy i hasła użytkownika mającego uprawnienia do drukowania) i gotowe.

#### 3.2.22.4. Konfiguracja serwera drukowania opartego o SMB

Jeżeli chcemy, aby skonfigurowane w CUPS'ie drukarki widoczne były w sieci dla komputerów z Windows, musimy upewnić się że:

- Na komputerze linuksowym uruchamiane są demony samby. Pod Debianem odpowiednia komenda dodająca sambę do plików startowych systemu to:

```
update-rc.d samba defaults
```

- Upewnić się, że w pliku konfiguracyjnym Samby `/etc/samba/smb.conf` mamy odkomentowane następujące wpisy:

```
printing = cups
printcap name = cups
```

W tym samym pliku należy też ustawić nazwę grupy dla sieci Windows, np.

```
[global]
```

```
## Browsing/Identification ###
```

```
# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = MYGROUP
```

i upewnić się, że mamy udział z drukarkami, np.:

```
[printers]
comment = All Printers
```

```

browseable = no
path = /tmp
printable = yes
public = no
writable = no
create mode = 0700

```

**Notatka:** Uwaga! Powyższy wpis udostępnia wszystkie skonfigurowane na danym hoście drukarki (również drukarki sieciowe, nie podłączone bezpośrednio do tego hosta). Jeżeli nie chcemy udostępniać wszystkiego, to możemy utworzyć sekcję o nazwie takiej, jak identyfikator danej drukarki (można je sprawdzić za pomocą polecenia **lpstat -a**). Na przykład jeśli chcemy udostępnić jedynie drukarkę **lp** to tworzymy sekcję:

```

[lp]
comment = My favourite printer
path = /tmp
printable = Yes

```

- Jeżeli Samba nie była uruchomiona, należy jeszcze ją uruchomić (lub uruchomić ponownie jeżeli była uruchomiona, a dokonywaliśmy zmian w plikach konfiguracyjnych Samby):

```
/etc/init.d/samba start
```

To wszystko - nasze drukarki powinny być widoczne dla wszystkich komputerów w sieci jako udostępnione drukarki Windows.

### 3.2.23. Katalog współdzielony z wykorzystaniem Samby

Czasami zachodzi potrzeba wymiany danych między użytkownikami logującymi się na komputery linuksowe i windowsowe. Jednym z rozwiązań jest udostępnienie publicznego katalogu, który będzie widoczny ze wszystkich komputerów w sieci i będzie pozwalał wszystkim użytkownikom na zapisywanie w nim danych.

Niniejszy opis dotyczy przykładowej konfiguracji, w której sieć lokalna ma adres IP 192.168.1.0, istnieje jeden serwer linuksowy o nazwie *linserv*, kilka terminali linuksowych i windowsowych. Wszystkie komputery linuksowe muszą mieć zainstalowaną Sambę. Kolejno:

1. Na serwerze *linserv* tworzymy nową grupę *wspolne* i dodajemy do niej użytkowników, którzy będą mieli mieć dostęp do katalogu. Wybierzmy numer grupy na tyle duży, żeby na żadnym z terminali linuksowych nie istniała już grupa o takim numerze. Może to oznaczać taki wpis w pliku

```
/etc/group:
```

```
wspolne:x:1050:kierownik,prezes,nobody,palacz
```

Do grupy muszą należeć wszyscy użytkownicy, którzy mają mieć dostęp do katalogu współdzielonego, oraz, dodatkowo, użytkownik z uprawnieniami którego działają demony Samby (czyli zwykle *nobody*).

2. Analogicznie, grupę o takiej samej nazwie i numerze tworzymy na wszystkich terminalach linuksowych.
3. Na serwerze tworzymy odpowiedni katalog:

```

mkdir /home/wspolne
chown nobody.wspolne /home/wspolne

```

```
chmod 2755 /home/wspolne
```

4. Na wszystkich terminalach linuksowych także tworzymy katalog:

```
mkdir /home/wspolne
```

5. Na wszystkich komputerach linuksowych (z serwerem włącznie) w pliku `/etc/samba/smb.conf` ustawiamy parameter `workgroup` na używaną w sieci nazwę grupy roboczej, np.

```
workgroup = WORKGROUP
```

Jeżeli jeden z komputerów ma pełnić rolę serwera WINS, to musimy na nim dodać wpis:

```
wins support = yes
```

Jeżeli pozostałe komputery mają problemy ze znalezieniem serwera WINS, to można dodać im wpis z numerem IP serwera, np.

```
wins server = 192.168.1.100
```

6. Na serwerze linuksowym modyfikujemy plik `/etc/samba/smb.conf`, aby obecne były następujące wpisy:

```
[global]
; zmiana ustawien domyslonych
security = share
load printers = no ; bonus

; dodana nowa sekcja
[wspolne]
comment = Katalog wspoldzielony
path = /home/wspolne
browseable = yes
writeable = yes
public = yes
force user = nobody
force group = wspolne
create mode = 0664
directory mode = 0775
force create mode = 0664
force directory mode = 0775
hosts deny = ALL
hosts allow = 192.168.1.21 192.168.1.22 192.168.1.2 192.168.1.3
```

Ostatnia linia to przykładowa lista komputerów, którym pozwalamy na dostęp do naszego katalogu. Można też podać całą podsieć:

```
hosts deny = ALL
hosts allow = 192.168.1.
```

7. Jeżeli używamy jądra 2.4, to na wszystkich komputerach linuksowych dodajemy do pliku `/etc/fstab` linię:

```
//linserv/wspolne /home/wspolne smbfs rw,gid=1050,mask=664,dmask=775, \
password= 0 0
```

Zamiast 1050 wstawiamy oczywiście numer grupy 'wspolne'. Nazwa komputera linserv musi też być poprawnie tłumaczona na IP (można dodać wpis do `/etc/hosts`).

Niestety, jeżeli używamy jądra 2.6, to po wykonaniu tego oraz opisanych niżej kroków nasz katalog prawdopodobnie będzie niedostępny dla użytkowników zalogowanych bezpośrednio na serwerze (programy chcące uzyskać dostęp do katalogu będą się "przywieszały"), a w logach otrzymamy komunikaty typu:

```
Dec 11 00:33:35 linserv kernel: smb_add_request: request [c50f2e80, mid=17] \
```

timed out!

Radą na to jest upgrade samby (w Debianie unstable pakiet co najmniej z wersją 3.0.7-2) i użycie nowej wersji protokołu SMB o nazwie CIFS (Common Internet File System). Oznacza to następujące wpisy w pliku `/etc/fstab` na serwerze i terminalach linuxowych:

```
//linserv/wspolne /home/wspolne cifs rw,gid=1050,file_mode=0664,dir_mode=0775, \
password= 0 0
```

8. Jeżeli dotychczas tego nie zrobiliśmy, to należy zrestartować Sambę, aby zostały uwzględnione zmiany poczynione przez nas w plikach konfiguracyjnych:

```
/etc/init/samba restart
```

9. Na komputerach linuxowych montujemy nasz nowy katalog za pomocą komendy:

```
mount /home/wspolne
```

Jest to czynność jednorazowa - katalog będzie zawsze domyślnie montowany podczas startu systemu.

10. W systemie Debian konieczna jest też zmiana w pliku `/etc/profile` wywołania 'umask 022' na 'umask 002'. Prawidłowe prawa dostępu do wspólnego katalogu będą mieli tylko użytkownicy zalogowani po tej zmianie.
11. Na koniec możemy na komputerach linuxowych utworzyć w katalogach domowych użytkowników linki symboliczne do katalogu `/home/wspolne`, a na komputerach windowsowych zmapować zasób `//linserv/wspolne` jako nowy dysk.

### 3.2.24. Synchronizowanie zawartości katalogów - rsync

**Rsync** - synchronizuje zbiory dyskowe (np. na odległych maszynach). Program jest częścią standardowej dystrybucji Linuksa. Został użyty do backupowania baz danych SZARPa oraz do przegrywania na bieżąco zawartości bazy danych z serwera (z działającym **parcookiem**) na klienta (z **SzarpDrawem**).

**rsync** działa w dwóch konfiguracjach:

- klient rsync <--> serwer rsync
- klient rsync (korzysta z rsh) <--> serwer rshd

Poniżej opisano **rsynca** w 1-szej konfiguracji dla zestawu dwóch komputerów, z których:

- na pierwszym siedzi parcook i meaner. Następuje ciągły zapis do bazy (serwer);
- na drugim mamy Linuksa w dużym dyskiem do robienia backupów (klient).

Zakładam, że oba kompy są połączone siecią. Pierwszy w komputerów (serwer) musi mieć działającego **rsync-demoną**, czyli:

- zainstalowany pakiet rsync (z rpm-ki),
- ustawiony w inetd:
- sprawdzić czy w `/etc/services` istnieje linijka

```
rsync 873/tcp
```

Jeśli nie, to ją dopisać!

- sprawdzić czy w `/etc/inetd.conf` jest coś na kształt:

```
rsync stream tcp nowait root /usr/bin/rsync rsyncd --daemon
Jeśli nie, to dopisać!
```

- PrzeHUPować demona internetowego:

```
killall -HUP inetd
```

- skonfigurowanego rsync-demonia: tworzymy plik `/etc/rsyncd.conf` następującego formatu:

```
# początek pliku /etc/rsyncd.conf
```

```
uid = root
[baza]
path = /opt/szarp/baza
[home]
path = /home
# .. i tak dalej
```

```
# EOF
```

oczywiście możemy mieć również jedną sekcję albo bardzo dużo. Do ściągania bazy nie trzeba wcale żadnych dodatkowych opcji. Jako uid możemy wpisać dowolnego użytkownika zamiast roota, ważne, żeby miał prawo do wchodzenia do katalogów i ich odczytu.

Teraz drugi komp:

- instalujemy pakiet rsync (jeśli go jeszcze nie ma),
- żeby ściągnąć bazę, wydajemy polecenie:

```
rsync -avz <adres_serwera>::baza <katalog_docelowy>
gdzie <adres_serwera> to może być IP lub nazwa, <katalog_docelowy> to istniejący lokalnie katalog z prawami do zapisu, np.
```

```
rsync -avz 192.168.0.1::baza /sciagnieta_baza
```

- żeby ściągnąć katalogi użytkowników wystarczy wklepać:

```
rsync -avz 192.168.0.1::home /sciagniete_katalogi
```

**UWAGA!**

- **WAŻNE!** Powtórne wywołanie tej samej komendy spowoduje ściągnięcie tylko nowych i zmodyfikowanych plików a nie wszystkiego od nowa.
- Każdy zasób, który ściągamy tym sposobem musi być obecny w jakiejś sekcji w pliku `/etc/rsyncd.conf`
- Bardzo ważny jest ten podwójny dwukropek „::” w komendzie. Oznacza on, że korzystamy z **rsyncd**, a nie z **rsh**.
- **rsync** może pracować nie tylko w trybie „pop”, ale również w trybie „push”. Ponieważ jest to mniej bezpieczne, wymagane jest ustawienie parametru w pliku `/etc/rsyncd.conf` po stronie serwera:

```
read only = false
```

Wywołanie komendy w tym przypadku różni się tylko kolejnością parametrów, np.:

```
rsync -avz /katalogi 192.168.0.1::home
```

- w poleceniu `rsync -avz . .` można pominąć parametr `-v` (bo to jest `verbose`) oraz `-z` (kompresja 'w locie'). Parametr `-a` oznacza archiwizację z zachowaniem wszystkich atrybutów plików (`uid/gid/rwx/typ pliku`).

W Legionowie stworzono konfigurację **rsync + crontab** do robienia backupów bazy. Dwa serwery „wymieniają” się własnymi bazami (**rsync** pracuje w trybie „pop”) i dodatkowo przesyłają bazę do terminala („push”). Plik `crontab` z jednego z serwerów przedstawiono poniżej, w rozdziale Sekcja 3.2.25.

### 3.2.25. Ustalanie zadań do wykonania w systemie - crontab

W Linuksie mamy możliwość ustalania jakie zadania mają być wykonywane w określonym terminie. Do dyspozycji mamy dwa standardowe klasyczne unixowe programy **at** i **cron**. Jaka jest między nimi różnica? Otóż **at** służy do jednorazowego wykonywania poleceń, np. *ósmego marca b.r. o godz. 9:00 wyślij wszystkim paniom w firmie odpowiedniego maila*. **Cron** służy do wykonywania zadań periodycznie, np. *co godzinę rób mirrora pewnej strony WWW i umieszczaj ją na naszym serwerze*.

Oba z tych programów pracują jako klient-serwer, tzn. zarządzaniem procesów steruje stale obecny w systemie demon (**atd** i **crond**), a użytkownik korzysta z programu klienckiego (odpowiednio **at** i **crontab**). Dla obu tych programów tworzy się przeważnie plik ze zleceniami. Dla **at** jest to skrypt shella. Natomiast format pliku `crona` jest dosyć skomplikowany i jest opisany poniżej:

Jedno zadanie jest opisane w jednej linii, zawierającej pięć pól wyjaśniających kiedy dane zadanie ma się wykonywać, po czym umieszcza się komendę shella, czyli:

```
<minuty> <godziny> <dni miesiąca> <miesiąc> <dni_tygodnia> <KOMENDA>
```

Można zastąpić dane pole gwiazdką, żeby zadanie wykonywało się „zawsze”, np.:

```
0 1 * * * mail bgates@microsoft.com < obelgi
```

Oznacza, że codziennie o 1-szej nad ranem będziemy wysyłać na adres `bgates@microsoft.com` zawartość pliku `obelgi`.

Żeby zgłosić cronowi nasze zlecenie, należy uruchomić komendę:

```
crontab <nazwa pliku konfiguracyjnego>
```

Ponieważ tradycyjnie konfigurację umieszcza w pliku `crontab`, więc najczęstsze wywołanie `crontaba` ma postać:

```
crontab crontab
```

Każdy z użytkowników może mieć po jednym aktywnym pliku konfiguracyjnym `crontaba` (tzn. jeśli wywołamy powtórnie komendę `crontab`, to stare ustawienia zostaną zatarte). Jeżeli programy wywoływane jako komendy `crontaba` wyświetlają cokolwiek na standardowym wyjściu, dane te są przychwytywane przez `crona` i wysyłane mailem do właściciela `crontaba`. Żeby uniknąć wysyłania maili należy ustawić zmienną środowiskową `MAILTO=""`. Można również wysyłać maile komuś innemu, ustawiając tę zmienną na określony adres, np. `MAILTO="lucek@praterm.com.pl"`. Zmienną `MAILTO` można (i należy) ustawiać w pliku konfiguracyjnym `crontaba`. Jeżeli chcemy wykonywać periodycznie wiele niezależnych zleceń to musimy w jednym pliku konfiguracyjnym dla `crontaba` wpisać wiele rekordów (linii), tak jak na przykładzie poniżej (dwa zlecenia):

```
MAILTO=""
```

```
1,11,21,31,41,51 * * * * /usr/bin/rsync -av /opt/szarp/baza/ 192.9.200.12::leg1
3,13,23,33,43,53 * * * * /usr/bin/rsync -av 192.9.200.170::baza \
/opt/szarp/leg2/baza/
```

### 3.2.26. Wirtualne konsole w Linuksie

Możemy ich mieć wiele - najprawdopodobniej do 255. Konsole te oznaczane są symbolem `ttyX`, gdzie `X` jest identyfikatorem konsoli (terminala) - liczbą naturalną (np. `tty1`, `tty2`, `tty3` itd.). Reprezentowane są przez urządzenia znakowe w katalogu `/dev` (np. `/dev/tty1`) o numerze `MAJOR=4` i `MINOR=0...255`. Bezpośredni dostęp przez kombinację klawiszy `ALT-<klawisz funkcyjny>` możemy mieć do początkowych 24-ch terminali (`tty1` - `tty24`). Terminale `tty1-tty12` osiągamy poprzez wciśnięcie lewego klawisza `ALT` i klawisza funkcyjnego `Fn`, co powinno nas przełączyć na terminal `ttyn` (np. po naciśnięciu `ALT-F6`, przełączamy się na terminal `tty6`). Kolejnych 12 terminali jest osiągalnych przez prawy `ALT-Fn`, co przełącza nas na terminal `tty{n+12}` (np. prawy `ALT-F2` przełącza nas na terminal `tty14`). Innym sposobem przełączania terminali jest użycie kombinacji `ALT-<strzałka>`. Po naciśnięciu prawej strzałki przełączamy się na kolejną konsolę, natomiast po naciśnięciu lewej przechodzimy na poprzedni terminal. Wszystkie te operacje są przeprowadzane modulo ilość terminali.

Przykład: założmy, że pracujemy w standardowym 3-cim runlevelu bez iksów. Niech aktualnym terminalem będzie `tty3`. Wtedy pięciokrotne naciśnięcie `ALT-<prawa strzałka>` przełączy nas kolejno na terminale `tty4`, `tty5`, `tty6`, `tty1`, `tty2`. Wówczas pięciokrotne naciśnięcie lewej strzałki przeniesie nas kolejno na terminal `tty1`, `tty6`, `tty5`, `tty4`, `tty3` (z powrotem do punktu wyjścia). Ten sam rezultat moglibyśmy osiągnąć naciskając jeszcze jeden raz `ALT-<prawa strzałka>`

Powyższe skróty (`ALT-<klawisz funkcyjny>` `ALT-<strzałka>`) nie działają niestety pod iksami, w iksach mają one inne znaczenie. Do przełączenia konsoli, gdy aktywne są iksy, używamy kombinacji lewy `ALT-CTRL-<klawisz funkcyjny>` o działaniu odpowiadającym kombinacji lewy `ALT-<klawisz funkcyjny>` w konsolach tekstowych. Uwaga: kombinacja `ALT-CTRL-<strzałka>` nie może być używana w ten sposób. Dostęp do konsoli o dużych numerach (np. `tty25`, `tty26`) jest tylko przez `ALT-<strzałka>`. Problem z przełączaniem konsoli pojawia się, gdy mamy równocześnie wiele sesji iksów. Wprawdzie taka sytuacja nie zdarza się w praktyce nigdy (iks + KDE to grubo ponad 20 MB ramu), ale niespójność systemu dostępu do odpowiednich terminali jest trochę irytująca.

Każdy terminal, zanim może być użyty, musi być otwarty. Służą to tego programy z rodziny `getty`. Sam `getty` (`/sbin/getty`) jest dosyć mocnym programem i może być używany do otwierania konsol wirtualnych jak i innych terminali (np. połączonych portem szeregowym). Dlatego używa się przeważnie jego uproszczonych, wyspecjalizowanych klonów: **mingetty** do otwierania konsol wirtualnych, **mgetty** do modemów i innych. **mingetty** jest uruchamiany przez `init`. W standardowym RedHacie `init` uruchamia `mingetty`'a sześciokrotnie. Fragment pliku `/etc/inittab`:

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Jeżeli mamy potrzebę użycia większej ilości terminali (konsol), to należy dopisać odpowiednie wywołania `mingetty`'a, np.:

```
11:2345:respawn:/sbin/mingetty tty11
```

```
12:2345:respawn:/sbin/mingetty tty12
```

*Uwaga!* Niektóre dystrybucje nie mają urządzeń specjalnych w katalogu `/dev/` reprezentujących wyższe numery terminali (`tty13-tty24`), nie mówiąc już o terminalach o jeszcze wyższych identyfikatorach. Tak jest w RedHacie. Należy wówczas (tylko gdy jest taka potrzeba, czyli nieczęsto) dodać te urządzenia ręcznie, np. komendą **mknod**:

```
mknod /dev/tty12 c 4 12
mknod /dev/tty13 c 4 13
mknod /dev/tty14 c 4 14
```

Dopiero wtedy wrzucamy konfigurację do `inittaba` i możemy się cieszyć prawym ALT-Fn.

Terminale otwierane są również przez ikсы (X-serwer jest związany z jakimś określonym terminalem). Ikсы wchodzi nam na pierwszy wolny terminal (z wyjątkiem „zarezerwowanego” `tty1`): w przypadku standardowych dystrybucji 6-cio terminalowych jest to terminal siódmy (`tty7`). Tak jest w przypadku uruchamiania Linuksa w runlevelu 5-tym, jak i w przypadku ręcznego uruchamiania ikсов komendą **startkde**, **startx** itp. Teoretycznie, gdyby żaden z terminali nie był inicjowany przez `inita`, ikсы pakowałyby się na terminal drugi (`tty2`). W przypadku, gdy wszystkie dostępne terminale są już zajęte (otwarte przez konsolowego `mingetty`'a lub otwarte przez inne sesje ikсов) X-server wyjdzie z błędem „Could not open `/dev/ttyXX`”. Wtedy należy użyć komendy **mknod** do stworzenia urządzenia wirtualnego `/dev/ttyXX`. Ikсы dobrze się czują również na wysokich numerach terminali (np. `tty30`), dostęp jednak do nich jest trochę utrudniony (wielokrotne ALT-<strzałka>).

### 3.2.26.1. „Naprawianie” terminali wirtualnych

Czasami terminal zostaje „zepsuty” (np. przez próbę wyświetlenia pliku binarnego za pomocą komendy **cat**). „Zepsucie” objawia się zwykle poprzez niezrozumiałe znaki na terminalu (lub brak wyświetlania czegokolwiek). Spowodowane jest zmianą pewnych parametrów terminala. Naprawienie polega na wpisaniu komendy:

```
stty sane
```

lub

```
reset
```

(działanie obu komend jest w tym przypadku identyczne). W trakcie wpisywania komendy nie należy zwracać uwagi na to, że znaki komendy nie są wyświetlane lub wyglądają niezrozumiale (i tak powinna przynieść skutek). Jeżeli jednak nie pomoże, może to oznaczać, że terminal został zablokowany. Zablokowanie to następuje zazwyczaj po naciśnięciu CTRL-S i łatwo je zrobić przypadkowo. Żeby odblokować terminal, należy zwykle nacisnąć CTRL-Q.

## 3.2.27. Uruchamianie wielu X-serwerów

W ramach jednego hosta może funkcjonować wiele X-serwerów. Oznaczone są przez tzw. display number, czyli liczbę poprzedzoną dwukropkiem, np. `:0`, `:1`, `:5`, `:666`, itp. W każdym przypadku domyślny display number to `:0`. Mechanizm obsługi ekranu jest analogiczny do terminali wirtualnych: równocześnie możemy uruchomić kilka serwerów, do których dostęp mamy poprzez ALT-Fn lub

CRTL-ALT-Fn (w X-ach ALT-Fx miewa trochę inne znaczenie niż na konsoli tekstowej, dlatego używamy CTRL-ALT-Fx). Np. w runlevelu 3-cim możemy uruchomić 3 X-serwery komendą:

```
X&
X :1&
X :2&
```

Do tak uruchomionych serwerów możemy się odwoływać najprawdopodobniej poprzez odpowiednio CTRL-ALT-F7, CTRL-ALT-F8, CTRL-ALT-F9. Trudno powiedzieć ile kopii X-ów może być równocześnie uruchomionych na jednej maszynie, prawdopodobnie tyle na ile wystarczy pamięci. Jeżeli używamy XDM/XDMCP to możemy użyć opcji `-query`, np.

```
X :1 -query 192.168.0.2
X :1 -query localhost
```

Opis parametrów przekazywanych do programu **X** można uzyskać przez `man Xserver`, ogólny opis X-ów przez `man X`.

### 3.2.28. X-serwer w oknie (Xephyr/Xnest)

Istnieje możliwość uruchomienia X-serwera w oknie wyświetlanym przez inny, działający X-serwer. Służy do tego **Xephyr** lub **Xnest** (poprzednia wersja Xephyra). Instalujemy go wywołując komendę:

```
apt-get install xserver-xephyr
```

Xephyr może posłużyć do uruchomienia dodatkowego lokalnego Xserwera lub do połączenia się z serwerem zdalnym przy użyciu opcji `-query`. Można go wykorzystać do testowania XDMCP.

Przykładowe użycie:

```
Xephyr :1 -query host
```

Inne użyteczne opcje:

- `-screen` - pozwala podać wymiary okna Xephyra. Jako argument należy podać wymiary w postaci SZEROKOSCxWYSOKOSC w pikselach.
- `-fullscreen` - pozwala uruchomić Xephyr na pełnym ekranie

### 3.2.29. Instalacja i uruchamianie emulatora Qemu

#### 3.2.29.1. Opis

*Qemu* to dostępny na licencji GPL/LPGL emulator procesora. Może pracować w dwóch trybach - użytkownika, w którym pozwala na uruchamianie programów linuxowych skompilowanych na jeden typ procesora na innym typie, oraz w trybie emulacji całego systemu. W tym drugim trybie tworzy wirtualną maszynę z procesorem i odpowiednimi peryferiami, pozwalając na uruchomienie pod jej kontrolą całego systemu operacyjnego - podobnie jak robi to oprogramowanie typu VMWare czy Xen. Pozwala to na przykład na przetestowanie płyty instalacyjnej czy dystrybucji *live* Linuksa bez konieczności wypalania płyty i resetowania komputera, a także np. na uruchomienie pod kontrolą

Linuksa sesji systemu Windows. Poza emulacją komputera typu PC Qemu pozwala także na emulację innych architektur, ale niniejszy opis skupia się na używaniu Qemu pod Linuxem na platformie PC do uruchamiania innej sesji Linuksa lub Windows.

Do Qemu przy emulacji PC->PC dostępny jest moduł do jądra *kqemu*, przyspieszający jego działanie 5 - 10 razy - dzięki temu programy uruchomione pod kontrolą systemu operacyjnego w wirtualnej maszynie działają niewiele wolniej, niż uruchomione bezpośrednio na danym komputerze. Kqemu nie jest oprogramowaniem open-source - jest bezpłatny do użytku, ale nie można go dystrybuować bez zezwolenia autora. Powstał więc także otwarty klon *kqemu* o nazwie *qvm86*.

Przydatne linki:

- [fabrice.bellard.free.fr/qemu/](http://fabrice.bellard.free.fr/qemu/) (<http://fabrice.bellard.free.fr/qemu/>) - główna strona Qemu i Kqemu.
- [kidsquid.com/cgi-bin/moin.cgi/FrequentlyAskedQuestions](http://kidsquid.com/cgi-bin/moin.cgi/FrequentlyAskedQuestions) (<http://kidsquid.com/cgi-bin/moin.cgi/FrequentlyAskedQuestions>) - obszerny FAQ do Qemu.
- [pl.wikipedia.org/wiki/QEMU](http://pl.wikipedia.org/wiki/QEMU) (<http://pl.wikipedia.org/wiki/QEMU>) - opis Qemu w polskiej Wikipedii.
- [en.wikipedia.org/wiki/QEMU](http://en.wikipedia.org/wiki/QEMU) (<http://en.wikipedia.org/wiki/QEMU>) - opis Qemu w angielskiej Wikipedii.
- [savannah.nongnu.org/projects/qvm86/](http://savannah.nongnu.org/projects/qvm86/) (<http://savannah.nongnu.org/projects/qvm86/>) - strona projektu *qvm86*.
- [emeitner.f2o.org/qemu\\_launcher](http://emeitner.f2o.org/qemu_launcher) ([http://emeitner.f2o.org/qemu\\_launcher](http://emeitner.f2o.org/qemu_launcher)) - interfejs graficzny dla Qemu.

### 3.2.29.2. Instalacja z paczki debianowej

Qemu jest dostępne w postaci paczki debianowej - komenda **apt-get install qemu** zainstaluje nam emulator wraz z BIOS'em do karty graficznej. Niestety, *qemu* w paczce debianowej nie wspiera akceleratorów *kqemu* ani *qvm86*. Jeśli więc chcemy mieć emulator działający na tyle szybko, by wygodnie uruchamiać np. sesję Windows, to musimy zainstalować go ze źródeł.

### 3.2.29.3. Instalacja ze źródeł

- Na początek upewniamy się, że mamy zainstalowane nagłówki do używanego przez nas jądra, pakiet *libsdl-dev* a także ustawiamy odpowiednio link `/usr/src/linux`:

```
# uname -a
2.6.16-1-686
# apt-get install linux-headers-2.6.16-1-686
# rm -f /usr/src/linux
# ln -s /usr/src/linux-headers-2.6.16-1-686 /usr/src/linux
# apt-get install libsdl-dev
```

- Następnie ze strony [fabrice.bellard.free.fr/qemu/download.html](http://fabrice.bellard.free.fr/qemu/download.html) (<http://fabrice.bellard.free.fr/qemu/download.html>) ściągamy źródła Qemu i Kqemu, przykładowo:

```
# wget http://fabrice.bellard.free.fr/qemu/qemu-0.8.0.tar.gz
# wget http://fabrice.bellard.free.fr/qemu/kqemu-0.7.2.tar.gz
```

Rozpakowujemy je (*kqemu* wewnątrz katalogu *qemu*):

```
# tar xzf qemu-0.8.0.tar.gz
```

```
# cd qemu-0.8.0
# tar xzf ../kqemu-0.7.2.tar.gz
# cd ..
```

- Kolejny problem dotyczy wersji kompilatora - qemu, przynajmniej w wersji 0.8 wymaga kompilatora gcc z serii 3.X, tymczasem w Debian unstable mamy obecną wersję 4.X. Przy próbie konfiguracji zobaczymy:

```
# ./configure
ERROR: "gcc" looks like gcc 4.x
...
```

Musimy więc zainstalować starszy kompilator:

```
# apt-get install gcc-3.4
```

a następnie skonfigurować qemu z odpowiednimi opcjami:

```
# ./configure --cc=/usr/bin/gcc-3.4 --target-list=i386-softmmu --enable-adlib \
--enable-alsa
```

Przykładowe opcje włączają obsługę dźwięku oraz wybierają do kompilacji tylko emulator PC.

Przeprowadzamy kompilację i instalację (to ostatnie jako root):

```
# make
# make install
```

Qemu zostało zainstalowane w katalogu /usr/local/bin.

- Następnie możemy spróbować załadować moduł *kqemu*, wykonując (jako root):

```
# modprobe kqemu
```

Jeżeli pojawi się informacja *Invalid Module Format*, to oznacza, że jądro zostało skompilowane innym kompilatorem niż użyliśmy do kompilacji Kqemu. Wersję gcc użytą do kompilacji jądra możemy sprawdzić:

```
# cat /proc/version
Linux version 2.6.16-1-686 (Debian 2.6.16-2) (tbm@cyrius.com) (gcc version
4.0.3 (Debian 4.0.3-1)) #1 Wed Mar 22 15:36:02 UTC 2006
```

Przykładowy wynik oznacza, że Kqemu także musimy skompilować za pomocą gcc 4.X. Na szczęście możemy to zrobić niezależnie od kompilacji pozostałej części Qemu, a następnie skopiować moduł ręcznie do odpowiedniego katalogu. Wywołujemy więc w katalogu ze źródłami Qemu:

```
# make -C kqemu clean
# ./configure --target-list=i386-softmmu --disable-gcc-check
# make -C kqemu
# cp kqemu/kqemu.ko /lib/modules/2.6.16-1-686/misc
# update-modules
```

### 3.2.29.4. Używanie Qemu

Strona podręcznika Qemu (**man qemu**) zawiera opis wszystkich opcji programu. Poniżej opisanych jest kilka przykładów typowego użycia.

- Aby włączyć akcelerację musimy upewnić się, że załadowany jest moduł *kqemu*. Ładujemy go komendą **modprobe kqemu** lub, jeśli korzystamy z *udev'a*, komendą **modprobe kqemu major=0**.
- Pierwszą rzeczą jaką będziemy musieli zwykle zrobić to utworzyć obraz dysku, na którym będziemy mogli np. zainstalować system. Robimy to poleceniem:

```
# qemu-img create test.img 2G
```

Przykładowe polecenie tworzy obraz o rozmiarze 2 GB w pliku `test.img`. Możemy teraz uruchomić Qemu korzystając z naszego obrazu:

```
# qemu -hda test.img -m 192
```

Opcja `-m` ustala wielkość pamięci w MB, domyślnie 128 (należy pamiętać, że Qemu może zajmować tyle fizycznej pamięci w naszym systemie...). Zgodnie z przewidywaniem zobaczymy komunikaty uruchamiającej się maszyny wirtualnej z informacją, że nie można załadować systemu z naszego dysku - co jest prawdą, bo żadnego systemu na nim nie ma.

- Za pomocą kombinacji klawiszy `Ctrl+Alt+F` przełączamy maszynę w tryb pełnoekranowy i z powrotem na okienko, `Ctrl+Alt` zwalnia myszkę z maszyny, `Ctrl+Alt+1` przełącza na główny ekran Qemu, `Ctrl+Alt+2` na konsolę monitorującą. W konsoli monitorującej możemy między innymi obejrzeć spis dostępnych poleceń (**help**), sprawdzić czy działa `kgemu` (**info kgemu**) a także zapisać lub załadować stan całej maszyny wirtualnej (**savevm/loadvm**).
- Aby na naszym utworzonym obrazie dysku zainstalować system Windows, musimy włożyć do napędu CD płytę instalacyjną Windows (zalecane ze względu na prędkość działania są wersje 98 lub 2000) i kazać Qemu załadować system z CD:

```
# qemu -hda test.img -cdrom /dev/cdrom -boot d
```

Przy instalacji Windows 2000 zalecane jest także dodanie opcji `-win2k-hack` - ale tylko przy instalacji, gdyż powoduje ona spowolnienie dysku.

**Notatka:** Uwaga - jeśli licencja Windows nie mówi inaczej, to nie można instalować ani uruchamiać więcej niż jednej kopii Windows mając jedną licencję.

- Teoretycznie możliwe jest (choć może nie zadziałać) załadowanie Windows zainstalowanego na dysku, przez wykonanie komendy (jako root):

```
# qemu -hda /dev/hda
```

Jest jednak duża szansa, że Windows się nie uruchomi (ze względu na inny sprzęt) - więcej informacji na ten temat można znaleźć na stronie wspomnianego FAQ

(<http://kidsquid.com/cgi-bin/moin.cgi/FrequentlyAskedQuestions>). Należy także zwrócić uwagę, żeby pod żadnym pozorem nie modyfikować dysku używanego przez Windows spod Linuksa.

- Ostatni przykład pokazuje, jak uruchomić system z obrazu dystrybucji Live takiej, jak np. Knoppix:

```
# qemu -cdrom knoppix.iso -boot d
```

### 3.2.30. NTFS-3g

*NTFS* jest systemem plików opracowanym przez firmę Microsoft, używanym w systemach Windows 2000 i XP. Jądro Linuksa pozwala na korzystanie z niego w ograniczonym zakresie. Możliwy jest odczyt ale zapis jest niebezpieczny i może prowadzić do uszkodzenia danych. Pełny dostęp do partycji NTFS oferuje moduł *ntfs-3g* korzystający z podsystemu Filesystem in User Space (FUSE). Moduł jest jeszcze w fazie BETA co oznacza, że istnieje ryzyko utraty danych. Jednak na liście dyskusyjnej projektu nie odnotowano informacji o poważnych problemach. Poniżej przedstawiony zostanie opis instalacji i korzystania z tego modułu w systemie Debian.

### 3.2.30.1. Instalacja

*ntfs-3g* wymaga modułu FUSE. Jest on obecny w jądrze od wersji 2.6.14. Dla starszych jąder konieczne jest zainstalowanie pakietu *fuse-source* i skompilowanie modułu przy pomocy *module-assistant*.

Nie zależnie od wersji jądra należy zainstalować pakiet *fuse-utils* oraz *libfuse-dev*. Pierwszy z nich zawiera on narzędzia niezbędne do korzystania z *ntfs-3g*, drugi wymagany jest do kompilacji.

Źródła modułu *ntfs-3g* należy pobrać ze strony <http://mlf.linux.rulez.org/mlf/ezaz/ntfs-3g-download.html>. Następnie należy je rozpakować i skompilować:

```
tar xzf ntfs-3g-XXXXXX-BETA.tgz
cd ntfs-3g-XXXXXX-BETA
./configure
make
```

Kolejnym krokiem jest instalacja, aby ją wykonać należy mieć uprawnienia roota.

```
make install
```

Jeśli podczas instalacji nie wystąpiły żadne błędy można korzystać z partycji NTFS.

### 3.2.30.2. Używanie

Przed próbą użycia *ntfs-3g* należy sprawdzić czy partycje NTFS nie są już zamontowane. Listę zamontowanych systemów plików otrzymuje się wydając polecenie: *mount*. Jeżeli w otrzymanej liście znajduje się wiersz z *type ntfs* należy wykonać polecenie:

```
umount /nazwa/katalogu
```

W miejsce */nazwa/katalogu* należy wstawić nazwę katalogu, do którego zamontowana jest partycja *ntfs*. Znajduje się ona między słowami *on* i *type* na listingu wyświetlonym przez *mount*. Po wydaniu komendy może pojawić się komunikat:

```
umount: /home: device is busy
```

Oznacza on, że jakiś program wykorzystuje pliki z zamontowanej partycji. Należy sprawdzić uruchomione programy i zamknąć te, które korzystają z danej partycji. Następnie ponownie wykonać komendę *umount*. Gdy partycja zostanie odmontowana, można przystąpić do zamontowania jej przy użyciu *ntfs-3g*.

W celu zamontowania partycji należy wydać komendę:

```
ntfs-3g /dev/hdXX /mnt/YYY
```

gdzie w miejsce *XX* wpisujemy nazwę partycji, a w *YYY* nazwę katalogu, w którym będzie ona zamontowana. Nazwę partycji i katalog można znaleźć na uzyskanej wcześniej liście z komendy *mount*.

## 3.2.31. Konfiguracja i użytkowanie środowiska DOSEMU w systemie GNU Debian

Aby zainstalować *dosemu* należy wydać komendę:

```
# apt-get install dosemu
```

W przypadku, gdy *dosemu* będzie używane w środowisku *X11* dobrze jest zainstalować dodatkowe czcionki:

```
# apt-get install xfonts-dosemu
```

jednak zalecane jest ściągnięcie plików binarnych lub źródłowych *dosemu* ze strony projektu *dosemu* (<http://dosemu.sourceforge.net/stable/>) z uwagi na szybsze i bardziej stabilne działanie programu.

Ponieważ samo *dosemu* jest jedynie emulatorem komputera zoptymalizowanego pod kątem uruchamiania systemu operacyjnego *DOS*. Aby można było korzystać z *DOS* konieczne jest jego zainstalowanie. Można to zrobić na dwa sposoby:

- Zainstalować *freedos*. W przypadku gdy *dosemu* był instalowany jako gotowy pakiet należy wydać komendę:

```
# apt-get install dosemu-freedos
```

Gdy *dosemu* był instalowany ręcznie z plików binarnych lub źródłowych należy ściągnąć *freedos* do odpowiedniej jego wersji.

- Zainstalować dowolny system operacyjny *DOS* na obrazie dysku. Ten sposób jest analogiczny jak w przypadku systemów Red Hat i został opisany tutaj, dlatego też nie będzie dalej opisywany w tym rozdziale.

### 3.2.31.1. Konfiguracja systemu *freedos* pracującego pod kontrolą *dosemu*

W niniejszym rozdziale zostanie opisana konfiguracja jedynie najważniejszych i najczęściej używanych rzeczy:

- konfiguracja *autoexec.bat* i mapowanie dysków. Plik *autoexec.bat* znajduje się w katalogu:

```
/etc/freedos/autoexec.bat
```

Standardowo po zainstalowaniu *freedos* są podmapowane dyski:

- *C:* - tylko do odczytu, jest to najczęściej katalog */usr/lib/freedos*
- *D:* - do odczytu i zapisu, jest to najczęściej katalog użytkownika
- *E:* - do odczytu i zapisu, jest to najczęściej katalog tymczasowy wygenerowany automatycznie przez *dosemu* jako podkatalog */tmp/*

Istnieje też możliwość podmapowania dowolnego katalogu na dowolną (wolną) literę dysku. W tym celu w *freedos* należy wydać polecenie:

```
lredir LITERA: linux\fs/KATALOG
```

- Wykorzystywanie portów szeregowych w *dosemu*. W celu umożliwienia dostępu programom uruchamianym przez *dosemu* należy wyedytować plik:

```
/etc/dosemu/dosemu.conf
```

i odnaleźć sekcję *Serial port settings*. Aby zmapować wbudowany port szeregowy */dev/ttyS0* na urządzenie widziane w *dos* jako *COM1* należy wpisać lub odkomentować:

```
$_com1 = "/dev/ttyS0 irq 4"
```

Aby zmapować port szeregowy */dev/ttyUSB0* z przejściówki *USBRS-232* na *COM1* należy wpisać:

```
$_com1 = "/dev/ttyS0 irq 4"
```

### 3.2.31.2. Instalacja i konfiguracja programu Versman

*Versman* to nakładka na kompilator *Keil* oraz narzędzie do zarządzania logami programów technologicznych sterowników *ZET* zawartych w drzewie *CVS programy*. Szczegółową dokumentację programu znaleźć w pliku:

*CVS programy/Utils/versman/versmanwin.html*

Program *Versman* można znaleźć w katalogu:

*CVS programy/Utils*

Do poprawnej pracy programu *Versman* konieczne będą jeszcze następujące programy:

- *Keil 3.20* (katalog *C51PDK*).
- *Perl dla DOS* (katalog *PERL*).

Powyższe dwa programy można znaleźć w katalogu:

*CVS programy/Utils*

Aby zainstalować program *Versman* należy:

- Skopiować z *CVS programy* katalogi *Versman*, *C51PDK*, *PERL* do katalogu głównego użytkownika
- Na końcu pliku *autoexec.bat* dodać linie:

```
PATH %PATH%;X:\PERL;X:\VERSMAN;X:\C51PDK\bin
SET TEMP=Y:\TEMP
set C51LIB=X:\C51PDK\LIB
set C51INC=X:\C51PDK\INC
SET USER=Z
SET WORKPATH=X:\PROGRAMY
SET EPROMPATH=X:\PROGRAMY\EPROM
SET perllib=X:\PERL\LIB
gdzie:
```

- litera *X* oznacza literę dysku, która jest podmapowana do katalogu użytkownika (W przypadku standardowej konfiguracji *freedos* jest to litera *D*).
- litera *Y* oznacza literę dysku na którym można przechowywać pliki tymczasowe (W przypadku standardowej konfiguracji *freedos* jest to litera *E*).
- litera *Z* oznacza numer programisty. W przypadku nowego pracownika numer powinien być skonsultowany z innymi programistami, w celu wybrania właściwego/wolnego numeru.

Zmienne środowiskowe *WORKPATH* i *EPROMPATH* wskazują na ścieżkę do repozytorium *CVS programy*

- Odnaleźć w pliku *X:\versman\versman.cfg* (gdzie *X*: litera podmapowanego dysku użytkownika) sekcje i ewentualnie zmodyfikować literę dysku:

```
KEILLIB:
X:\c51pdk\lib
```

- Odnaleźć w pliku X:\versman\versman.pl (gdzie X: litera podmapowanego dysku użytkownika) sekcje i ewentualnie zmodyfikować literę dysku:

```
§cfg          = "X:\\Versman\\versman.cfg"; # nazwa pliku konfiguracyjnego VersMana
```

Po wykonaniu powyższych czynności można już używać programu *Versman*.

### 3.3. Terminal z MS Windows

Przeważnie na terminalu instalowane są Win9x/2000/XP. Robi się to przede wszystkim po to, żeby umożliwić użytkownikom korzystanie ze znanych im programów (np. aplikacji pakietu MS Office itp.).

W trakcie instalacji systemu Windows należy zainstalować protokół/protokoły TCP/IP i powiązać je z kartą sieciową (w Windows 2000/XP są one instalowane standardowo). Numer IP terminala musi się różnić od numeru IP serwera tylko ostatnią liczbą - w przeciwnym razie łącząca je sieć będzie musiała być kategorii wyższej niż C, co jest kłopotliwe i zbędne. Najlepiej też wyrzucić kartę Dialup.

Jeżeli instalowane jest Windows 98 SE, to ustawiany jest domyślnie polski układ klawiatury (być może ta niedogodność występuje również w innych wersjach Windows). Korzystanie ze zwykłych (nie-polskich) klawiatur jest wtedy bardzo uciążliwe (zamiast dwukropka, średnika i innych znaków mamy polskie literki). Dlatego zaraz po zainstalowaniu należy ten układ zmienić. W tym celu należy otworzyć folder **Mój Komputer**, w nim folder **Panel Sterowania**, tam odszukać ikonę o nazwie **Klawiatura** i uruchomić. Pojawi się okienko o nazwie „Właściwości: Klawiatura”. W nim uaktywniamy zakładkę **Język**, a tam w okienku „Zainstalowane Języki i Układy Klawiatury” odszukujemy język polski / układ polski i klikamy na przycisk **Właściwości**. Zmieniamy układ z „polskiego” na „polski (programisty)”. Dodatkowo należy w pliku C:\autoexec.bat zmodyfikować wywołanie:

```
keyb pl,,C:\WINDOWS\COMMAND\keybrd2.sys
```

na

```
keyb pl,,C:\WINDOWS\COMMAND\keybrd4.sys
```

Po reboocie wszystko powinno działać normalnie.

#### 3.3.1. Instalacja, konfiguracja i używanie pakietu Cygwin

Cygwin jest to, zgodnie z zamysłem twórców, środowisko umożliwiające działanie pod systemem Windows tych aplikacji, które zostały stworzone z myślą o systemie Unix. Pakiet Cygwin jest zwykle instalowany w następujących przypadkach:

- gdy chcemy skorzystać z (darmowego) X-serwera Cygwin/Xorg,
- gdy chcemy użyć narzędzi typu **rsync** oraz **ssh** i **sshd** pod systemem Windows.

Środowisko Cygwin składa się z:

- biblioteki procedur (*cygwin.dll*), zamieniającej wywołania funkcji uniksowych na wywołania systemu Windows,
- zestawu programów skompilowanych tak, żeby z tej biblioteki korzystały.

W ten sposób użytkownik systemu Windows może mieć standardowo dostęp do wielu podstawowych narzędzi uniksowych takich jak: **bash**, **awk**, **grep**, **less**, **tail**, **head**, **iconv**, **cat** oraz bardziej złożonych programów takich jak: **apache**, **ssh**, **rsync** oraz **Xorg**. Bardzo ważnym elementem systemu Cygwin jest aktualnie X-serwer (z pakietu **Xorg**). Pozwala on na łączenie się z systemem Unix/Linux i zdalną pracę na maszynach uniksowych/linuxowych.

### 3.3.1.1. Instalacja

Możliwe są 2 sposoby instalacji środowiska Cygwin:

- Z sieci, ze strony <http://www.cygwin.com>.
- Z nośnika dostępnego lokalnie, np. płyty albo twardego dysku.

Ponieważ system rozwija się bardzo dynamicznie (najnowsze wersje są najpełniejsze i najstabilniejsze), więc przed instalacją z nośnika lokalnego należy zwrócić uwagę na to, żeby wersja instalacyjna była możliwie najnowsza. Problemu tego nie ma podczas instalacji sieciowej (gdyż używane są tu bieżące wersje pakietów).

- Zalecaną metodą jest instalacja *tylko wybranych pakietów*. Wybór pakietów można przeprowadzić podczas instalacji z sieci, można również stworzyć własną dystrybucję systemu Cygwin z tylko tymi pakietami, które są najważniejsze (sposób jak zrobić własną dystrybucję Cygwina i propozycja wyboru pakietów znajduje się tutaj).
- Można również ściągnąć gotową dystrybucję w formie obrazu ISO i zainstalować ją całą bądź jej fragment. W Internecie dostępnych jest wiele dystrybucji (np. na <http://inst.eecs.berkeley.edu/~instcd/iso/>) są obrazy, które można bezpośrednio wypalić na płycie.
- Cygwina w wersji instalacyjnej można również znaleźć na wielu serwerach, w tym - polskich, np. na <ftp://sunsite.icm.edu.pl/pub/cygwin/>.

W celu instalacji, czy to sieciowej czy lokalnej, uruchamiamy program **setup.exe** (który znajduje się zarówno w sieci, pod adresem <http://www.cygwin.com/setup.exe>, jak i na każdej wersji instalacyjnej z płyt). Program **setup.exe**, oprócz rozpakowywania i kopiowania pakietów, przygotowuje dla użytkownika całe środowisko dla Cygwina. Po uruchomieniu **setup.exe** należy postępować zgodnie z instrukcjami. Poniżej przedstawiamy ogólny opis postępowania:

- W oknie "Cygwin Setup - Choose Installation Type" wybieramy opcję "Install from Local Directory" (oczywiście jeśli instalujemy z płytki lub dysku twardego). W następnym oknie, o nazwie "Cygwin Setup - Choose Installation Directory", zostawiamy domyślne opcje "Install For All Users" oraz "Default Text File Type: Unix" a następnie naciskamy przycisk "Dalej". W oknie "Cygwin Setup - Select Local Package Directory" wybieramy katalog instalacyjny Cygwin (zwykle odpowiedni katalog jest automatycznie wybierany) i naciskamy "Dalej".
- Domyślnie w oknie "Cygwin Setup - Select Packages" instalowana jest podstawowa, niewielka dystrybucja Cygwina (**bash** oraz kilka podstawowych programów i bibliotek), która zajmuje na dysku ok. 30 MB.
- Żeby móc korzystać z możliwości opisanych poniżej, trzeba zainstalować programy: **rsync**, **openssh** a także **Xorg**.
- Dodatkowo mogą się przydać pakiety: **iconv**, **perl** (jest to stosunkowo duży pakiet), **tcl** no i oczywiście **vim** (lista pakietów zalecanych do zainstalowania jest tutaj).

- Opcję zainstalowania lub niezainstalowania pakietu (lub grupy pakietów) wybiera się klikając myszką obok nazwy pakietu, przy "pętelce" (uwaga - czasem przełączenie opcji trwa kilka chwil). Mamy tu możliwości: „Default” (domyślnie), „Install” (zainstaluj), „Reinstall” (przeinstaluj) oraz „Uninstall” (odinstaluj).
- Uwaga! W szczególności możemy wybrać opcję „Install” dla całego drzewa pakietów („All”), żeby zainstalować wszystkie pakiety (co jest dosyć wygodne), np. jeżeli instalacja przebiega z przygotowanej wcześniej dystrybucji jak to opisuje Sekcja 3.3.2. Gdy instalujemy system z sieci, instalacja może potrwać w tym przypadku dosyć długo i zajmie ponad 1GB miejsca na dysku.
- Tak zainstalowany pakiet Cygwin zajmuje, zależnie od użytej dystrybucji, od ok. 300 MB do ok. 1 GB (stan na maj 2003). W oknie "Cygwin Setup - Create Icons" należy zaciśnąć "Zakończ" przy zaznaczonych opcjach "Create Icon on Desktop", "Add Icon to Start Menu". W pełnej wersji dostaniemy tu wszystkie narzędzia developerskie, kompilatory, debuggera i całą masę innych mniej lub bardziej przydatnych pakietów.
- Uwaga! Podczas instalacji sieciowej można wybrać jeden z wielu serwerów lustrzanych (czyli tzw. *mirrorów*). Czasami instalacja jest przerywana np. komunikatem „Download Incomplete”. Należy wówczas spróbować innego serwera.
- Po zainstalowaniu należy sprawdzić system, poprzez uruchomienie konsoli tekstowej, która znajduje się w menu „Start/Cygwin/Cygwin Bash Shell” (jeżeli w czasie instalacji wybierzemy, że chcemy, żeby skrót do konsoli tekstowej pojawił się na pulpicie, będziemy mogli uruchamiać środowisko Cygwin również dwoma kliknięciami na ikonkę na pulpicie). Jeżeli wszystko działa (konsola się uruchamia), to warto zdefiniować w Windowsie zmienną środowiskową CYGWIN o wartości „ntsec tty” (w Windows XP robimy to wchodząc do Start/Panel Sterowania/System, następnie wybierając zakładkę Zaawansowane wybierając przycisk Zmienne Środowiskowe i klikając „Nowa” przy polu Zmienne Systemowe). W oknie zmiennych środowiskowych warto również do zmiennej PATH dodać ścieżkę dostępu do programów wykonywalnych Cygwina (zwykle jest to C:\CYGWIN\BIN). Pozwoli nam to korzystać z programów Cygwina nie tylko wywołując komendy z poziomu basha, lecz również uruchamiając je z poziomu skryptów .bat.

Instalację i konfigurację systemu X Window opartego na Cygwinie opisuje sekcja Sekcja 3.3.3.2.

### 3.3.1.2. Użytkowanie

Środowisko Cygwin uruchamia się zwykle w sposób interaktywny (klikając na ikonę, która *de facto* uruchamia powłokę **bash**). Uruchomi się konsola, w której można wydawać komendy, podobnie jak w terminalu linuxowym. Można uruchamiać wiele konsol systemu Cygwin równocześnie. Procesy systemu Cygwin (takie jak **apache**, **inetd** czy **sshd**) mogą jednak również działać w tle, jako usługi.

Trzecią możliwością jest wywoływanie skryptów batchowych (bez potrzeby uruchamiania powłoki), które wywołują komendy systemu Cygwin. W tym celu należy stworzyć Windowsowy plik .bat o zawartości (zakładamy, że Cygwin zainstalowany jest w katalogu C:\CYGWIN):

```
cd C:\
cd CYGWIN\BIN

bash --login -c "<komenda do wywołania>"
```

Kilku słów komentarza wymaga sposób dostępu do zasobów dyskowych spod Cygwina. Otóż programy cygwinowe widzą system plików w sposób uniksowy: rootem / jest tu katalog c:\cygwin. Wszystkie dyski DOS-owe / Windowsowe są udostępnione jako /cygdrive/<nazwa dysku>/, np. dysk c:\ jest

udostępniony jako `/cygdrive/c`, dysk `d:\` jako `/cygdrive/d` itd. W ten sposób mamy dostęp do zasobów Cygwina na dwa sposoby: `/` oraz `/cygdrive/c/cygwin/`.

W systemie Cygwin konta użytkowników nie są tym samym czym w Linuksie. Obecne są wprawdzie pliki `/etc/passwd` i `/etc/group`, lecz odzwierciedlają one jedynie użytkowników systemu Windows („główna kopia” danych znajduje się w plikach systemowych Windowsów). Pliki `/etc/{passwd,group}` powstają w trakcie instalacji systemu Cygwin i nie są automatycznie poprawiane po dodaniu bądź usunięciu użytkownika. Niezmieniane, odzwierciedlają stan na dzień instalacji systemu Cygwin. Żeby wygenerować zawartość pliku `/etc/passwd`, należy wywołać komendę **mkpasswd**. Komenda **mkgroup** wygeneruje z kolei zawartość pliku `/etc/group`. Obie komendy przyjmują następujące parametry: „-l” oznacza konta lokalne, „-d” - konta domenowe. Ponieważ obie komendy zwracają zawartości plików na standardowe wyjście, zwykle wywołuje się je w następujący sposób:

```
mkpasswd -l >/etc/passwd
mkgroup -l >/etc/group
```

Również zawartość wygenerowanych plików może się wydawać „dziwna”. W polu opisowym znajduje się ciąg znaków, np. `S-1-5-21-2130074997-1700076171-1250845650-4684`. Jest to pełny identyfikator użytkownika, grupy, zasobu w systemie Windows. Jego znaczenie jest więc szersze niż uniksowego UID-a. Funkcję UID-a w pliku `/etc/passwd` spełnia w tym przypadku ostatni człon - tutaj 4684.

### 3.3.1.3. Uruchamianie serwerów `sshd` i `crond`

Możliwe jest zainstalowanie pod kontrolą Cygwina (przynajmniej na Windows z serii NT - a więc NT, 2000 i XP) linuksowych demonów, takich jak `sshd` czy `crond`. W tym celu należy:

- *Zainstalować odpowiednie pakiety.* Uruchamiamy program instalacyjny cygwina (`setup.exe`) i wybieramy bądź instalację z internetu, bądź też z lokalnego nośnika, na przykład płytki. Klikamy na guzik **View** i przełączamy typ widoku z *Categories* na *Full*. Zaznaczamy do instalacji pakiet `cygrunsrv` oraz pakiet dla odpowiedniego demona, np. `OpenSSH` czy `crond`, dodatkowo także np. `rsync`. Instalujemy wybrane pakiety.
- *Instalacja `sshd` jest prosta, sprowadza się do uruchomienia skryptu `ssh-host-config` (obecny w ścieżce, wymaga uprawnień administratora). Na zadawane pytania odpowiadamy *yes*. Na pytanie o zmienne środowiskowe (proponowana odpowiedź *ntsec*) odpowiadamy *ntsec tty*. Upewniamy się także, że zmienna systemowa Windows o nazwie *CYGWIN* ma właśnie wartość *ntsec tty* (zobacz Sekcja 3.3.1.1). Aby uruchomić serwer, wydajemy komendę:*

```
cygrunsrv --start sshd
```

Od tej chwili `sshd` powinien być uruchamiany automatycznie przy starcie Windows. Można go zastopować przez wydanie komendy

```
cygrunsrv --stop sshd
```

- *Instalacja `cron`'a może napotkać kilka problemów. Teoretycznie powinna się sprowadzać do wydania następujących komend:*

```
cygrunsrv -I cron -p /usr/sbin/cron.exe
cygrunsrv --start sshd
```

Jeżeli zobaczymy jednak komunikat o błędzie, to musimy zrobić kilka dodatkowych rzeczy. Po pierwsze odinstalowujemy usługę `crond` komendą:

```
cygrunsrv --top sshd
```

```
cygrunsvr -R cron
```

Następnie upewniamy się, że istnieje katalog `/etc/cron.d` i ma prawa do zapisu (najlepiej, niestety, dla wszystkich):

```
mkdir /etc/cron.d
chmod +w /etc/cron.d
```

Jeszcze tylko jedna drobna rzecz:

```
chmod +x /bin/cygwin1.dll
```

i możemy ponownie spróbować zainstalować i uruchomić usługę:

```
cygrunsvr -I cron -p /usr/sbin/cron.exe
cygrunsvr --start sshd
```

Jeżeli nadal pojawiają się błędy, to ich przyczyn należy szukać w pliku `/var/log/cron.log`.

Uruchomione w ten sposób 'demony' nie są widoczne jako uniksowe procesy. Są natomiast pełnoprawnymi usługami Windows - można je obejrzeć przez *Ustawienia -> Narzędzia administracyjne -> Usługi*. Można też zmieniać sposób ich uruchamiania.

### 3.3.2. Tworzenie własnej dystrybucji Cygwina

Pełny system Cygwin zajmuje dzisiaj już ponad 1GB na dysku. W rozdziale niniejszym opisany jest sposób stworzenia dystrybucji znacznie mniejszej, zawierającej narzędzia: **rsync**, **ssh**, **sshd** oraz **cygwin/Xorg**. Dodatkowo w dystrybucji znajdzie się kilkadziesiąt przydatnych programów uniksowych i bibliotek (tzw. baza - *Cygwin-base*).

Przygotowanie dystrybucji jest bardzo proste. W kroku pierwszym należy ściągnąć dokładnie te pakiety, które nas interesują (wypisane poniżej). Wykorzystujemy w tym celu standardowy program `setup.exe` (<http://www.cygwin.com/setup.exe>), gdzie należy wybrać opcję "Download from Internet", wybrać katalog docelowy (w okno "Local Package Directory" wpisać/wybrać nazwę, np. `C:\CYGWIN-INSTALL`) i nacisnąć "Dalej".

Następnie należy wybrać optymalny rodzaj połączenia (zwykle „Use IE 5 Settings”) i ponownie nacisnąć "Dalej". W oknie "Available Download Sites" podświetlić serwer, z którego będą ściągane pakiety (najlepiej z domeny .pl).

Po następnym naciśnięciu "Dalej" należy dokonać odpowiedniego wyboru pakietów. Po tym uruchomione zostanie ściągnięcie pakietów z sieci. Ściągnięty zestaw pakietów umieszczany jest we wskazanym katalogu, np. `C:\CYGWIN-INSTALL`, w podkatalogu o przedziwnej nazwie rozpoczynającej się od słowa „ftp”, który ma w nazwie zakodowaną również nazwę mirrora, z którego właśnie ściągnęliśmy całość (mam nadzieję, że opis jest wystarczająco obrazowy, żeby podkatalog odnaleźć bez trudu). Wraz w tym podkatalogiem w katalogu `C:\CYGWIN-INSTALL` musi znaleźć się program `setup.exe`, który także ściągamy.

Innym sposobem jest ściągnięcie całości pakietów (lub na przykład skorzystanie z istniejącej dystrybucji ISO i skasowanie tych, które są zbyteczne). Poszczególne pakiety są umieszczone w podkatalogach w katalogu `release` (który sam znajduje się w katalogu o przedziwnej nazwie, p. wyżej). Żeby pozbyć się danego pakietu z dystrybucji, należy usunąć cały podkatalog z pakietem (np. `apache`).

Gdy jest przygotowany katalog z pakietami i programem `setup.exe`, można już z niego zaczynać instalowanie system Cygwin. Wybieramy tu opcję "Install from Local Directory". Katalog, z którego instalujemy, nie wymaga prawa do zapisu, więc można go wypalić na płycie i instalować bezpośrednio z płyty.

Uwaga! W czasie wybierania pakietów do instalacji w opisanej dystrybucji musimy wybrać opcję „Install” dla całego drzewa pakietów (sekcji All).

Pakiety do zainstalowania (w stosunku do standardowej wersji „wszystko Default”):

- W kategorii **Admin**: dodajemy **cygrunsrv**.
- W kategorii **Net** dodajemy: **openssh, openssl, rsync**.
- W kategorii **Xorg** dodajemy: **xorg-x11-base**. Uwaga! Z powodu zależności między pakietami, automatycznie zostanie dodanych kilka innych pakietów z rodziny xorg-x11-.

Powyższa wersja instalacyjna ma około 30MB, natomiast system Cygwin po zainstalowaniu zajmie ok. 100MB.

### 3.3.3. X-terminale w systemie Windows

Rozdział ten opisuje sposób instalacji i konfiguracji X-serwerów (WinaXe i Cygwin/Xorg). X-serwer jest to oprogramowanie służące do wzbogacenia hosta o funkcjonalność X-terminala (co umożliwia zdalną pracę w trybie graficznym na systemach linuksowych). Dodatkowo, w rozdziale omówiono kilka spraw pokrewnych (tunelowanie połączeń X-owych).

#### 3.3.3.1. Instalacja XSerwera Labf.com

Jednym z najlepszych XServerów pod Windows jest WinaXe (dawniej XWin-Pro) fińskiej firmy Labf.com (dawniej Labtam). Dystrybucja to jeden plik `winaxe.exe` (`Xwp32.exe`) z programem (ok. 6MB) + plik z licencją `licence.exe` (ok. 60kB).

- W katalogu tymczasowym uruchomić `winaxe.exe/Xwp32.exe` - archiwum samorozpakowujące się; standardowo rozpakowuje się do katalogu `c:\xinstall` - lepiej nie zmieniać.
- Jeśli jest wykupiona licencja, w oknie WinZip Self-Extractor wyłączyć opcję **When Done Unzipping Run: SETUP.EXE** i nacisnąć **Unzip**, po zakończeniu - **Close**; potem skopiować plik `licence.exe` do katalogu instalacji - `c:\xinstall` i uruchomić (to jest też samorozpakowujące się archiwum) tak, jak `winaxe.exe/Xwp32.exe`.
- W katalogu `c:\xinstall` uruchomić `Setup.exe`, wersja instalacji **Typical**.
- Przekopiować do miejsca zainstalowania (zwykle `c:\Program Files\Labf.com\WinaXe`, w starszych wersjach `c:\Program Files\Labtam\X-WinPro`) plik klawiatury `jojo.kmf`. Od wersji 6.0 WinaXe'a plik ten nie jest potrzebny - standardowo w dystrybucji znajduje się właściwa klawiatura `polish.kmf`.
- Można korzystać z fontów przez xserwer albo (szybsze) z lokalnych. Z różnych powodów (m.in. większa stabilność) zaleca się to drugie. Jeżeli terminal ma pracować z lokalnymi fontami należy:
  - utworzyć podkatalog, np. `75dpi1`, w miejscu zainstalowania fontów WinaXe: `c:\Program Files\Labf.com\WinaXe\fonts` (w starszych wersjach `c:\Program Files\Labtam\X-WinPro\Fonts`).
  - Zalogować się przez telnet do serwera (telnet `ip_serwera`) i do pomocniczego katalogu np. `/temp` skopiować cały katalog z fontami polskimi 75dpi (gdzie jest ten katalog można znaleźć zaglądając w sekcję `FontPath` w pliku `/etc/X11/XF86Config`. Jeżeli jest tam wpis `"unix/:-1"`, to znaczy, że jest uruchomiony font server i ścieżki można odczytać z pliku `/etc/X11/fs/config` z sekcji `catalogue`).

- Aby sprawdzić, które katalogi zawierają polskie fonty, należy sprawdzić wpisy w plikach `fonts.dir` poszczególnych katalogów czy są tam wpisy zakończone `iso8859-2` (standard kodowania polskich liter). Zwykle ta nazwa jest wpleciona w ścieżkę katalogu z fontami, np. w RH v6.1 jest to `/usr/share/fonts/ISO8859-2/75dpi`.
- Po przegraniu zawartości katalogu, np.
 

```
cp /usr/share/fonts/ISO8859-2/75dpi/* /temp/
```

 należy rozpakować wszystkie fonty (nie jest to konieczne w wersjach WinaXe 6.2 i późniejszych):
 

```
gzip -d /temp/*
```

 (gzip powinien napisać, że zignorował dwa pliki: `fonts.alias` i `fonts.dir` a resztę rozpakował).
- W pliku `fonts.dir` należy usunąć z wszystkich nazw plików końcówkę `".gz"` (w przypadku wersji WinaXe 6.2 lub późniejszej, jeżeli fonty nie zostały rozpakowane, nie należy zmieniać zawartości `fonts.dir`).
- W Windows z utworzonego dla nowych fontów katalogu należy się połączyć przez ftp z serwerem (**ftp ip serwera**), ustawić transfer na binarny (**binary**), przejść do podkatalogu z przygotowanymi fontami (**cd /temp/**), wyłączyć tryb interaktywny (**prompt**) i ściągnąć wszystko (**mget \***) i wreszcie zakończyć (**bye**).
- Lokalnie na Windows należy zmienić nazwę pliku `fonts.alias` na `fonts.ali`.
- w sesji telnetu wymazać cały katalog pomocniczy:
 

```
rm -rf /temp/
```

 I można już zakończyć sesję telnet.
- Uruchomić XSettings z pakietu WinaXe i ustawić tam:
  - RGB Path: `rgb.txt`.
  - Keyboard Settings \ Keyboard File List: `jojo.kmf` (to jest plik przygotowany przez firmę PRATERM i należy go wcześniej umieścić w katalogu z WinaXe) lub w WinaXe 6.x: `polish.kmf`.
  - Keyboard Settings \ Block Key Syms Changing: `+` (w WinaXe 6.x standardowo zaznaczone).
  - Startup \ Window Mode List: `Full Screen`.
  - Startup \ XDMCP \ Use XDMCP: `+` w oknie wywoływanym przyciskiem Settings w Startup \ Window Mode List \ XDMCP ustawić:
    - XDMCP mode: `Query`.
    - X depth (bits per pixel): `Auto`.
    - Reset XServer at XDMCP Close: `+` (w WinaXe 6.x standardowo zaznaczone).
    - Connect Host: `ip_serwera`.
 W oknie XSettings w oknie wywoływanym przyciskiem Font Control \ FontPath ustawić:
    - Przez Browse wsadzić do okienka Font directory ścieżkę z przekopiowanymi polskimi fontami albo fontserwer - wpis postaci: `tcp/ip_serwera:port`, gdzie „port” przyjmuje zwykle wartość 7100.
    - Zaznaczyć pierwszą ścieżkę w oknie Priority ordered path i nacisnąć przycisk **Insert Before**, potem **Save**.
- Reszta ustawień bez znaczenia.

- Na koniec nacisnąć **OK** w głównym oknie XSettings, w dialogu z zapytaniem o plik jojo.kmf odpowiedzieć twierdząco.
- Na serwerze, należy skonfigurować odpowiednio program **xdm** (lub jeden z pochodnych: **kdm** bądź **gdm**), tak jak to opisuje Sekcja 3.3.3.5.
- XWin-Pro / WinaXe jest gotów do uruchomienia.

### 3.3.3.2. Instalacja X-serwera Cygwin/Xorg

System Cygwin/Xorg jest darmowym X-serwerem z otwartymi źródłami. Stanowi dziś najpoważniejszą konkurencję dla produktów komercyjnych (typu WinaXe). Nie dostarcza wprawdzie tak wielu udogodnień a jego interfejs jest dosyć „surowy”, ale bieżące wersje działają bardzo stabilnie i spełniają swoje podstawowe funkcje. Najważniejszą funkcją X-serwera Cygwin/Xorg jest umożliwienie komputerowi PC zdalnej pracy na maszynach uniksowych/linuxowych (komputer PC staje się wówczas tzw. X-terminalem). X-terminal jest więc tylko „wyświetlarką” informacji z aplikacji, która w rzeczywistości działa na innym, zdalnym komputerze.

Żeby zainstalować Xorg na systemie Windows, trzeba wcześniej skonfigurować i zainstalować środowisko Cygwin. Pełną informację o możliwych sposobach instalacji cygwina zawiera Sekcja 3.3.1. Minimalna, w pełni funkcjonalna wersja (bez obsługi polskiej klawiatury), może być ściągnięta ze strony <https://sourceforge.net/projects/szarp/files/Cygwin%20X.Org%20Viewer/xorg-setup.exe/download> (<https://sourceforge.net/projects/szarp/files/Cygwin%20X.Org%20Viewer/xorg-setup.exe/download>). Przy instalacji pyta się ona jedynie o adres IP serwera X-ów. Może być także konieczne zastosowanie opcji *-from* (zobacz dalej).

Inny, dość prosty i ogólny sposób sprowadza się do następujących czynności:

1. Uruchomić program ściągnięty z [www.cygwin.com/setup.exe](http://www.cygwin.com/setup.exe) (<http://www.cygwin.com/setup.exe>).
2. Potwierdzić domyślny typ instalacji, domyślne opcje i sposób połączenia z internetem (trzy kolejne ekrany).
3. Wybrać jeden z serwerów lustrzanych, z których będą ściągane pakiety. Jeżeli dane z jakiegoś serwera są ściągane wolno lub w ogóle, należy spróbować z innym serwerem.
4. Wybrać pakiety do instalacji - w tym celu na górze po prawej stronie okienka do wyboru pakietów klikamy na guzik *View*, aby uzyskać alfabetyczną listę pakietów, a potem klikamy na napisie *Skip* w linii z pakietem *xorg-x11-xwin* - zostanie zaznaczony do instalacji pakiet z X-serwerem oraz kilka innych pakietów. Zatwierdzamy nasz wybór.
5. Czekamy aż instalator ściągnie i zainstaluje wybrane pakiety - może to trochę potrwać. Na końcu możemy wybrać dodanie pozycji w menu start i utworzenie skrótów na pulpicie.

*Uwaga! Cygwin/Xorg, w przeciwieństwie do reszty programów pakietu Cygwin, jest systemem graficznym, więc ma większe wymagania. Korzysta z systemu DirectX. Musi być zainstalowana wersja co najmniej 4. Problem może pojawić się w przypadku starszych wersji systemu Windows (95, 98), nie dotyczy jednak nowszych systemów (Me, 2000, XP), gdzie system DirectX jest standardowo obecny.*

Konfiguracja X-terminala nie jest skomplikowana. W pierwszym kroku musimy odszukać plik o nazwie `startxdmcp.bat`. Powinien znajdować się on w katalogu `C:\CYGWIN\USR\X11R6\BIN` (jeżeli system Cygwin został zainstalowany na dysku C:). Przegrywamy ten plik w miejsce, z którego będzie nam wygodnie uruchamiać system Cygwin (np. na pulpit). Następnie musimy wyedytować ten plik (np. za pomocą edytora Notepad). Modyfikujemy tu następującą linię w pliku:

```
SET REMOTE_HOST=10.0.0.1
```

Powinniśmy ustawić `REMOTE_HOST` jako adres serwera uniksowego/linuksowego, z którym będziemy się łączyć.

Jeżeli chcielibyśmy skorzystać z serwera fontów, to należy przejść na koniec pliku i odnaleźć linię wywołania X-serwera (programu **XWin**):

```
REM Connect to a specified machine.

%RUN% Xwin -query %REMOTE_HOST% -nodecoration -lesspointer

...

REM Connect to a specified machine and a specified font server.

REM %RUN% Xwin -query %REMOTE_HOST% -nodecoration -lesspointer -fp \
tcp/%REMOTE_HOST%:7100
```

Można ją rozpoznać po braku „REM” na początku. Linię tę należy wykomentować dodając „REM” a odkomentować linię zawierającą opcję **-fp tcp/...**

Jeśli wystąpią problemy z obsługą klawiatury (powtarzanie wpisanych znaków), należy dodać **-kb** na końcu linii wywołania **Xwin**.

Dodatkowo, jeżeli Cygwin jest zainstalowany na komputerze z systemem Windows 98, to konieczne jest podanie w linii wywołania programu **Xwin** numeru IP komputera z Windows. Przykładowo, jeżeli komputer ma numer IP 192.168.0.100, to na końcu linijki z wywołaniem program **XWin** dopisujemy:

```
-from 192.168.0.100
```

Zdarza się, że taki wpis musimy też zrobić na komputerze z nowszym systemem (także Windows XP), np. gdy komputer ma kilka interfejsów sieciowych.

Po zmodyfikowaniu pliku, zapisujemy zmiany i wychodzimy z edytora. Po uruchomieniu powinniśmy się połączyć z serwerem i po kilku sekundach otrzymać graficzne okno logowania się do systemu.

Jeżeli z jakiegoś powodu nie otrzymujemy okna logowania się do systemu, to powinniśmy sprawdzić czy to „sprawka” ustawień sieci czy raczej złej instalacji X-serwera (np. braku niezbędnych pakietów). W celu sprawdzenia czy system X-Window został poprawnie zainstalowany, uruchamiamy konsolę tekstową i wpisujemy komendę:

```
/usr/X11R6/bin/startxwin.bat
```

W wyniku powinien uruchomić się X-serwer z prostym window managerem i pojedynczym shellem. Kłopoty z uruchomieniem się X-serwera w tym momencie mogą wynikać z nieprawidłowego zainstalowania (lub braku) pakietów systemu Xorg. Żeby poprawić ten problem, należy jeszcze raz uruchomić program **setup.exe** i włączyć pakiety z listy **X11**. *Uwaga! Nie można w ten sposób uruchomić X-serwera, gdy mamy zainstalowane zupełnie minimalne środowisko Cygwin (np. bez lokalnych fontów, bez window managera). Taka instalacja jest przez nas jednak niezalecana.*

*Uwaga (2)! Czasami zdarza się, że X-serwer nie chce się uruchomić zwracając komunikat „Brak miejsca na środowisko” (lub w wersji angielskiej „Out of environment space”). Należy wówczas przydzielić w systemie więcej miejsca na środowisko w programach konsolowych. Szczegółową instrukcję można znaleźć na stronach Microsoftu:*

<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80\support/kb/articles/Q230/2/05.ASP&NoWebContent=1>  
(<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80\support/kb/articles/Q230/2/05.ASP&NoWebContent=1> ). Po wprowadzeniu zmian komunikat powinien przestać się pojawiać.

Jeżeli X-serwer uruchamia się poprawnie, możemy spróbować połączyć się ręcznie z jakąś aplikacją iksową do naszego serwera (wykorzystując metodę „putty”, opisaną w rozdziale Sekcja 3.3.3.4). Jeżeli aplikacja się nie uruchamia (jej interfejs nie jest wyświetlany na serwerze), to może oznaczać, że lokalny system Windows ma zainstalowanego firewalla i nie wpuszcza żadnej aplikacji na porty X-serwera (od 6000 w górę).

Jeżeli aplikacja się uruchamia i jest wyświetlana, może to oznaczać, że to serwer z kolei ma zainstalowanego firewalla i nie wpuszcza do siebie pakietów XDMCP. W tym przypadku należy skontaktować się z administratorem serwera.

X-serwer możemy zabić na 2 sposoby:

- przez naciśnięcie Alt-F4 (standardowy sposób zamykania programów w Windowsach). Pojawi nam się wówczas okno dialogowe, gdzie powinniśmy potwierdzić nasz zamiar;
- poprzez naciśnięcie małego znaczka X na pasku zadań (znaczek ten pojawia się na czas uruchomienia X-serwera). W tym celu musimy najpierw jakoś „przejsć” ze środowiska X-Windows do Windowsów (np. naciskając CTRL-ESC, Alt-ESC albo Alt-Tab).

Stronę serwera (Linux) opisuje Sekcja 3.3.3.5.

### 3.3.3.3. Instalacja X-serwera Pexus X-Deep/32

Jeszcze inną możliwością, jeżeli chodzi o darmowe X-serwery pod Windows, jest aplikacja X-Deep/32. Był to produkt komercyjny, rozwijany przez firmę Pexus. W sierpniu 2004 roku firma przestała istnieć, a prawa autorskie do programu zostały przekazane jego autorowi, który opublikował go na licencji freeware (nie OpenSource !), co oznacza, że można go wykorzystywać za darmo, ale nie można np. modyfikować.

X-Deep ma małe wymaganie sprzętowe i programowe, według autorów wystarczy procesor 486DX i dowolna wersja Windows od 95 włącznie (choć w przypadku Windows 95 może być konieczne zainstalowanie dodatkowych bibliotek). Zaletą jest także mały rozmiar plików do ściągnięcia (9 MB). Niestety program prawdopodobnie nie będzie już rozwijany, więc nie ma co liczyć na poprawki czy nowe funkcje.

Instalację rozpoczynamy od ściągnięcia programu ze strony [www.pexus.com](http://www.pexus.com) (<http://www.pexus.com>) - wybieramy wersję *Basic, Single file download*. Po ściągnięciu pliku `xdp40Basic.exe` uruchamiamy go, klikamy na *Next* na pierwszym ekranie, na kolejnym zaznaczamy, że zgadzamy się na warunki licencji (*I agree with above terms and conditions*) i ponownie klikamy *Next*. Kolejnym krokiem jest wybór katalogu instalacyjnego - możemy zostawić proponowany `c:\xdeep32_40` i kliknąć *Start*. Po zakończeniu instalacji klikamy *Ok* i na następnym ekranie zostawiamy wybrane *Create shortcut(s) on Desktop* - utworzenie skrótów na ekranie, potwierdzamy *OK*.

Następnie albo z menu Start, albo korzystając z utworzonego skrótu na pulpicie, uruchamiamy program X-Server. Pojawi się po chwili okienko ze spisem dostępnych interfejsów sieciowych. Na razie możemy zostawić domyślny *ANY* i kliknąć *Select*. Po chwili zobaczymy kolejne okienko, z listą wykrytych w sieci serwerów XDM. Wybieramy serwer, z którym będziemy się łączyli i naciskamy *Connect*. Po chwili powinniśmy zobaczyć okno logowania na serwer.

Możemy także skonfigurować X-Deep/32 w taki sposób, aby automatycznie łączył się od razu z wybranym przez nas serwerem. W tym celu:

- Klikamy prawym przyciskiem myszy na belce tytułowej okna Windows z uruchomionym X-Serwerem i wybieramy z menu *X-Server Menu* a następnie *X-Server Options*.
- W okienku dialogowym, które się pojawi wybieramy zakładkę XDMCP. Zaznaczamy opcję *XDMCP by Query Host*, wpisujemy także adres IP serwera - *XDMCP Hostname or IP address*.
- W zakładce *Network Interfaces* wybieramy jako domyślny interfejs sieciowy sieć lokalną. Jeżeli nie wiemy, która to jest, możemy zostawić ANY. Odznaczamy obie opcje poniżej (*Never show...* i *Always prompt...*).
- W zakładce *Font*, jeżeli chcemy korzystać z font serwera (co jest najprostszą metodą uzyskania np. polskich znaków), klikamy na *Add Font Server* i wpisujemy adres IP serwera (port zostawiamy bez zmian 7100) i klikamy OK - oczywiście font serwer musi być uruchomiony na serwerze i musi nasłuchiwać na porcie TCP/IP (zobacz Sekcja 3.3.3.5).
- Aby uzyskać polską klawiaturę, należy ściągnąć plik *polish.map* (<http://www.szarp.com.pl/polish.map>) i skopiować go do katalogu `/usr/lib/X11/keymaps` (ścieżka podana od katalogu, w którym zainstalowaliśmy X-Deep/32). Następnie w zakładce *Input* klikamy *Select Keymap File* i wybieramy plik `polish.map`.
- W zakładce *Input* należy odznaczyć opcję *Keyboard autorepeat*. Oznacza to włączenie automatycznego powtarzania klawiszy, które powoduje problemy z obsługą niektórych programów przy pomocy klawiatury.
- Następnie w głównym oknie opcji klikamy OK. Na pytanie o restart programu możemy odpowiedzieć *Nie* i samemu zamknąć program, a następnie uruchomić ponownie, aby upewnić się, że jest poprawnie skonfigurowany. Najprawdopodobniej w wyniku błędu w programie pojawi się okno wyboru serwera XDMCP. Musimy wybrać nasz serwer, a następnie ponownie wybrać *X-Server Menu* -> *X-Server Options* -> *XDMCP* -> *XDMCP by Query Host*. Po kolejnym uruchomieniu okno z pytaniem o serwer nie powinno już się pojawiać.

Jeżeli dostaniemy komunikat o błędnym ustawieniu ścieżki do fontów (*FontPath*), oznacza to najprawdopodobniej problem w połączeniu się z serwerem fontów. Można wykasować serwer z początku ścieżki i uruchomić aplikację, aby móc sprawdzić i ewentualnie zmienić ustawienia.

#### 3.3.3.4. Tunelowanie połączeń X-owych

Po co tunelować? Są dwa główne powody:

- Domyślnie X-serwery z X-klientami komunikują się za pomocą nieszyfrowanych kanałów. Tunelowanie X-ów przez ssh umożliwia bezpieczną zdalną pracę przez Internet.
- Tunelowanie umożliwia łączenie się z X-serwera z X-klientami schowanymi tak za translatorami adresów (NAT-ami, Masquerade'ami), jak i za firewallami.

Umieszczono tu opis dwóch sposobów tunelowania połączeń X-owych:

- Z wykorzystaniem *putty*  
Wymagane oprogramowanie:

- Terminal: putty.exe w wersji jak najświeższej (dostępne z <ftp://sunsite.icm.edu.pl/site/putty-devel> (<ftp://sunsite.icm.edu.pl/site/putty-devel>)), jakiś XServer (WinaXe, Cygwin/Xorg, ...).
- Serwer: żadne dodatkowe oprogramowanie nie jest wymagane.

Uruchamianie systemu:

- Program putty nie wymaga instalacji i jest dystrybuowany w postaci jednego pliku wykonywalnego (`putty.exe`).
  - Uruchamiamy putty, ustawiamy odpowiedni adres serwera i wybieramy opcję używania protokołu „ssh”. Wybieramy okienko Connection/Ssh/Tunnels, gdzie ustawiamy opcję „X11 Forwarding”.
  - Logujemy się na serwer. Możemy sprawdzić, że putty odpowiednio ustawił zmienną środowiskową DISPLAY.
  - Po stronie Terminala uruchamiamy Xserver - w przypadku WinAxe/XWinPro nie jest wymagane ustawianie dodatkowych parametrów (Xsettings) w stosunku do ustawień domyślnych.
  - Po stronie serwera możemy już uruchamiać programy X-owe (np. xterm). Teoretycznie możemy uruchomić również jakiegoś Desktop/Window Managera .
- Przy użyciu WinaXe v6.2 i LbxLoxy

Wymagane oprogramowanie:

- Terminal: WinaXe v6.2.
- Serwer: lbxloxy, dystrybuowany w formie gotowego programu wykonywalnego (1 plik). Nie używa plików konfiguracyjnych.

Uruchamianie systemu:

- Na Terminalu instalujemy standardowy WinaXe (nie jest wymagana zmiana ustawień domyślnych w XSettings).
- Po stronie Serwera instalujemy lbxloxy (np. rozpakowujemy zipa i kopiujemy plik `lbxloxy` do odpowiedniego katalogu, np. `/usr/bin`).
- Uruchamiamy lbxloxy:
 

```
lbxloxy -waitXServer
```

 Wywołany w ten sposób lbxloxy skorzysta z domyślnego portu 6163 (`DISPLAY=:63`).
- Uruchamiamy WinaXe i klikamy prawym przyciskiem myszy w symbol WinaXe na pasku zadań. Wybieramy Options/Connect LbxLoxyServer. Pojawi się okienko, w którym ustawiamy adres hosta, na którym uruchomiliśmy lbxloxy oraz numer portu (6163). Ustawienia zatwierdzamy ENTER-em, bo przycisk OK źle działa.
- Klikamy znowu prawym przyciskiem myszy na WinaXe na pasku zadań i sprawdzamy w Messages czy udało nam się połączyć z lbxloxy (ostatnią wiadomością powinien być napis:
 

```
ConnectLbxLoxy/EstablishNewConnections(mail.praterm.com.pl,6163)=1 : OK).
```
- Po stronie serwera ustawiamy odpowiednio zmienną środowiskową DISPLAY, np.
 

```
export DISPLAY=:63
```

 Od tej pory możemy już uruchamiać programy X-owe na serwerze.

Oba rozwiązania (putty + WinaXe) oraz (lxbloxy + winaXe) są identyczne pod względem funkcjonalności.

Metoda „putty” jest szybsza i prostsza w konfiguracji (nie robimy nic na serwerze), jednak wymagane jest wywoływanie komend z shella po stronie serwera.

Sposób „lxbloxy” wymaga skonfigurowania programu lxbloxy na serwerze i dbaniu o to, żeby był uruchamiany (czasami zdarza się, że lxbloxy sam „wychodzi” - wtedy należy go ponownie uruchomić). Wprawdzie użytkownik nie musi korzystać z shella, ale jest zmuszony do każdorazowego klikania prawym klawiszem myszy na pasku zadań i ręcznego ustawiania parametrów dla lxbloxy.

Jak dotąd nie udało mi się sprawić, żeby którykolwiek z powyższych sposobów zaczął współpracować z kdm/xdm, a bez tego nie jest możliwe przyjazne użytkownikowi korzystanie z X-ów.

### 3.3.3.5. Konfiguracja xdm/kdm

Ta sekcja opisuje sposób konfiguracji programu **xdm** i **kdm** w taki sposób, żeby umożliwiły zdalne korzystanie z systemu. Pierwszy opis (preferowany) dotyczy protokołu XDMCP. Drugi, mniej elastyczny, to połączenie bez wykorzystywania XDMCP.

- Program **kdm** pochodzi bezpośrednio od programu **xdm** (mają wspólny kod źródłowy). Używają się one tych samych plików konfiguracyjnych. Protokół XDMCP działa następująco: połączenie jest inicjowane z X-terminala do programu **xdm** (który nasłuchuje na porcie 177). Po zgłoszeniu się (ang. „query”), **xdm** próbuje się przyłączyć do hosta od którego przyszło głoszenie. Żeby skonfigurować **xdm/kdm** do połączeń za pomocą protokołu XDMCP, należy zmodyfikować pliki:

- W pliku `/etc/X11/xdm/Xaccess` ma być odkomentowana *tylko* linia:

```
* #any host can get a login window
```

Reszta, a w szczególności linia:

```
#* CHOOSER BROADCAST #any indirect host can get a chooser
```

powinna być zakomentowana.

- W systemach v6.2, v7.x, v8.x i nowszych należy w pliku `/etc/X11/xdm/xdm-config` wykomentować ostatnią linię, żeby wyglądała następująco:

```
! DisplayManager.requestPort: 0
```

(w tym pliku znakiem komentarza jest wykrzyknik - „!” , a nie hasz - „#”).

- (Jeżeli korzystamy z KDE 2.2/3.x, czyli na nowych systemach). Konfiguracja **kdm** znajduje się tu w katalogu `/etc/kde/kdm/` lub `/etc/kde3/kdm/`. Większość plików to odnośniki (linki) do ich odpowiedników w `/etc/X11/xdm/xdm-config`, tak więc modyfikować należy np. plik `/etc/kde3/kdm/Xaccess`. Plikiem charakterystycznym dla systemu **kdm** jest `/etc/kde/kdm/kdmrc`. W celu uruchomienia protokołu XDMCP (bez niego nie otrzymamy okna **xdm/kdm**), należy w sekcji `[Xdmcp]` ustawić parametr „Enable=true” (lub opcję tę usunąć zupełnie).

- Konfiguracja systemu **xdm/kdm** oraz X-serwera bez użycia XDMCP. W tym przypadku to nie X-terminal rozpoczyna sesję lecz **xdm**. Dlatego zwykle trudno znaleźć dla tego przypadku sensowne zastosowanie. Bo skąd ma **xdm** wiedzieć kiedy rozpocząć sesję?

Żeby skonfigurować system w ten sposób, należy w pliku `/etc/X11/xdm/Xservers` wstawić dla każdego terminala następuną linię:

```
<ip_terminala>:0 foreign,
```

a po stronie X-terminala (wyłączyć) opcję wykorzystania protokołu XDMCP. W przypadku systemu WinaXe polega to na odznaczeniu zaznaczonej uprzednio opcji „XDMCP”, a w systemie Cygwin/Xorg na nie korzystaniu z opcji „-query”. Po stronie serwera można w tym przypadku wykomentować wszystkie wpisy w `/etc/X11/xdm/Xaccess`.

Jeżeli chcemy umożliwić zdalnym klientom korzystanie z fontów z serwera (co zwykle jest opcją najprostszą, choć zwiększa obciążenie sieci), to musimy zadbać, aby font server był uruchamiany przy starcie systemu (w sposób odpowiedni dla danej dystrybucji, zwykle przez skrypt z katalogu `/etc/init.d/`). Dodatkowo font server musi nasłuchiwać na porcie TCP/IP, co nie zawsze jest opcją domyślną. Należy się więc upewnić, że w pliku konfiguracyjnym, zwykle `/etc/X11/fs/config`, nie ma wpisu `no-listen tcp`, bądź też jest wykomentowany. Zmiana opcji w tym pliku wymaga przestartowania serwera fontów. Domyślny port, na którym nasłuchuje serwer ma numer 7100.

**Notatka:** W pakietach xfs Debiana z serii X-org 7.0 pliki z katalogu `/etc/X11/fs` istnieją, ale są ignorowane - odpowiednie wpisy należy zrobić w plikach z katalogu `/usr/lib/X11/fs`. Błąd ten powinien zostać poprawiony w przyszłych wersjach pakietu.

### 3.3.3.6. Najczęstsze kłopoty z X-terminalami

Czasami nasz X-terminal nie może się przyłączyć do serwera, czyli nie wyświetlane jest zgłoszenie systemu **xdm** (lub pokrewnego). W takich przypadkach warto jest sprawdzić kilka rzeczy, po kolei:

- Serwer. Czy inne X-terminale mogą się przyłączyć do serwera? Jeśli nie to czy po stronie serwera nie ma uruchomionego firewalla? Jeżeli tak to jakie porty są zablokowane/otwarte? Czy jest odblokowany XDMCP-owy port 177 (zwykle UDP, ale także TCP)? A może zablokowany jest cały interfejs?
- X-terminal. Należy pamiętać, że architektura X-Window System wymaga możliwości „dobicia się” do X-terminala. Czy X-terminal nie ma firewalla, który blokuje połączenia? Czy X-terminal nie jest schowany za NAT-em (w Linuksie: IP masquerading)?

Jeśli problemy z systemem **xdm/kdm** nie są spowodowane żadnym z powyższych błędów, można spróbować debuggowania. Opcje do programu **xdm** są opisane w manualu (**man xdm**), natomiast **kdm** przyjmuje dwie opcje debuggujące: „-debug <debug\_level>” i „-logfile <plik>”. Obie opcje muszą być przekazane *przed* innymi opcjami. Argumentem do „-debug” jest wartość dziesiętna 1, 2, 4, 8, 16, ..., 512, oznaczająca odpowiednią część systemu. Jeśli chcemy debuggować więcej niż jedną rzecz równocześnie, to sumujemy wartości (np. 1+2+16), rzecz analogiczna jak w przypadku sumy logicznej w przypadku ustawiania flag. Argumentem drugiego parametru (czyli „-logfile”) jest nazwa pliku, do którego będą zrzucane wszystkie komunikaty.

## 3.3.4. Pomoc zdalna dla Windows

### 3.3.4.1. Wstęp

W sytuacji, gdy użytkownik znajduje się w oddalonej lokalizacji a kontakt telefoniczny nie doprowadził do rozwiązania jego problemu, rozwiązaniem może być użycie systemu VNC. Pozwala on na zdalną pracę na systemie Windows. Typowe aplikacje VNC wymagają jednak instalacji, która dla niedoświadczonego użytkownika może stanowić problem. Z tego powodu powstał system *UltraVNC*

*Single Click* dostępny na stronie <http://ultravnc.sourceforge.net/addons/singleclick.html>. Umożliwia on stworzenie programu nie wymagającego instalacji, który po uruchomieniu połączy się z wybranym przez użytkownika serwerem pomocy technicznej i udostępni jego pulpit. Poniżej przedstawiony został przykład jego wykorzystania.

### 3.3.4.2. Przygotowanie programu dla użytkowników

Opis przygotowany został na podstawie dokumentacji ze strony:  
<http://ultravnc.sourceforge.net/addons/singleclick.html>

Pierwszym krokiem jest pobranie i rozpakowanie archiwum `custom.zip` spod adresu:  
<http://www.uvnc.com/custom.zip>

Następnie należy wyedytować plik `helpdesk.txt`. Serwery dodaje się wpisując sekcję *HOST*.  
 Przykładowa sekcja:

```
[HOST]
Serwer pomocy technicznej
-connect 192.168.1.1:5500 -noregistry
```

Pierwsza linia będzie nazwą serwera wyświetlaną w oknie wyboru. Druga zawiera opcje wywołania programu. Opcja *-noregistry* jest obowiązkowa a po opcji *-connect* należy podać adres i port serwera.

Następnie należy wyedytować sekcje odpowiedzialne za treść komunikatów wyświetlanych przez program. *BALLON1* odpowiada za komunikaty wyświetlane w trakcie nawiązywania połączenia, a *BALLON2* za komunikaty wyświetlone, kiedy połączenie zostanie nawiązane.

Sekcje *TEXT\** odpowiadają za dodatkowe komunikaty wyświetlane w oknie wyboru serwera. Ich funkcję ilustruje zrzut okna zamieszczony na stronie dokumentacji.

Do tak przygotowanego pliku konfiguracyjnego można dołączyć ikony i logo. Ich wymiary i nazwy plików opisane są w na stronie dokumentacji.

Kolejnym etapem przygotowywania programu jest stworzenie archiwum zip zawierającego wymienione wyżej pliki (wymagany jest tylko plik `helpdesk.txt`). Nazwa archiwum musi zmieniać się przy tworzeniu kolejnych wersji programu. W przeciwnym wypadku może dojść do pomieszania plików z różnych wersji. Archiwum należy wysłać przy pomocy formularza: <http://sc.uvnc.com/index.php?section=19>. Nazwa użytkownika i hasło wymagane przez formularz podane są na stronie dokumentacji. Po wysłaniu nastąpi przekierowanie na stronę z gotowym plikiem wykonywalnym. Plik ten należy zapisać i udostępnić użytkownikom.

### 3.3.4.3. Używanie UltraVNC SC

Do rozpoczęcia korzystania z pomocy zdalnej potrzebne jest jeszcze uruchomienie na serwerze *UltraVNC Viewer*. Można go pobrać ze strony: <http://www.uvnc.com/>. Nie udało się zastąpić go innymi programami (np. `krdc`), ale można używać go na Linuksie przy pomocy pakietu *wine* i wersji *Standalone*, która nie wymaga instalacji. Przykład wywołania:

```
wine vncviewer.exe -listen
```

## 4. Dodatki

### 4.1. Dokumentowanie kodu w C

#### 4.1.1. Idea

Doxygen służy do generowania dokumentacji kodu napisanego w C, C++ i Javie. Strona domowa projektu to <http://www.doxygen.org>. Są do ściągnięcia pakiety RPM na większość popularnych dystrybucji. W nowszych program jest już standardowo obecny.

Jak wszyscy wiemy, nie wszyscy piszą komentarze w kodzie, a później wszyscy tego żałują - zwłaszcza, że długość życia kodu np. w Pratermie jest dość duża. Doxygen poza tym, że pozwala generować ładną dokumentację, która pozwala się zorientować w strukturze kodu i służyć jako *reference* do funkcji i typów danych, wymusza także pewien styl komentowania, który na dłuższą metę okazuje się całkiem sensowny. Zapewnia także generację dokumentacji w różnych formatach - jako strony HTML, jako LaTeX, RTF lub strony podręcznika man. Obsługuje wiele języków, w tym język polski.

Aby uzyskać dokumentację kodu, należy po pierwsze opisać odpowiednio kod (za pomocą odpowiednio sformatowanych komentarzy), a po drugie uruchomić Doxygena z odpowiednim plikiem konfiguracyjnym. Oba te elementy są pokrótce opisane poniżej. Pełny opis wszystkich opcji można znaleźć w dokumentacji programu.

#### 4.1.2. Komentowanie kodu

Komentarze, które mają być przetwarzane przez Doxygena zaczynają się dwoma gwiazdkami lub trzema slashami:

```
/** Komentarz blokowy */
/// Komentarz jednolinijkowy
```

Komentarz umieszczamy *PRZED* opisywaną funkcją, strukturą danych itp. Pierwsze zdanie (do kropki), służy jako krótki ogólny opis. Potem możemy napisać trochę szczegółów (jeśli trzeba). A na koniec opisujemy wszystkie parametry funkcji, używając konstrukcji:

```
@param nazwa opis_parametru
```

oraz wartość zwracaną, przez konstrukcję:

```
@return opis
```

Ważne, żeby opisy parametrów zawierały także informacje o dopuszczalnych wartościach, ewentualnie reakcji na nietypowe wartości itp. Cały opis funkcji (metody), może wyglądać tak:

```
/**
 * Funkcja nic nie robi (opis ogólny). Ta funkcja naprawdę nic nie robi
 * (opis szczegółowy).
 * @param par1 naprawdę wszystko jedno, co podasz
 * @param par2 też nie ma znaczenia
 * @return zawsze -18
 * @see coś_robi
 */
int nic_nie_robi(int par1, char *par2)
```

```
{
    return -18;
}
```

Użyto tu też konstrukcji *@see*, powodującej wygenerowanie linku do dokumentacji funkcji *coś\_robi*.

Opis funkcji możemy podać zarówno w miejscu deklaracji, jak i definicji. Opisując typy (struktury danych) czasami wolelibyśmy umieszczać opisy pól struktury (klasy) po nich. Możemy robić to używając komentarza w postaci */\*\*< ... \*/*, w ten sposób:

```
/** Opis ogólny struktury ble. Opis szczegółowy. */
struct ble {
    int pole1;      /**< To jest pierwsze pole struktury ble */
    int pole2;      /**< A to jest drugie pole. */
};
```

W komentarzach można używać także innych dyrektyw postaci *@costam*, oraz większości tagów html-owych. Przedstawiona konwencja dokumentowania jest zgodna z Javadoc, można także używać innych (np. Qt).

### 4.1.3. Plik konfiguracyjny

Operacja generowania dokumentacji polega na wywołaniu programu doxygen z jedną opcją - nazwą pliku konfiguracyjnego.

Generowanie tego pliku można sobie ułatwić przez użycie programu doxywizard. Uruchamia się go pod X-ami. Umożliwia wygodne ustawienie wszystkich potrzebnych opcji za pomocą klikanego interfejsu. Najważniejsze jest podanie katalogu, w którym znajdują się pliki z kodem. Można kazać generować dokumentację dla wszystkich plików o podanej masce, lub o konkretnych nazwach. Ważne jest zaznaczenie opcji *EXTRACT\_ALL*, dzięki czemu zostanie wygenerowany opis także tych funkcji, które nie są opisane komentarzami. Należy także zaznaczyć, jakich formatów danych wyjściowych się spodziewamy i w jakich katalogach mają być wygenerowane.

Po ustawieniu opcji każemy zapisać plik konfiguracyjny na dysku, a potem możemy go używać jako parametru dla programu Doxygen. Uwaga! W niektórych wersjach Doxygena (np. 1.2.8.1) program doxywizard miał błąd, objawiający się niezapisywaniem do wyjściowego pliku prawidłowej wartości opcji *EXTRACT\_ALL*. Konieczna jest wtedy ręczna edycja pliku konfiguracyjnego i ustawienie jej wartości na *YES*.

## 4.2. Przeszukiwanie dokumentacji - system ht://Dig

**ht://Dig** jest systemem używanym do indeksowania i wyszukiwania informacji na stronach WWW (na danym serwerze, w domenie lub intranecie). Interfejs użytkownika przypomina popularne wyszukiwarki, takie jak Yahoo!, Google czy Infoseek: użytkownik ma możliwość wpisania słów, które następnie będą wyszukiwane (po naciśnięciu przycisku „Znajdź”). Zwracana jest pełna lista trafień.

System ht://Dig składa się z zestawu komend oraz bazy słów kluczowych. Komendy, m.in. **htdig**, **htsearch**, **htstat**, **htmerge**, służą do konfiguracji i monitorowania pracy systemu oraz spełniają funkcje skryptów CGI, uruchamianych prosto ze strony WWW. Te, które nie są skryptami CGI (np. **htdig**), wywołuje się z poziomu shella. Baza słów kluczowych jest z kolei zbiorem wszystkich słów w przeszukiwanej domenie wraz z miejscami ich wystąpienia. Przeszukiwanie bazy jest znacznie wydajniejsze niż byłoby każdorazowe przeszukiwanie całej dokumentacji.

Pakiety instalacyjne systemu ht://Dig są standardowo rozpowszechniane z każdą dystrybucją systemu RedHat. Najświeższa wersja systemu jest dostępna z <http://www.htdig.org>.

Plikiem konfiguracyjnym dla systemu ht://Dig jest `/etc/htdig.conf`. Wersja tego pliku, która jest dołączona do systemu (domyślna), może być wykorzystana do konfiguracji systemu bez żadnych zmian. Należy jednak sprawdzić, czy wartość zmiennej „start\_url” w pliku to „http://localhost” i czy wartość „localhost” jest odpowiednio ustawione w `/etc/hosts` (z reguły wartość 127.0.0.1). Należy również sprawdzić czy działa demon httpd, np. poprzez przyłączenie się do serwera WWW: **lynx http://localhost**. Wyświetlona strona powinna być stroną główną przeszukiwanej dokumentacji, wraz z odnośnikami do wszystkich poddokumentów. Naturalnie, adresy podane jak wyżej mogą być zmienione, w naszym przypadku zamiast `http://localhost` ustawiono `http://www.praterm.com.pl/howto`.

Gdy serwer WWW działa tak, jak to opisano powyżej, należy wytworzyć bazę danych. Służy do tego polecenie: **rundig** (bez parametrów). Czas trwania procesu tworzenia bazy jest uzależniony od ilości przeszukiwanych informacji. W przypadku niniejszego HOWTO i dokumentacji do systemu SZARP całość trwa kilka sekund. Jeżeli informacje na stronie często się zmieniają, to należy ten proces przeprowadzać wielokrotnie, na przykład po każdej zmianie w dokumentacji, najlepiej automatycznie.

Kolejnym krokiem jest odpowiednia modyfikacja interfejsu ht://Dig. Wprawdzie całość interfejsu jest dostarczana wraz z systemem (znajduje się w katalogu `/usr/share/htdig`, wskazywanym domyślnie przez `/var/www/html/htdig`) i można się do niej odnieść poprzez adres `http://localhost/htdig`, lecz jest to interfejs przykładowy, który należy dopiero zintegrować z własną stroną. Poza tym jest to wersja angielska, więc w przypadku polskich stron powinien być przetłumaczony na polski. Interfejs składa się z kilku plików `html` o określonych nazwach, które są wyświetlane w odpowiednich momentach (np. gdy nie znaleziono żadnego wystąpienia danego słowa, wyświetlana jest zawartość pliku `nomatch.html`). W opisywanym tu systemie wyszukiwawczym dla dokumentacji SZARP-a skopiowano zawartość katalogu `/usr/share/htdig` i zmodyfikowano odpowiednie pliki. Następnie usunięto istniejący link `/var/www/html/htdig` i stworzono nowy, wskazujący na zmodyfikowany katalog z plikami ht://Dig, tak jak poniżej:

```
rm /var/www/html/htdig
ln -s /opt/szarp/resources/documentation/htdig /var/www/html/htdig
```

Strona, w której wywoływane jest przeszukiwanie domeny, została zintegrowana ze spisem treści dokumentacji SZARP-a. Od tej pory system ht://Dig powinien być w pełni sprawny.

## 4.3. Tips & Tricks

### 4.3.1. Problemy z DCOPServer

Jeżeli w KDE 3.X po pewnym czasie pracy pojawiają się komunikaty o problemach w komunikacji międzyprocesowej KDE i przestaje działać panel, należy skasować całą zawartość katalogu `/tmp` oraz pliki zaczynające się od `.DCOPserver` w katalogach domowych użytkowników.

### 4.3.2. Narzędzie xsri

Zapewne wielu z Was zastanawiało się nad sposobem usunięcia denerwującego logo RedHata z tła KDM-a. Otóż logo to jest ustawiane za pomocą programu **xsri**. Program ten jest wywoływany automatycznie przy starcie menedżerów logowania KDM/XDM z pliku: `/etc/X11/xdm/Xsetup_0`.

Żeby logo się nie pojawiało, wystarczy wycommentować wywołanie programu **xsri**, tzn. wszystko zaczynające się od linii:

```
if [ -x /usr/bin/xsri -a -f /etc/X11/xsrirc ]; then
```

Program **xsri** można wykorzystać także w innym celu: żeby ustawić własne logo, np. logo Pratermu. Można przy tym posłużyć się opcją `--tile` (która ustawia sąsiadująco wiele kopii obrazka):

```
/usr/bin/xsri --tile= \  
/opt/szarp/resources/documentation/new/common/praterm-logo.png
```

RedHat wykorzystuje plik konfiguracyjny `/etc/X11/xsrirc`, w którym jest zdefiniowany specjalny alias. W przypadku własnego logo można zrobić to samo.

### 4.3.3. Robienie zrzutów ekranu (*screenshotów*) pod Linuksem

Do robienia zrzutów okienek (lub całego ekranu) można pod Linuksem wykorzystać jeden z wielu programów. Tu zostanie opisany program **display**, należący do pakietu ImageMagick. Główną funkcją tego programu jest wyświetlanie plików graficznych (obsługuje stosunkowo dużo formatów graficznych), ich modyfikacja (w ograniczonym zakresie) oraz właśnie robienie zrzutów okien. W tym celu należy:

- Uruchomić program **display** i kliknąć lewym klawiszem na wyświetlonym okienku, żeby pojawiło się menu główne programu.
- Z menu wybrać opcję File, a następnie Open. Pojawi się okienko wyboru pliku „Browse and Select a File”.
- W okienku nie wybieramy żadnego pliku, lecz klikamy na klawisz **Grab** (żądanie zrzutu ekranu). Pojawi się okienko „Enter any delay in seconds”, wymagające wpisania ilości sekund do uaktywnienia trybu zrzutów ekranu.
- W okienku należy wpisać kilka sekund (np. 5) i potwierdzić klawiszem **Grab**.
- Teraz mamy kilka sekund (np. 5 ;) na otwarcie okienka, które chcemy zrzucić (lub przełączenie się na inny pulpit). Po tym czasie nasz kursor zamieni się w krzyżyk, którym należy kliknąć w interesujące nas okienko.
- Okienko to zostanie przekopiowane do programu **display** i wyświetlone (przez chwilę może się wydawać, że otworzono drugie, identyczne okienko, lecz to tylko kopia w programie **display**).
- Teraz wybieramy opcję **Save**, żeby zapisać obrazek w odpowiednim formacie (np. png). Można to zrobić półautomatycznie (poprzez napisanie nazwy pliku wraz z rozszerzeniem). Przed zapisaniem pliku możemy go oczywiście jeszcze przetworzyć.

## 5. Historia

### 5.1. Historia

**Notatka:** Historia dokumentu obecnie nie jest uaktualniana, do śledzenia zmian w dokumencie należy posługiwać się CVS'em lub głównym plikiem ChangeLog systemu SZARP. Ze względu na duże zmiany w strukturze dokumentu, usunięte zostały dowiązania w historii do konkretnych rozdziałów.

- v 1.103, 2004.11.10 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Konfiguracja Neostrady+ przy pomocy modemu ADSL Sagem F@st 800 (USB) pod Debianem (...).
- v 1.102, 2004.11.02 Ecto (mailto:ecto@praterm.com.pl)  
Konfiguracja ISDN pod Debianem (...). Nie jest to Da Debian Proper Way, ale tak robimy ...
- v 1.101, 2004.08.31 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Oddzielny rozdział o NAT/masqueradzie (...), większość treści była już wcześniej. Teraz z odpowiednich miejsc są linki do rozdziału o NAT.
- v 1.100, 2004.08.27 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodanie rozdziału od usłudze TPSA "Internet DSL" (...), dopisanie instrukcji jak należy robić aliasy IP w Debianie (...).
- v 1.99, 2004.05.20 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Napisanie paru dodatkowych słów o PPP na początku rozdziału (...).
- v 1.98, 2004.05.19 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Konfiguracja Sferii (...).
- v 1.97, 2004.04.15 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodanie sekcji o GRUB-ie: ....
- v 1.96, 2003.10.19 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Poprawki i uzupełnienia w rozdziale o RAIDzie: ....
- v 1.95, 2003.10.14 Paweł (mailto:pawel@praterm.com.pl)  
Króciutki rozdział o FreeDOS'ie: ....  
Poza tym w rozdziale szarp-redhat ostrzeżenia przed strasznymi babolami w redhatowym gcc-2.96.
- v 1.94, 2003.09.26 Paweł (mailto:pawel@praterm.com.pl)  
Dodanie rozdziału o sieciach WLAN: ..., opis instalacji karty OvisLink AirLive 1120PCI: ....
- v 1.93, 2003.09.24 Paweł (mailto:pawel@praterm.com.pl)  
Przejrzenie HOWTO pod kątem nieaktualnych ścieżek (stara konwencja /opt/szarp/config), uaktualnienia w rozdziałach o instalacji i konfiguracji SZARPA.
- v 1.92, 2003.07.01 Vooyeck (mailto:vooyeck@praterm.com.pl)

Jak stworzyć własną okrojona dystrybucję Cygwina? Odpowiedź tutaj.

- v 1.91, 2003.06.25 Paweł (mailto:pawel@praterm.com.pl)  
Poprawki w opisie konfiguracji firewalla (...), informacja o nieaktualności informacji w rozdziale o polskich literach na wydrukach.
- v 1.90, 2003.06.25 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Wyodrębniony rozdział (...).
- v 1.89, 2003.06.20 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Wyodrębniony rozdział o programie **x<sub>dm</sub>** (...). Dodanie informacji jak konfigurować xfree86/cygwin, żeby działało pod Windows 9x/Me (działa!).
- v 1.88, 2003.06.18 Paweł Pałucha (mailto:pawel@praterm.com.pl)  
Uzupełnienia w opisie uzyskiwania polskich liter na wydrukach z programu przeglądającego.
- v 1.87, 2003.06.04 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis instalacji i konfiguracji systemów RedHat 9.0/Aurox 9.0 (...). Poprawki w rozdziałach o konf. drukowania.
- v 1.86, 2003.05.13 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Jak się łączyć z Internetem przez GPRS? ....
- v 1.85, 2003.05.09 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Uzupełnienie informacji o Cygwinie (instalacja, użytkowanie - ...). „Stawianie” sshd na Cygwinie, jak to zrobić: ....
- v 1.84, 2003.04.11 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Jak zainstalować Debiana, żeby SZARP na tym działał (...).
- v 1.83, 2003.03.31 Paweł (mailto:pawel@praterm.com.pl)  
Uaktualnienia w sekcji o sprzęcie, instalacja i konfiguracja modemu wewnętrznego PCI Pentagram Hex2 56I (...).
- v 1.82, 2003.03.31 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis programu redhat-config-network, obecnego w RH 7.x (kliknij tutaj).
- v 1.81, 2003.03.22 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodanie informacji o aliasach IP (...).
- v 1.80, 2003.03.08 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Ogólny rozdział o Cygwinie (...), Wrzucenie informacji o WinaXe i XFree86 do wspólnego rozdziału o X-Terminalach (...). Poprawki dot. serwera XFree86 (...).
- v 1.79, 2003.03.02 Paweł (mailto:pawel@praterm.com.pl)  
Poprawki w rozdziale o lm-sensors (co zrobić jeśli moduły nie zostały prawidłowo przekopiowane) - ... i uwagi o instalacji oprogramowania LanSafe do UPS-ów Powerware 5115 w wersji 4.17a dostarczonej na płycie CD z UPS-em - ....
- v 1.78, 2003.02.18 Paweł (mailto:pawel@praterm.com.pl)  
Przepisany na nowo rozdział o lm-sensors. Ponieważ w wersji 2.7.0 sporo się zmieniło, trudno byłoby zapanować nad bałaganem w tym rozdziale. Wobec tego zostawiłem właściwie tylko opis instalacji

lm-sensors 2.7.0 ze źródeł a dodałem opis instalacji i konfiguracji demona *mondo*, którego można użyć do kontrolowania stanu czujników i podejmowania odpowiednich akcji (...).

- v 1.77, 2003.02.17 Paweł (mailto:pawel@praterm.com.pl)  
Uniwersalny sposób konfigurowania GhostScripta, żeby na wydrukach były polskie litery - działa w LPD i w CUPS.
- v 1.76, 2003.01.09 Paweł (mailto:pawel@praterm.com.pl)  
Uwagi o konfigurowaniu przez CUPS drukarki podłączonej do Windows XP, rozdział o problemach z plikami kdelnk, rozdział o konfiguracji WindowMakera (Sekcja 3.2.20).
- v1.75, 2002.12.2 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Małe rozdziałki o: programiku xsri (...), o naprawianiu terminala jak się skopsa (...), i o screenshotach (...).
- V 1.74, 2002.11.30 Paweł (mailto:pawel@praterm.com.pl)  
Uaktualnienie sekcji o sprzęcie (...).  
Dodano opis drukowania z użyciem systemu CUPS.  
Nowy rozdział "Tips & Tricks" na drobne rzeczy, które nigdzie nie pasują (...).  
Opis rozwiązania problemu z DCOPServer w KDE (...).
- v1.73, 2002.11.09 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodano informację o zdalnym wywoływaniu programów iksowych przy pomocy ssh (...), dopisane na końcu).
- v1.72, 2002.10.31 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodany nagłówek z logo i informacjami o Pratermie SA. Identyczne nagłówki są w reszcie dokumentacji SZARP-a.
- v1.71, 2002.10.30 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Porządki w HOWTO: UPS-y do jednego rozdziału (...), powstał rozdział o protokołach p-t-p (...), tamże przeniósł się rozdział o modemie GSM Siemensa. Dodatkowo zmieniona została kolejność niektórych rozdziałów (mniejszy bałagan), np. najpierw konfiguracja sieci, potem ssh, przekierowywanie portów, a na końcu SMTP.
- v1.70, 2002.10.28 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodanie opisu konfiguracji drukarki lokalnej na Linuksie oraz opisu zdalnego drukowania pod Windowsami (...). Przeniesienie sekcji „Konfiguracja serwera drukowania poprzez SMB” z rozdziału o SZARP-ie do rozdziału o terminalu z MS Windows, delikatna zmiana nazwy (...).  
Rozwiązanie problemu z kdm-em pod RH 7.x (...), pod sam koniec).
- v1.69, 2002.09.24 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Jak skonfigurować drukowanie pod RH7.x.
- v1.68, 2002.08.28 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Uzupełnienie informacji o KickStarcie (...), usunięcie zawartości pliku `ks.cfg` dla konfiguracji minimalnej. Parę drobnych poprawek.
- v1.67, 2002.08.24 Vooyeck (mailto:vooyeck@praterm.com.pl)  
System indeksowania i wyszukiwania informacji `ht://Dig` (...).
- v1.66, 2002.08.07 Ecto (mailto:ecto@praterm.com.pl)

- Konfiguracja UPS-a EVER (...), autologowanie do KDE (...).
- v1.66, 2002.08.07 Lucek (mailto:lucek@praterm.com.pl)  
Pinologia kabla do PLIPa (...). Obudowa pyłoszczelna Advantech (...).
- v1.65, 2002.08.07 Ecto (mailto:ecto@praterm.com.pl)  
Konfiguracja PLIPa (...). Nowa płyta główna (...).
- v1.64, 2002.07.26 Lucek (mailto:lucek@praterm.com.pl)  
Rozdział o instalacji CDI (...).
- v1.63, 2002.07.08 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodanie opisu mechanizmu RedHat KickStart (...), automatyzującego instalację RedHata.
- v1.62, 2002.06.25 Paweł (mailto:pawel@praterm.com.pl)  
Zmiany w WinaXe 6.2 i późniejszych (...).
- v1.61, 2002.05.24 Paweł (mailto:pawel@praterm.com.pl)  
Poprawki dotyczące instalacji Linuksa na RH 7.3, uwagi o karcie S3 Paradise i downgrade'u X-ów z wersji 4.\* do 3.3.6 (...).
- v1.60, 2002.05.11 Ecto (mailto:ecto@praterm.com.pl)  
Udostępnianie połączenia innym użytkownikom sieci lokalnej przy serwerze SDI w Red Hat v7.2 (...).
- v1.59, 2002.05.05 Ecto (mailto:ecto@praterm.com.pl)  
APM i ACPI, czyli zamykanie systemu w komputerach z nowymi płytami głównymi, nowe karty graficzne (...).
- v1.58, 2002.05.03 Lucek (mailto:lucek@praterm.com.pl)  
lm-sensors (szczególnie wersja 2.6.3) a Red Hat v7.2 (...).
- v1.57, 2002.03.21 Lucek (mailto:lucek@praterm.com.pl)  
Dodanie (na podstawie doświadczalnej) opcji 'lba32' do /etc/lilo.conf (...).
- v1.56, 2002.03.16 Paweł (mailto:pawel@praterm.com.pl)  
Uaktualnienie dla instalacji na Red Hat v7.2.
- v1.55, 2002.03.16 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis rsyncowego backupu na terminal windowsowy przy pomocy Cygwin (...).
- v1.54, 2002.02.22 Lucek (mailto:lucek@praterm.com.pl)  
Namieszane w rozdziale ..., dodanie podrozdziału o polskich fontach dla większej czytelności.
- v1.53, 2002.02.22 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Wreszcie uporządkowane zostało trochę drukowanie: powstał duży rozdział o konfiguracji zdalnego drukowania pod Linuksem po stronie klienckiej jak i po stronie serwera (...), tamże - dodano o konfigurowaniu drukarek poprzez przekopiowanie odpowiednich plików, odchudzono rozdział w sekcji SZARP (...).
- v1.52, 2002.02.19 Paweł (mailto:pawel@praterm.com.pl)  
Opis netconf'a (...).
- v1.51, 2002.02.19 Paweł (mailto:pawel@praterm.com.pl)

- Opis konfiguracji i używania systemu do dokumentacji kodu Doxygen (...).
- v1.50, 2002.01.19 Lucek (mailto:lucek@praterm.com.pl)  
Wywalenie rozdziału o terminalu z wykorzystaniem netbasd i o terminalu mieszanym - rzeczach nigdy nie wykorzystywanych.
  - v1.49, 2002.01.19 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Rozdział o tym jak można tunelować ikisy (...).
  - v1.48, 2001.12.20 Ecto (mailto:ecto@praterm.com.pl)  
Rozdziały: o forwardowaniu portów (...) oraz o konfiguracji sendmaila jako mailera dla sieci lokalnej (...).
  - v1.47, 2001.12.14 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Cała masa poprawek w HOWTO, wśród nich: dodanie opisu instalacji pakietów openssh (...), rozwiązywanie problemów z klawiaturą pod Win 98 SE (...), dodanie sekcji o BODASie *pull*.
  - v1.46, 2001.10.28 Ecto (mailto:ecto@praterm.com.pl)  
Małe zmiany - sw\_cursor dla S3 Trio, oraz opcja persist dla SDI.
  - v1.45, 2001.09.13 Ecto (mailto:ecto@praterm.com.pl)  
Dodanie informacji o tworzeniu tunelików i skrypcie do tego ...
  - v1.44, 2001.09.10 Ecto (mailto:ecto@praterm.com.pl) & Vooyeck (mailto:vooyeck@praterm.com.pl)  
Rozdział o instalacji SDI ...
  - v1.43, 2001.08.27 Ecto (mailto:ecto@praterm.com.pl)  
Zmiany i uzupełnienia:
    - Dodanie do wymagań SZARPa tclreadline wraz z metodą instalacji.
    - Nowy rozdział - instalacja nowego kernela.
    - Korekta ortograficzna całości.
    - Interferencje minicom'a i mgetty (...)
  - v1.42, 2001.08.28 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis konfiguracji serwera + terminala na Linuksie w oparciu o rsync BODASA i netpar(d)a (...).
  - v1.41, 2001.08.28 Paweł (mailto:pawel@praterm.com.pl)  
Opis lokalizacji i sposobu instalacji plików w /etc/skel dających nowo tworzonym użytkownikom polskie literki. ....
  - v1.40, 2001.08.28 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Modyfikacje rozdziału o DocBooku - teraz na pewno zadziała (pod RH 6.2); poza tym dodano ekspresową konfigurację.
  - v1.39, 2001.08.27 Ecto (mailto:ecto@praterm.com.pl)  
Zmiany i uzupełnienia:
    - Nowe płyty główne ...
    - Uzupełnienia konfiguracji BIOS'u ...

- Uzupełnienia (drobne) w konfiguracji startowej Linuksa ...
- v1.38, 2001.08.18 Lucek (mailto:lucek@praterm.com.pl)  
Zmiany i uzupełnienia w liście potrzebnych pakietów.
- v1.37, 2001.08.17 Lucek (mailto:lucek@praterm.com.pl)  
Drobne modyfikacje opisu instalacji UPS Powerware w przypadku LanSafeIII w wersji 4.16 ....
- v1.36, 2001.08.16 Paweł (mailto:pawel@praterm.com.pl)  
Krótki rozdziałik o tym, do czego służy katalog `/etc/skeleton` (...).
- v1.35, 2001.08.14 Lucek (mailto:lucek@praterm.com.pl)  
Opis karty Moxa C168H/PCI (...), opis ustawiania nazwy komputera ..., dodanie nowych obsługiwanych płyt głównych ..., kart graficznych ..., kart sieciowych i HUBów ..., uwagi dotyczące dysków ATA/100 ..., uwaga czego nie da się wykonać przy pomocy anakondy ..., poprawki i uzupełnienia do konfiguracji `lm_sensors` ..., dodanie opisu instalacji i konfiguracji `vim-5.8`.
- v1.34, 2001.08.08 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Jeszcze o ssh (...), dodanie opisu konfiguracji BODAS-ów.
- v1.33, 2001.07.24 code (mailto:code@praterm.com.pl)  
Cygwin - opis
- v1.32, 2001.07.21 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Uzupełnienie rozdziału Pawła o konfigurowaniu ssh (...) o opis różnych wersji protokołu. Jak z tym walczyć.
- v1.31, 2001.07.06 Lucek (mailto:lucek@praterm.com.pl)  
Uzupełnienia związane z nową wersją emulatora xserwera Labtama ....
- v1.30, 2001.06.16 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Rozdział "Ciekawostki" rozpadł się już zupełnie, powstały rozdziały: ... i ....  
Opisany został sposób tworzenia własnej dystrybucji RedHata, czyli RedHattie (...).  
Wyglądzenie rozdziału o konfiguracji samby do drukowania.
- v1.29, 2001.06.15 Maciek (mailto:code@praterm.com.pl)  
Nowy rozdział Praterm HUB (...) dotyczący serwera WWW Pratermu.
- v1.28, 2001.05.31 Paweł (mailto:pawel@praterm.com.pl)  
Opis synchronizacji czasu (...).
- v1.27, 2001.05.04 Lucek (mailto:lucek@praterm.com.pl)  
Drobne dodatki do rozdziału ....
- v1.26, 2001.04.27 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Rozbiłem rozdział ... na dwa, i tak powstał .... Rozbiłem także informacje o sambie na dwa oddzielne: na serwer i klienta (...).
- v1.25, 2001.04.09 Paweł (mailto:pawel@praterm.com.pl)  
Opis konfiguracji DocBook'a, coby działał.

- v1.24, 2001.04.06 Paweł (mailto:pawel@praterm.com.pl)  
Dużo drobnych poprawek po kolejnym SZARP install party.
- v1.23, 2001.03.19 Paweł (mailto:pawel@praterm.com.pl)  
Instalowanie oprogramowania do Fideltronika z rpm'a (...).
- v1.22, 2001.04.04 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Poprawiony rozdział o konfiguracji modemu i ppp (...), dorzucone informacje o konfiguracji uugetty (tamże).
- v1.21, 2001.03.19 Paweł (mailto:pawel@praterm.com.pl)  
Konfigurowanie ssh (...) i opis BODASa - koncepcja, konfigurowanie wariantu *push*.
- v1.20, 2001.03.10 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodam informacje o UPS-ie Powerware ... przerzuciłem info o APC i Fideltroniku do odrębnych rozdziałów (... i ...).
- v1.19, 2001.02.21 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Nowe rozdziały ze starą, ale poszerzoną zawartością przerzucone z „Ciekawostek”: (... i ...), Dodany rozdział o lm\_sensors (...), któremu jeszcze trochę brakuje do doskonałości.
- v1.18, 2001.02.14 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Trochę informacji (by Lucek) o konfiguracji sieci 100Mbit (...), Poprawienie, uzupełnienie wpisów z v1.17, szczególnie o terminalach wirtualnych (...).
- v1.17, 2001.02.12/13 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Podstawowe informacje HUB-ach (...), Trochę o X-ach, trochę o crontabie, trochę o anacondzie (...).
- v1.16, 2001.01.15 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Dodano opis rsynca (...)
- v1.15, 2001.01.04 Paweł Pałucha (mailto:pawel@praterm.com.pl)  
Drobne poprawki, zwłaszcza jeśli chodzi o ściąganie backup'u z terminala - teraz powinien to robić meaner.
- v1.15, 2001.01.02 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis drukowania w SzarpDraw.
- v1.14, 2000.11.25 Lucek (mailto:lucek@praterm.com.pl)  
kilkanaście zmian i wstawek: wdrożone instalacje (...), kcmdpms, X'y (...), alarmy w BIOSie (...), mgetty (...), inne moduły Specialix (...), nowe drivery do Specialixa (...), smb (...), backup bazy danych na terminalu z Windows (...).
- v1.13, 2000.11.24 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Opis klienta ppp opartego o skrypty (...), Intel 815, jak z tym walczyć (gdzieś w rozdziale ...).
- v1.12, 2000.10.28 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Weryfikacja HOWTO w warunkach bojowych. Delikatne poprawki - wszędzie.
- v1.11, 2000.10.16 Vooyeck (mailto:vooyeck@praterm.com.pl)  
Techniczne poprawki: Makefile dla sgmla (make html, make clean); wywalenie labeli, wstawienie xrefów i id gdzie tylko się da. Wywalenie buga z ampersandem (zamiast & w kodzie SGML należy używać &amp;!).

- v1.10, 2000.10.08 Maciek (mailto:code@praterm.com.pl)  
Raid, Moxa.
- v1.9, 2000.07.15 Lucek (mailto:lucek@praterm.com.pl)  
Kilkanaście wstawek (m.in. icewm, dosemu, lilo-colors, ux-1.0, nowe i już zrealizowane systemy i inne).
- v1.8, 2000.06.30 Maciek (mailto:code@praterm.com.pl)  
Dokumentacja nowego SzarpDraw z parametrami definiowalnymi.
- v1.7, 2000.05.01 Jarek (mailto:jojo@praterm.com.pl)  
Wersja SGML dla DocBook'a.
- v1.6, 2000.03.08 Maciek (mailto:code@praterm.com.pl)  
Wersja SGML, opis instalacji i konfiguracji SZARPa.
- v1.5, 2000.03.05 Jarek (mailto:jojo@praterm.com.pl)  
Dodany opis konfiguracji kppp.
- v1.4, 2000.03.03 Jarek (mailto:jojo@praterm.com.pl)  
Poprawienie opisu konfiguracji /etc/rc.d.
- v1.3, 2000.02.25 Jarek (mailto:jojo@praterm.com.pl)  
Dodanie opisu konfiguracji lilo, uporządkowanie spisu treści (Terminal).
- v1.2, 2000.02.13 Lucek (mailto:lucek@praterm.com.pl)  
Dodanie nowej płyty głównej (...), dodanie nowego modemu (...).
- v1.1, 2000.02.04 Jarek (mailto:jojo@praterm.com.pl)  
Zmiana na iso-8859-2, dodany opis instalacji modemu i ppp, poprawione formatowanie, cały plik umieszczony w cvs.
- v1.0, 2000.01.31 Lucek (mailto:lucek@praterm.com.pl)  
Założenie pliku.

## Przypisy

1. Nie oznacza to, że RedHat dostarcza jakiegoś wyspecjalizowanego kernela. Każde jądro, jeżeli nie zrozumie jakiegoś parametru, (w tym przypadku „ks=”), przekazuje go dalej - do inita. Wyspecjalizowany (ale i bardzo uproszczony) jest natomiast instalacyjny **init**, który po otrzymaniu parametru „ks=” uruchamia anacondę z odpowiednim parametrem.