

Maciej Klesiewicz

**Instalacja, konfiguracja
i administrowanie
serwera sieciowego**

KRAKÓW 2003

O PRAWACH AUTORSKICH

Niniejsza praca powstała w oparciu o dokumenty i opracowania, które są wymienione w Bibliografii na końcu tej pracy. Rozpowszechnianie tej pracy, możliwe jest tylko po uprzednim uzgodnieniu z autorem, z wyłączeniem sytuacji, w których naruszałoby to w jakikolwiek sposób prawa autorskie opracowań źródłowych. Zabronione jest również czerpanie jakichkolwiek korzyści majątkowych z tytułu rozpowszechniania niniejszej pracy.

Kontakt z autorem: dyplo@fraxinus.tk.krakow.pl

Maciej Klesiewicz

Chciałbym przekazać serdeczne podziękowania wielu osobom z kanału **#debian.pl** sieci IRCNET za cenne rady i wskazówki dotyczące różnych aspektów praktycznych omawianych w tej pracy zagadnień.

SPIS TREŚCI

1. CEL PRACY	5
2. ZAGADNIENIA WSTĘPNE.....	6
SYSTEMY UNIKSOWE	6
Systemy plików	6
Konto	7
Sesja.....	8
Powłoka	8
Procesy	9
Zarządzanie pamięcią	11
SIECI KOMPUTEROWE.....	12
Model warstwowy OSI	12
Model sieci TCP/IP	13
Protokoły sieciowe	13
Sposoby transmisji i adresowania.....	14
Zestaw TCP/IP.....	14
Komunikacja sieciowa.....	17
3. INSTALACJA SYSTEMU	20
PROCES INSTALACJI.....	21
System instalacyjny	21
Program dbootstrap.....	24
Początkowa konfiguracja systemu.....	31
ZARZĄDZANIE PAKIETAMI	34
Program dpkg	36
Program dselect	38
Program apt	43
4. KONFIGURACJA SYSTEMU	45
PROCES INIT	46
Podtrzymywanie zasilania (UPS)	48
Drzewo dowiązań symbolicznych rc	49
KONFIGURACJA SIECI.....	50
Pliki konfiguracyjne	50
Programy konfiguracyjne	58
Programy testujące	60

5. SERWERY USŁUGOWE	64
SUPERSERWER INETD	65
Konfiguracja demona inetd.....	65
SERWER DHCP	68
Protokół DHCP	68
Konfiguracja serwera DHCP	68
SERWER SAMBY	72
Składniki pakietu Samba	72
Konfiguracja Samby	73
Konfiguracja za pomocą programu swat	100
Konfiguracja klientów Windows	101
6. BEZPIECZEŃSTWO SYSTEMU	106
ZAGADNIENIA BEZPIECZEŃSTWA.....	106
Rodzaje ataków.....	106
Polityka bezpieczeństwa.....	109
LINUX-PAM.....	110
Konfiguracja PAMów.....	111
Konfiguracja ograniczeń.....	113
FIREWALL	115
Tablice	115
Stany pakietów	116
Łańcuchy	117
Dopasowania ogólne.....	118
Dopasowania pośrednie.....	118
Dopasowania wyraźne	120
Cele i skoki	125
Konfiguracja iptables.....	128
7. ADMINISTROWANIE SERWERA	130
ZARZĄDZANIE UŻYTKOWNIKAMI.....	131
Cechy kont użytkowników	131
Pliki konfiguracyjne	134
Programy zarządzania użytkownikami	135
Limity dyskowe	137
KONTROLOWANIE SYSTEMU	139
Demon syslogd	139
Demon klogd	143
Administracja dziennikami.....	143
Analiza wydajności systemu.....	144
Rozliczanie użytkowników.....	149
ODTWARZANIE PO AWARII.....	150
Kopie zapasowe.....	150
Dyskiety ratunkowe.....	153
Dokumentacja systemu.....	154
SŁOWNIK POJĘĆ	155
BIBLIOGRAFIA	161

1. CEL PRACY

Celem mojej pracy był opis zainstalowania i skonfigurowania systemu pracującego na komputerze pełniącym rolę lokalnego serwera oraz opis późniejszego administrowania tym serwerem.

W swojej pracy przedstawiłem proces instalacji, a także najważniejsze zadania podczas konfiguracji systemu. Jeśli chodzi o usługi serwera, w pracy ograniczyłem się tylko do tych, które są istotne przede wszystkim dla właściwego funkcjonowania sieci lokalnej. Sporo miejsca poświęciłem także zagadnieniom bezpieczeństwa, a na koniec pokrótce przedstawiłem zadania i środki służące do administrowania serwerem.

2. ZAGADNIENIA WSTĘPNE

SYSTEMY UNIKSOWE

System operacyjny (*Operating System*) to oprogramowanie nadzorujące pracę komputera. Generalnie system operacyjny składa się z jądra (*kernel*) oraz programów użytkowych. Jądro przyjmuje zlecenia przesłane do niego przez programy użytkowe, użytkowników itd. i wykonuje je przydzielając im zasoby komputera takie jak pamięć, czas procesora czy urządzenia zewnętrzne. Jądro działa zawsze. Jest pierwszym programem, który startuje po włączeniu komputera i ostatnim, który jeszcze działa, gdy system zostaje zatrzymany.

Systemy uniksowe mają swoje implementacje na różnych maszynach. Są one systemami wielodostępnymi i wielozadaniowymi. Wielodostępność oznacza, że w tej samej chwili może, na tym samym komputerze, pracować wielu ludzi (np. łącząc się poprzez sieć). Wielozadaniowość pozwala na jednoczesne uruchamianie wielu programów. W system wbudowane są mechanizmy rozróżniania użytkowników (konta, hasła) oraz zabezpieczania plików przed niepożądanym dostępem (prawa dostępu). Możliwa jest także praca w sieci – potrzebne narzędzia standardowo znajdują się w systemie. Wiele wersji systemów uniksowych jest wyposażonych w środowisko graficzne (*X Window System*).

Jednak największym atutem systemów uniksowych jest ich silnie zmodularyzowana budowa. Oznacza to, że w systemie znajduje się bardzo dużo drobnych programów wykonujących niewielkie zadania.

Inną ważną cechą wyróżniającą systemy uniksowe spośród innych systemów jest reprezentowanie urządzeń jako specjalnych plików. Zwykle jednak bezpośredni zapis czy odczyt z takich plików specjalnych nie jest wymagany. Służą do tego odpowiednie narzędzia systemowe.

Systemy plików

System plików zarządza danymi zapisanymi na dysku. Chociaż każdy system komputerowy ma taki mechanizm, mogą się one jednak znacznie różnić. Współczesny system plików ma strukturę hierarchiczną. Użytkownik może umieszczać pliki w różnych katalogach ułatwiając sobie ich przeglądanie, a odwołuje się do nich za pomocą ścieżek. Linux wykorzystuje znak ukośnika (/) jako separatora w ścieżkach.

W systemie Linux ścieżki mogą być podawane bezwzględnie (ze znakiem / na początku) albo względem katalogu bieżącego. Katalog domowy użytkownika ma tu szczególne znaczenie. Przechowywane są tutaj wszystkie dane prywatne. Jest to także miejsce, w którym znajduje się użytkownik po załogowaniu.

W systemie uniksowym każdemu plikowi jest przydzielony i-węzeł (i-node), w którym są zapisane najważniejsze atrybuty, takie jak nazwa, prawa dostępu i adres pierwszego bloku. W katalogach są więc jedynie odwołania do odpowiednich i-węzłów.

Wirtualny system plików

Aby umożliwić rozwój różnych systemów plików, Linux implementuje dodatkową warstwę, wirtualny system plików (*virtual file system*) między jądrem a właściwymi procedurami systemu plików. Wirtualny system plików definiuje zestaw procedur, które muszą być dostępne w każdym systemie plików, służąc do ich otwierania, czytania i zapisywania. Ten jednoznaczny interfejs umożliwia bezkonfliktowe współlistnienie różnych systemów plików.

Konto

W systemach uniksowych, w celu umożliwienia weryfikacji danej osoby – a co się z tym wiąże jej praw do poszczególnych elementów systemu – każdemu użytkownikowi jest przypisany identyfikator. Całość elementów związanych z użytkownikiem określa się mianem konta (*account*), na które składa się m.in. nazwa, hasło, grupa, wydzielony katalog na dysku itd. Konto zakłada administrator (*superuser*) systemu, który standardowo ma identyfikator *root*. Konto administratora ma nieograniczone prawa.

Konta zwykle mają przypisane hasła, by uniknąć niepowołanego dostępu. Hasło nadaje administrator podczas tworzenia konta, może ono jednak być zmienione w każdej chwili przez danego użytkownika. Hasło jest znane jedynie użytkownikowi i nikt inny nie może go odczytać.

Prawa dostępu

Gdy w systemie uniksowym jest tworzony plik, system operacyjny zapisuje nie tylko nazwę i datę utworzenia pliku, ale także identyfikator użytkownika twórcy (lub właściciela) pliku oraz identyfikator grupy właściciela pliku. Aby chronić pliki w systemie plików przed niepożądanym dostępem, dla każdego z nich są zapisywane oddzielne prawa dostępu.

Dostęp do plików może więc być ograniczony do właściciela i pewnej grupy użytkowników. Możliwe jest zdefiniowanie ogólnych praw dostępu, wśród których rozróżnia się prawa czytania, pisania oraz wykonywania, co jest odzwierciedlane w wyjściu polecenia *ls* (list) literami *r*, *w* i *x*. Pozycja liter wskazuje, czy prawa dotyczą właściciela, grupy właścicieli czy wszystkich pozostałych użytkowników. Wyjątkiem są podkatalogi. Prawo czytania wystarcza jedynie do obejrzenia zawartości katalogu. Aby do niego wejść, użytkownik musi mieć dwa prawa – czytania i wykonywania.

Dowiązania

Inną cechą systemu plików uniksowych jest możliwość tworzenia dowiązań. Jeżeli plik ma być dostępny z różnych miejsc systemu plików, może zostać zwyczajnie skopiowany. Oczywiście takie podejście powoduje straty miejsca na dysku. W systemie Linux w takich przypadkach bardziej praktyczną możliwością jest tworzenie dowiązań. Uniksove dowiązania do plików mogą być twarde lub symboliczne. Dowiązanie twarde (*hard link*) jest to dodatkowe odwołanie z katalogu do pliku lub jego i-węzła. Licznik dowiązań przechowuje liczbę takich odwołań. Jeżeli ma być skasowany plik, do którego odwołuje się kilka dowiązań, to związany z nim licznik dowiązań jest zmniejszany o jeden. Dopiero, kiedy licznik dowiązań osiągnie wartość zero, plik może być fizycznie skasowany.

Ponieważ numery i-węzłów są unikatowe jedynie w ramach systemu plików, twarde dowiązania nie mogą być tworzone dla plików, które mają być dostępne spoza danego systemu plików.

Inaczej jest z dowiązaniem symbolicznymi (*symbolic links*), które mogą odwoływać się do dowolnych pozycji w katalogach (podkatalogów lub plików). To czy plik, do którego się odwołujemy, w ogóle istnieje, nie ma żadnego znaczenia przy tworzeniu dowiązania symbolicznego. Wyjście z polecenia *ls* pokazuje różnicę między tymi dwoma rodzajami dowiązań.

Sesja

Wszystko, co robi użytkownik od momentu rozpoczęcia pracy z systemem aż do jego zakończenia nazywa się sesją. Sesja rozpoczyna się od zalogowania się użytkownika, czyli podania systemowi przez użytkownika identyfikatora i hasła.

Po pomyślnym zalogowaniu użytkownika do systemu, jest ładowana powłoka, która przejmuje od tej chwili rolę nadrzędną.

Powłoka

Powłoka jest interpreterem poleceń i może być używana do uruchamiania, zatrzymywania, przerywania, a nawet pisania programów. Jest to integralna część Linuksa, jest również stałym elementem w przypadku systemów uniksowych. Kiedy przekazywane są polecenia z poziomu powłoki lub innych programów – jądro systemu odpowiednio na nie reaguje.

Linia poleceń powłoki

Gdy użytkownik korzysta z powłoki do uruchomienia programu, interpretuje ona polecenia i wyświetla na ekranie rezultat ich wywołania. Można również wysłać ten rezultat w inne miejsce, na przykład do pliku. Powłoka pozwala również na przesyłanie danych z wyjścia jednego programu na wejście innego.

Większość programów, które są uruchamiane z linii poleceń, ma możliwość czytania ze standardowego wejścia oraz zapisywania danych na standardowym wyjściu. Oprócz tych dwóch strumieni komunikacyjnych istnieje jeszcze jeden – standardowe wyjście diagnostyczne (zwykle przekazywane jest ono na ekran monitora).

Każde wejście i wyjście ma także przypisany numer pliku. Standardowe wejście ma numer 0 (`stdin`), standardowe wyjście – numer 1 (`stdout`), a standardowe wyjście diagnostyczne – numer 2 (`stderr`).

Procesy

Najmniejsza jednostka, która może być obsługiwana równolegle w systemie uniksowym, jest nazywana procesem (*process*) bądź zadaniem (*task*). Procesy wykonujące się równolegle w tym systemie mogą być programami różnych użytkowników albo programami zawsze działającymi w tle (demonami).

Ważną cechą charakterystyczną współczesnych systemów wielozadaniowych jest możliwość komunikacji między procesami (*interprocess communication*). Obejmuje ona funkcje synchronizacji lub wymiany danych między procesami.

Na konwencjonalnych komputerach z jednym procesorem musi być on dzielony między poszczególne procesy, aby dać użytkownikowi wrażenie równoczesnej ich pracy. Zadanie to jest wykonywane przez program szeregujący (*scheduler*) specjalny proces zarządzający listą zwykłych procesów i zlecający procesorowi obsługę kolejnych procesów we właściwych odcinkach czasu. Istnieją różne strategie, które mogą być wykorzystywane przez program szeregujący w celu wyznaczenia kolejnego procesu do obsługi. W jednej z najprostszych strategii (*round robin*) kolejny proces jest wybierany z listy w ustalonych odstępach czasu (np. co 50 ms), a następnie po wykorzystaniu przydzielonego czasu wstawiany na koniec listy, jeżeli proces nie zakończył jeszcze działania. Inna strategia przypisuje każdemu procesowi priorytet, po czym procesy z wyższym priorytetem otrzymują większy przydział czasu procesora.

W systemach uniksowych są stosowane poziomy uprzejmości (*nice levels*), dzięki którym użytkownik może wpływać na wewnętrzne priorytety procesów. Pozwala to znacznie zmniejszyć obciążenie systemu przez programy pracujące w tle. Ponadto administrator systemu może także zwiększyć priorytet ważnych procesów.

Systemy uniksowe pozwalają na wysyłanie do procesów krótkich komunikatów, nazywanych sygnałami. W zależności od sygnału, procesy mogą przechwycić i zinterpretować sygnał lub zakończyć działanie. Sygnały są także dostarczane w przypadku, kiedy proces wykonuje niepożądane działania lub jako odpowiedź na wciśnięcie pewnych klawiszy.

Specjalną grupę procesów stanowią demony. Są to procesy uruchomione w tle i wykonujące określone operacje w określonych predefiniowanych momentach czasu lub w odpowiedzi na określone zdarzenia. Demon jest cały czas aktywny w czasie, gdy system jest uruchomiony, do momentu, kiedy nie zostanie zatrzymany.

Procesy wyróżniają następujące atrybuty:

- Unikalny numer ID procesu (*pid*) odróżniający dany proces od innych procesów
- Numer ID (*pid*) procesu nadrzędnego
- Identyfikator ID użytkownika oraz identyfikator ID grupy użytkownika, który uruchomił proces
- Stan wskazujący na to, czy proces wykonuje się, jest uśpiony, jest gotowy do uruchomienia, jest w stanie zamrożonym lub jest zatrzymany
- Nazwa terminala, do którego proces jest dołączony (zwykle tylko procesy interaktywne mają przypisane terminale).

Model warstwowy systemów uniksowych

Struktura systemu Linux jest często przedstawiana za pomocą modelu warstwowego. Rdzeniem systemu Linux jest jego jądro (*kernel*), najbardziej wewnętrzna warstwa systemu operacyjnego. Jądro systemu odgrywa rolę pośrednika między programami a sprzętem. Jądro systemu nadzoruje i koordynuje posługiwanie się sprzętem przez różne programy użytkowe, które pracują na zlecenie różnych użytkowników. Wykonuje ono różne operacje podstawowe w imieniu procesów użytkownika. Jądro dostarcza sterowników dla wszystkich najpopularniejszych elementów sprzętu i z pewnością ma najbardziej istotny wpływ na wydajność systemu. W jądrze realizowane jest także zarządzanie pamięcią.

Procesy w jądrze różnią się od procesów wykonywanych w warstwie je otaczającej. Zwyczajne procesy użytkowników, którymi zarządza proces szeregujący, mogą być przerwane w dowolnym momencie, a każdy z nich ma przydzielony pewien obszar pamięci. Jeżeli proces użytkownika próbuje dostać się do nie swojego obszaru pamięci, to zostaje przerwany z komunikatem *segmentation fault* (błąd segmentacji). Bieżąca zawartość pamięci procesu może zostać w takim przypadku zapisana do pliku o nazwie *core* (zrzut pamięci). Plik ten może przydać się programiście przy szukaniu błędów.

Inaczej jest w przypadku procesów jądra, które mają dostęp do wszystkich zasobów komputera. Rozróżnia się zatem dwa tryby, w których może być wykonywany proces: tryb użytkownika oraz tryb jądra.

Zewnętrzna warstwa systemu Linux składa się z programów, które kontaktują się bezpośrednio z użytkownikiem. Warstwa ta zawiera interpretator poleceń (*command shell*), uruchamiający polecenia systemu operacyjnego oraz programy użytkowe.

Między zewnętrzną warstwą a jądrem znajdują się różne biblioteki zapewniające dostęp do funkcji bibliotecznych (zwykle napisanych w języku C) oraz do procedur jądra. Biblioteki te są zwykle dołączane do programu po kompilacji przez dodanie procedur bibliotecznych do procedur samego programu.

Ponieważ statycznie skonsolidowane programy wymagają dużych obszarów pamięci, zazwyczaj są używane biblioteki współdzielone (*shared libraries*), składające się z dwóch części. Mała część, zawierająca jedynie odwołania do biblioteki, jest dołączana do

programu. Sama biblioteka jest ładowana dopiero w momencie uruchomienia programu. Biblioteki współdzielone umożliwiają wielu programom jednocześnie używanie znajdujących się w nich procedur, co oszczędza pamięć.

Zarządzanie pamięcią

Linux wykorzystuje mechanizm zarządzania pamięcią wirtualną, tzn. system operacyjny wydaje się zapewniać więcej pamięci operacyjnej niż ma w rzeczywistości do dyspozycji.

Metoda wykorzystywana do implementacji pamięci wirtualnej w systemie Linux jest nazywana stronicowaniem (*paging*). Za pomocą specjalnych tablic system operacyjny odwzorowuje dużą logiczną przestrzeń adresową w mniejszą fizyczną. Kiedy proces żąda większej ilości pamięci niż jest fizycznie dostępnej, pojedyncze strony pamięci logicznej, do których nie było ostatnio odwołań, są przenoszone na dysk.

Kiedy program sięga do adresu logicznego znajdującego się właśnie na dysku, odpowiedni segment pamięci, nazywany stroną (*page*), jest umieszczany w pamięci głównej, podczas gdy inny musi być dla równowagi zapisany na dysk. Z powodu znacznie dłuższego czasu dostępu do dysku w porównaniu z czasem dostępu do pamięci operacyjnej, szybkość działania zmniejsza się.

Aby umożliwić wykorzystanie dysku przy zarządzaniu pamięcią wirtualną oraz logiczną pamięcią operacyjną, muszą być na nim utworzone pliki lub partycje wymiany.

SIECI KOMPUTEROWE

Model warstwowy OSI

Model OSI (*Open Systems Interconnection*) opisuje sposób przepływu informacji między aplikacjami w jednej stacji sieciowej a aplikacjami w innej stacji sieciowej przy użyciu medium transmisyjnego. Model OSI jest ogólnym modelem koncepcyjnym, skomponowanym z siedmiu warstw, z których każda opisuje określone funkcje sieciowe, ale nie określa szczegółowych metod komunikacji. Mechanizmy rzeczywistej komunikacji są określone w formie protokołów komunikacyjnych.

Warstwy modelu OSI:

- **Warstwa 7 – Aplikacji**

Jest bramą, przez którą procesy aplikacji dostają się do usług sieciowych. Ta warstwa prezentuje usługi, które są realizowane przez aplikacje (przesyłanie plików, poczta elektroniczna itp.).

- **Warstwa 6 – Prezentacji danych**

Odpowiada za format używany do wymiany danych pomiędzy komputerami w sieci. Na przykład kodowanie i dekodowanie danych odbywa się w tej samej warstwie.

- **Warstwa 5 – Sesji**

Pozwala aplikacjom z różnych hostów nawiązywać, wykorzystywać i kończyć połączenie (sesję). Warstwa ta tłumaczy nazwy systemów na właściwe adresy.

- **Warstwa 4 – Transportu**

Jest odpowiedzialna za dostawę wiadomości, które pochodzą z warstwy aplikacyjnej. U nadawcy warstwa transportu dzieli długie wiadomości na kilka pakietów, natomiast u odbiorcy odtwarza je i wysyła potwierdzenie odbioru. Sprawdza także, czy dane zostały przekazane we właściwej kolejności i na czas.

- **Warstwa 3 – Sieciowa**

Kojarzy logiczne adresy sieciowe i ma możliwość zamiany adresów logicznych na fizyczne. U nadawcy warstwa sieciowa zamienia duże pakiety logiczne w małe fizyczne ramki danych, zaś u odbiorcy składa ramki danych w pierwotną logiczną strukturę danych.

- **Warstwa 2 – Łączy transmisyjnego**

Zajmuje się pakietami logicznymi (lub ramkami) danych. Pakuje nieprzetworzone bity danych z warstwy fizycznej w ramki, których format zależy od typu sieci. Ramki używane przez tą warstwę zawierają fizyczne adresy nadawcy i odbiorcy danych.

- **Warstwa 1 – Fizyczna**

Przesyła nieprzetworzone bity danych przez fizyczny nośnik. Ta warstwa przenosi dane generowane przez wszystkie wyższe poziomy.

Model sieci TCP/IP

W większości zastosowań przyjmuje się model warstwowy usług sieciowych, który może być odwzorowany w modelu odniesienia OSI. Na przykład model sieciowy TCP/IP można adekwatnie wyrazić przez uproszczony model odniesienia.

Aplikacje sieciowe zazwyczaj zajmują się trzema najwyższymi warstwami (sesji, prezentacji i aplikacji) siedmiowarstwowego modelu odniesienia OSI. Stąd te trzy warstwy mogą być połączone w jedną, zwaną warstwą aplikacyjną.

Dwie najniższe warstwy modelu OSI (fizyczna i łącza transmisyjnego) także można połączyć w jedną warstwę. W efekcie otrzymujemy uproszczony czterowarstwowy model:

- Warstwa 4 – Aplikacyjna
- Warstwa 3 – Transportu
- Warstwa 2 – Sieciowa
- Warstwa 1 – Fizyczna

W każdej z tych warstw informacje są wymieniane przez jeden z wielu protokołów sieciowych.

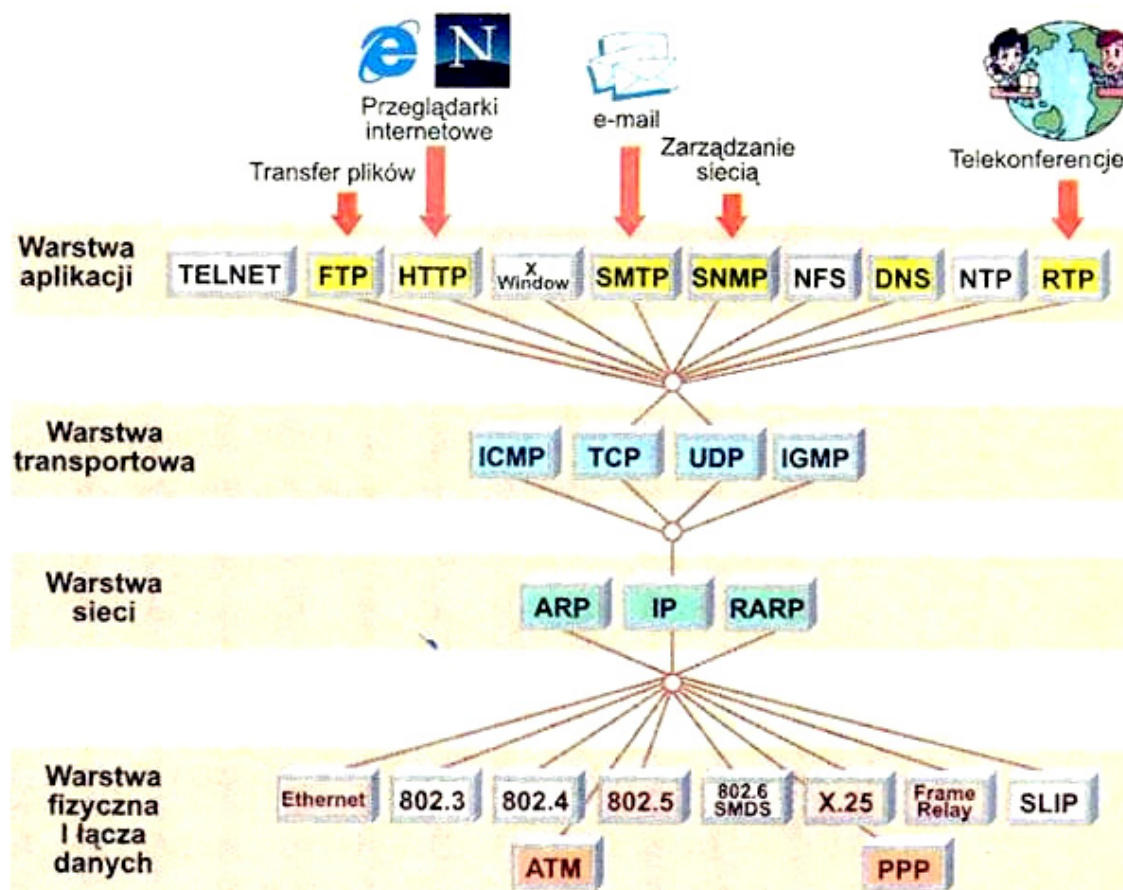
Protokoły sieciowe

Protokół sieciowy określa sposób transmisji danych na poziomie kanału fizycznego, zbiór procedur sterowania transmisją i sposób postępowania podczas inicjowania, utrzymania i zakończenia transmisji, a także sposób kontroli poprawności przekazu na określonej warstwie modelu sieciowego. W uproszczonym czterowarstwowym modelu sieciowym można wyróżnić następujące protokoły:

- Protokoły warstwy fizycznej (np. Ethernet)
- Protokoły warstwy sieciowej (np. IP)
- Protokoły warstwy transportu (np. TCP i UDP)
- Protokoły warstwy aplikacyjnej (np. FTP i HTTP)

Określenie „zestaw protokołów” oznacza dwa lub więcej protokołów z tych warstw, które stanowią podstawę sieci. Do najczęściej spotykanych zestawów protokołów należą:

- **Zestaw protokołów IPX/SPX**
„międzysieciowa wymiana pakietów” / „sekwencyjna wymiana pakietów” używany przez system Novell Netware.
- **NetBIOS i NetBEUI**
„rozszerzony interfejs użytkownika podstawowego sieciowego systemu wejścia / wyjścia” zaprojektowany przez firmę IBM, wykorzystywany m.in. przez systemy operacyjne Microsoftu. Ponadto NetBIOS może być tunelowany dowolnym innym protokołem.
- **Zestaw protokołów TCP/IP**
„protokół kontroli transmisji” / „protokół internetowy” używany powszechnie w Internecie oraz sieciach lokalnych mających do niego dostęp.



Rys. 1. Protokoły zestawu TCP/IP w czterowarstwowym modelu sieciowym

Sposoby transmisji i adresowania

Wyróżnia się trzy sposoby transmisji i adresowania w sieci:

- **Transmisja pojedyncza (*Unicast*)**
Stacja nadawcza adresuje pakiet używając adresu stacji odbiorczej. Pojedynczy pakiet jest wysyłany przez stację nadawczą do stacji odbiorczej.
- **Transmisja grupowa (*Multicast*)**
Stacja nadawcza adresuje pakiet używając adresu multicast. Pojedynczy pakiet danych jest wysyłany do grupy stacji sieciowych (określonej przez adres multicast).
- **Transmisja rozgłoszeniowa (*Broadcast*)**
Stacja nadawcza adresuje pakiet używając adresu broadcast. W tym typie transmisji pakiet jest wysyłany do wszystkich stacji sieciowych.

Zestaw TCP/IP

Adresy IP (IPv4)

W sieciach TCP/IP adres komputera zwany jest adresem IP. Oryginalny adres IP jest czterobajtową (32-bitową) liczbą. Przyjęła się konwencja zapisu każdego bajtu w postaci dziesiętnej i oddzielania ich kropkami. Ten sposób zapisu zwany jest notacją kropkowo-dziesiętną. Określona liczba 32-bitowego adresu IP jest adresem sieciowym,

a reszta adresem hostowym. Adres sieciowy określa sieć LAN, zaś adres hosta konkretną stację roboczą w tej sieci. Adres hosta złożony z samych zer (binarnie) jest zarezerwowany do identyfikacji sieci, natomiast adres złożony z samych jedynek służy do rozgłaszania (broadcast) w tej sieci.

Wynika stąd, że adresy IP mieszczą się w zakresie od 0.0.0.0 do 255.255.255.255. Pewne grupy adresów są jednak zarezerwowane do specjalnych celów:

- Adresy 0.xxx.xxx.xxx wskazują na dany host w lokalnej sieci
Dodatkowo adres 0.0.0.0 oznacza wszystkie komputery w całej sieci
- Adresy 10.xxx.xxx.xxx są zarezerwowane do używania wewnątrz sieci lokalnej (1 sieć klasy A)
- Adresy 127.xxx.xxx.xxx są używane do testu zwrotnego (loopback) – komunikacji hosta z samym sobą, bez wysyłania pakietów w sieć
- Adresy od 172.16.0.0 do 172.31.255.255 są zarezerwowane do używania wewnątrz sieci lokalnej (16 sieci klasy B)
- Adresy 192.168.xxx.xxx są zarezerwowane do używania wewnątrz sieci lokalnej (255 sieci klasy C)
- Adresy od 224.0.0.0 do 239.255.255.255 służą do transmisji grupowej (multicast)
- Adresy od 240.0.0.0 do 255.255.255.255 są zarezerwowane przez zespół IETF do własnych badań.

Maska sieciowa

Maska sieci składa się podobnie jak adres IP z 32 bitów (4 bajty po 8 bitów każdy). Jest używana do wydzielenia części adresu, która odpowiada za identyfikację sieci i części odpowiadającej za identyfikację komputera w danej podsieci z adresu IP. Jeżeli bit w masce wynosi 1 to odpowiadający mu bit w adresie IP jest implementowany jako bit adresu sieci, jeżeli bit maski wynosi 0, to oznacza jego przynależność do części określającej hosta.

Poniżej znajduje się przykład wyznaczania adresu sieci i adresu rozgłoszeniowego (Broadcast) na podstawie adresu IP i maski sieci.

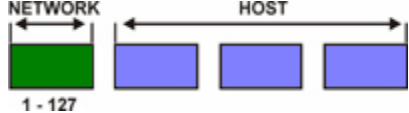
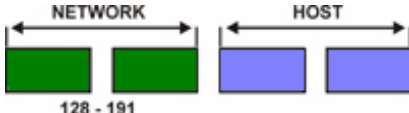
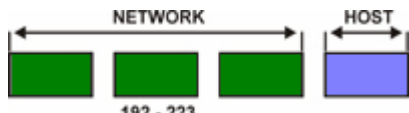
	Notacja kropkowo-dziesiętna	Notacja binarna
Adres IP	81. 21.195.159	01010001 00010101 11000011 10011111
Maska sieciowa	255.255.255.192	11111111 11111111 11111111 11000000

Adres sieci uzyskujemy w wyniku iloczynu binarnego adresu IP i maski sieciowej. Aby utworzyć adres rozgłoszeniowy należy do zanegowanego adresu maski sieciowej dodać binarnie adres IP.

	Notacja binarna	Notacja kropkowo-dziesiętna
Adres sieci	01010001 00010101 11000011 10000000	81.21.195.128
Adres broadcast	01010001 00010101 11000011 10111111	81.21.195.191

Klasy sieci

Pierwotnie bity określające sieć i bity określające komputer były rozróżniane za pomocą tzw. klas adresów IP. Klasy prezentują odmienne uzgodnienia dotyczące liczby obsługiwanych sieci i hostów. Klasy były definiowane za pomocą kilku pierwszych bitów adresu. Na podstawie ich wartości oprogramowanie określało klasę adresu, a tym samym które bity odpowiadają za adres sieci, a które za adres hosta.

	Pierwsze 4 bity adresu	Zakres adresów IP	Domyślna maska sieci	Ilustracja podziału adresu IP na część sieciową i część hosta
Klasa A	0xxx	1.0.0.0 – 127.255.255.255	255.0.0.0	
Klasa B	10xx	128.0.0.0 – 191.255.255.255	255.255.0.0	
Klasa C	110x	192.0.0.0 – 223.255.255.255	255.255.255.0	

Tab. 1. Podział adresów IP na klasy

Rozróżnia się jeszcze klasy D i E, jednak nie są one stosowane do adresowania poszczególnych hostów. Adresy klasy D służą do obsługi transmisji grupowej (multicastingu), natomiast adresy klasy E są zarezerwowane przez zespół IETF do własnych badań.

Domeny i hosty

Nazwy domen określające przynależność poszczególnych sieci do większych struktur są tworzone z uwzględnieniem pewnych reguł. Pełna nazwa domeny składa się z kilku członów rozdzielonych kropkami. Każdy z tych członów oznacza pewien poziom w hierarchii. Pierwszy człón z prawej strony to nazwa domeny najwyższego poziomu (*top-level domain name*). Na lewo od niego, po kropce, znajduje się kolejny człón itd., aż do człónu znajdującego się na ostatniej pozycji licząc od prawej strony (nazwa części hosta), który zwykle można wybrać samodzielnie.

Komputerom i urządzeniom w sieci nadawane są nazwy hostów będące uzupełnieniem adresu IP. Nazwy hostów można stosować zamiennie z adresami IP, możliwa jest także konwersja jednych na drugie.

Nazwa domeny identyfikuje sieć, do której należy komputer lub urządzenie. Kiedy komputerowi zostanie nadana jakaś nazwa, to razem z nazwą domeny tworzy ona pełną nazwę domenową (*Fully Qualified Domain Name*). Nazwa hosta składa się z dwóch części: części domenowej – wskazującej na przynależność do sieci – i części hosta określającej nazwę komputera lub urządzenia. Na przykład w nazwie *fraxinus.tk.krakow.pl* część domenowa to *tk.krakow.pl*, a część hosta (nazwa komputera) to *fraxinus*.

Porty

W pojedynczym komputerze może być uruchomionych kilka usług. Aby oddzielić te usługi, potrzeba właśnie portów, gdyż sam adres IP hosta nie wystarczy. Komputer z pojedynczym adresem IP może oferować wiele różnych usług pod warunkiem, że każda z nich wykorzystuje port o oddzielnym numerze. Każdy protokół posiada własny zbiór numerów portów, dlatego różne usługi mogą korzystać z tych samych numerów portów pod TCP oraz pod UDP.

Serwer może „słuchać” dowolnego portu. Standardy w odpowiadającym usługom numerach portów stanowi dokument RFC-1700.

Port	Protokół	Nazwa	Zastosowanie
7	TCP	<i>echo</i>	Wyświetla wszystko, co otrzymuje
13	TCP	<i>daytime</i>	Wysyła w odpowiedzi bieżącą datę i czas
21	TCP	<i>ftp</i>	Zdalne przesyłanie plików
22	TCP	<i>ssh</i>	Zdalne logowanie SSH
23	TCP	<i>telnet</i>	Emulacja zdalnego terminala
25	TCP	<i>smtp</i>	Transfer poczty elektronicznej
53	UDP	<i>domain</i>	System nazw domen (DNS)
80	TCP	<i>www</i>	Ruch stron WWW
110	TCP	<i>pop3</i>	Protokół pocztowy, wersja 3
220	TCP	<i>imap3</i>	Interakcyjny protokół udostępniania poczty
6667	UDP	<i>ircd</i>	Internet Relay Chat

Tab. 2. Niektóre znane numery portów

Gniazda

Gniazdo (*socket*) to połączenie sieciowe pomiędzy dwoma procesami, które mogą działać na tym samym lub różnych komputerach. Gniazdo posiada porty po obu stronach połączenia. Kiedy klient próbuje połączyć się z serwerem, najpierw prosi o wolny port (taki, który nie jest używany przez żaden inny program). Następnie prosi system o połączenie z hostem docelowym i jego portem docelowym za pomocą portu źródłowego. Dlatego pomiędzy tymi samymi hostami może być połączonych kilka programów. System przechowuje informacje zarówno o porcie źródłowym, jak i docelowym i posiada różne gniazda dla każdego z połączeń.

Komunikacja sieciowa

Rozwiązywanie adresów

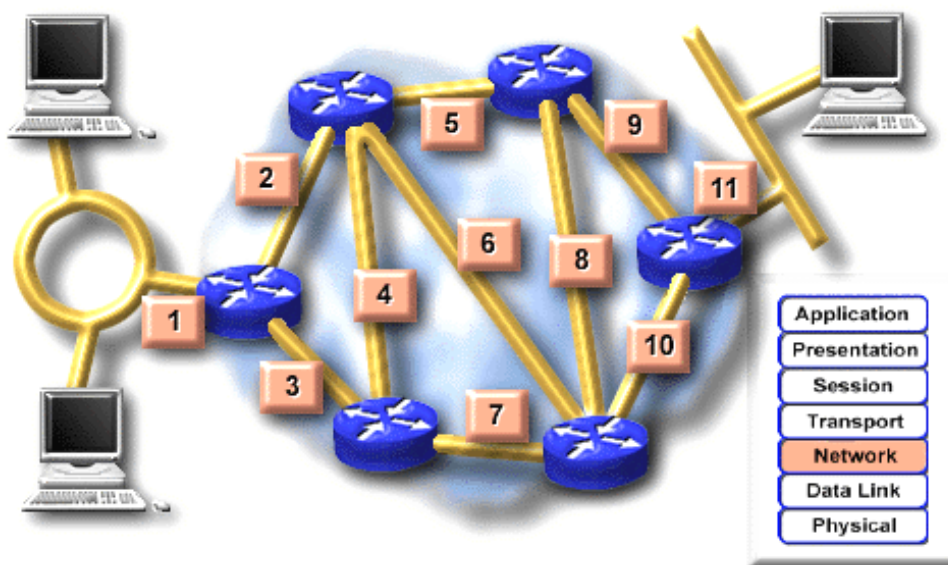
Aby adresy IP były używane do adresowania hostów w sieci potrzebny jest mechanizm, który odwzorowuje adresy IP na adresy sieci niższej warstwy. Tym mechanizmem jest protokół rozwiązywania adresów (*ARP, Address Resolution Protocol*).

Gdy ARP chce znaleźć adres ethernetowy odpowiadający określonemu adresowi IP, wykorzystuje rozgłaszanie. Datagram rozgłoszeniowy wysyłany przez ARP zawiera zapytanie o adres IP. Każdy host, który go odbierze, porównuje to zapytanie ze swoim własnym adresem IP. Jeżeli znajdzie się host, którego adres IP odpowiada poszukiwanemu, to zwraca on odpowiedź ARP do pytającego hosta. Pytający host może teraz – na podstawie odpowiedzi – odczytać adres ethernetowy nadawcy.

Trasowanie

Trasowanie, czyli wybór marszruty, należy do najważniejszych funkcji sieci IP. Jest to proces odkrywania, porównywania i wyboru ścieżek prowadzących przez sieć do docelowego adresu IP. Na ogół przeprowadzają go urządzenia, zwane routerami.

Router posiada połączenia z dwoma lub większą liczbą sieci. Jego zadaniem jest przesyłanie pakietów pomiędzy nimi. Kiedy host przesyła pakiet, który ma dotrzeć do hosta wewnątrz tej samej sieci, przesyła go bezpośrednio do hosta docelowego. Jeżeli jednak host docelowy znajduje się w innej sieci, przesyła ten pakiet do routery, który z kolei przesyła go do właściwej sieci.



Rys. 2. Przykładowa trasa między dwoma sieciami

Maskowanie

Maskowanie adresów IP (*masquerading*) polega na zamienianiu adresów IP w przesyłanych pakietach. Router przechwytuje pakiety wysyłane przez hosty z sieci lokalnej i wykonuje w czasie rzeczywistym inteligentne tłumaczenie adresów IP i portów.

Gdy router odbierze datagram od hosta z sieci lokalnej, sprawdza typ datagramu i modyfikuje go tak, że wygląda on jakby był wygenerowany przez sam router. Następnie jako adres źródłowy ustawia swój adres i wysyła tak zamienione pakiety do sieci zewnętrznej. Po odebraniu pakietu z odpowiedzią adres docelowy zostaje zamieniony na adres hosta w sieci wewnętrznej i pakiet zostaje przesłany do danego hosta.

Tunelowanie

Sieciowe tunele pozwalają w wielu przypadkach obejść techniczne ograniczenia środowisk sieciowych. Tunelowanie możliwe jest dzięki kapsułkowaniu (hermetyzacji), czyli zagnieżdżaniu pakietów jednego w drugim – w ten sposób można przenieść przez sieć rozległą IP pakiety protokołu sieci lokalnej, np. IPX. W pakietach IP można również kapsułkować pakiety IP na przykład w celu przesyłania tunelem pakietów IP multicast poprzez routery, które nie obsługują trasowania multicastowego.

Przez tunel można „przenieść” logiczne stacje sieciowe z jednej sieci lokalnej do innej, rozciągając usługi sieci lokalnej na większy obszar, bez konieczności rozbudowy fizycznej infrastruktury. Tunele umożliwiają zaawansowanych intranetów ze wszystkimi szukaniami. Pozwalają też utajnić transmisję na poziomie protokołu sieciowego.

Wadą technologii tunelowania jest narzut spowodowany przesyłaniem dodatkowych nagłówków kapsułkowanych pakietów, który zmniejsza wydajność przesyłania danych.

3. INSTALACJA SYSTEMU

Instalowany system operacyjny, na którym będzie funkcjonować serwer to dystrybucja *Debian GNU/Linux*, która sięga korzeniami roku 1993. Nad rozwojem tej dystrybucji pracuje rzesza programistów - ochotników z całego świata.

Dystrybucję tą wyróżnia przede wszystkim:

- Dbłość o to, aby wszystkie wchodzące w skład dystrybucji pakiety były wolnodostępne (*free*).
- Nastawienie na stabilność i bezpieczeństwo.
- Wysokiej jakości format pakietów oprogramowania (format *deb*) oraz bogate narzędzia do zarządzania nimi.
- Bardzo duża liczba pakietów wchodzących w skład dystrybucji.
- Łatwy i efektywny sposób aktualizacji oprogramowania.
- Szybki system reakcji na wykryte usterki – poprawione wersje pakietów pojawiają się w sieci zazwyczaj w ciągu 48 godzin od ujawnienia błędu.

PROCES INSTALACJI

System Debian można zainstalować z wielu źródeł, zarówno lokalnych (CD-ROM, dysk twardy, dyskietki), jak i z sieci (FTP, NFS, PPP, HTTP). Debian obsługuje różne konfiguracje sprzętu, więc często można wybrać spośród wielu ewentualności.

Dla różnych etapów instalacji można dokonać różnych wyborów. Na przykład instalację można zacząć z CD-ROMu, a później dostarczyć resztę potrzebnych plików z sieci.

W czasie instalacji, system będzie się zmieniał: od małego, ograniczonego i mieszczącego się na RAM-dysku – do pełnego systemu Debian, zainstalowanego na dysku twardym. Jednym z podstawowych zadań początkowych faz instalacji jest zwiększenie ilości obsługiwanego przez system sprzętu i oprogramowania. Dzięki temu w późniejszych fazach instalacji można korzystać z większej ilości źródeł plików niż wcześniej.

Proces instalacji składa się z następujących kroków:

- Załadowanie systemu instalacyjnego
- Odpowiedź na kilka pytań wstępnie konfiguracyjnych system
- Wskazanie nośnika zawierającego jądro i sterowniki
- Wybranie sterowników do załadowania
- Wskazanie nośnika zawierającego źródło systemu podstawowego
- Zrestartowanie systemu i zakończenie konfiguracji
- Instalacja dodatkowych pakietów oprogramowania.

System instalacyjny

Załadowanie systemu instalacyjnego jest jednym z najtrudniejszych etapów instalacji. Do wyboru jest zazwyczaj jeden z nośników:

- dyskietka ratunkowa (*Rescue Floppy*)
- dysk twardy, dzięki programowi ładującemu w innym systemie operacyjnym
- CD-ROM.

Nośniki instalacyjne

Obraz dyskietki to plik zawierający całą zawartość dyskietki w „surowej” postaci. Do umieszczenia obrazu na dyskietce należy użyć specjalnego programu, który zapisze je w „surowym” trybie.

Aby zapisać obrazy na dyskietki w systemie Linux lub UNIX, należy jako administrator wydać polecenie:

```
dd if=<plik obrazu> of=/dev/fd0 bs=1024 conv=sync ; sync
```

W systemach DOS, Windows lub OS/2 należy użyć programu *rawrite2.exe*. W przypadku użycia wiersza poleceń DOS-u należy użyć komendy:

```
rawrite2 -f <plik obrazu> -d <napęd>
```

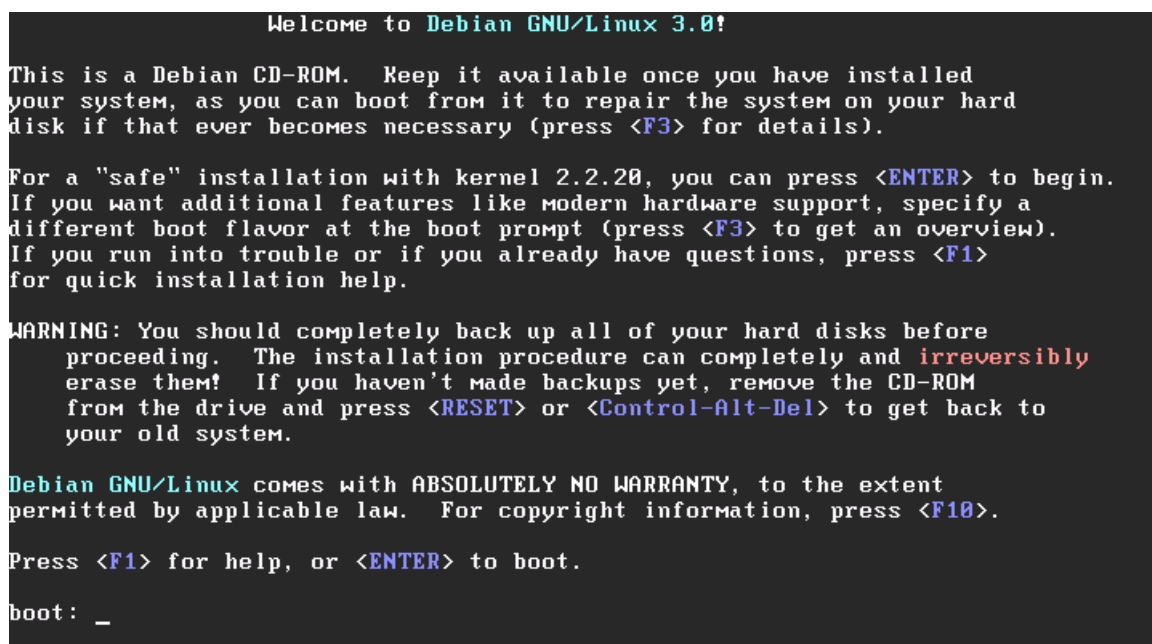
Ładowanie systemu z dyskietki *Rescue Floppy* jest łatwe: należy ustawić w BIOSie uruchamianie komputera z napędu dyskietek, włożyć do napędu dyskietkę ratunkową i zrestartować komputer. Dyskietka nazywa się *Rescue Floppy*, ponieważ można jej użyć do załadowania systemu i wykonania napraw, jeśli pojawi się problem, który spowoduje, że systemu nie będzie się dało uruchomić z dysku twardego.

W przypadku instalacji systemu, po naciśnięciu [Enter] nastąpi wstępne skonfigurowanie sprzętu, a następnie pojawi się prośba o dyskietkę z głównym systemem plików.

Istnieje możliwość zainstalowania Debiana z istniejącej już partycji DOS. Aby zainstalować system bez użycia dyskietki ratunkowej, należy umieścić pliki z serwera FTP w katalogu na partycji DOS, z zachowaniem struktury katalogów. Wśród plików muszą znajdować się: program *loadlin.exe*, jeden z obrazów ratunkowych, jeden z obrazów głównego systemu plików, jeden z plików jądra Linux, jeden z wsadowych plików DOS i jedno z archiwów sterowników (musi ono odpowiadać wybranemu jądru).

Następnie należy załadować system DOS bez ładowania sterowników (w przypadku systemu Windows należy nacisnąć [F8] podczas ładowania systemu i wybrać opcję Tylko wiersz poleceń trybu awaryjnego). Z kolei należy przejść do katalogu, w którym znajduje się jądro w wybranym „smaku” i uruchomić skrypt *install.bat*.

Ładowanie systemu z CD-ROMu jest jednym z najłatwiejszych sposobów na zainstalowanie systemu. Jeśli dana architektura sprzętowa obsługuje ładowanie z CD-ROMu, nie będą potrzebne żadne dyskietki. Wystarczy w BIOSie ustawić uruchamianie komputera z napędu CD, włożyć do napędu CD-ROM z obrazem systemu i zrestartować komputer.



```

Welcome to Debian GNU/Linux 3.0!

This is a Debian CD-ROM.  Keep it available once you have installed
your system, as you can boot from it to repair the system on your hard
disk if that ever becomes necessary (press <F3> for details).

For a "safe" installation with kernel 2.2.20, you can press <ENTER> to begin.
If you want additional features like modern hardware support, specify a
different boot flavor at the boot prompt (press <F3> to get an overview).
If you run into trouble or if you already have questions, press <F1>
for quick installation help.

WARNING: You should completely back up all of your hard disks before
proceeding.  The installation procedure can completely and irreversibly
erase them!  If you haven't made backups yet, remove the CD-ROM
from the drive and press <RESET> or <Control-Alt-Del> to get back to
your old system.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.  For copyright information, press <F10>.

Press <F1> for help, or <ENTER> to boot.

boot: _
```

Rys. 3. Start z płyty CD

Wybór „smaku”

Oficjalne CD-ROMy dla architektury Intel x86 zawierają różne „smaki” jądra. „Smaki” są obrazami jądra, z których każdy obsługuje inny zestaw sprzętu. Dostępne w systemie Debian GNU/Linux 3.0 obrazy jądra dla architektury Intel x86 to:

- **vanilla**
Standardowy pakiet jądra w Debianie. Zawiera prawie wszystkie sterowniki obsługiwane przez Linuksa, zbudowane w postaci modułów. Są to między innymi sterowniki dla urządzeń sieciowych, urządzeń SCSI, kart dźwiękowych itp.
- **compact**
Podobny do „smaku” *vanilla*, ale nie zawiera wielu rzadziej używanych sterowników. Dodatkowo posiada wbudowaną obsługę popularnych kart sieciowych PCI (*NE2000*, *3com*, *Tulip*, *Via-Rhine* i *Intel EtherExpress Pro100*). Dzięki tym sterownikom można zainstalować sterowniki i system podstawowy przez sieć. „Smak” ten zawiera również kilka popularnych sterowników RAID.
- **idepci**
Jądro to obsługuje tylko urządzenia IDE i PCI (oraz bardzo niewielką liczbę urządzeń ISA). Tego „smaku” należy użyć w przypadku, gdy sterowniki SCSI w innych „smakach” powodują zawieszenie systemu przy starcie. Jądro *idepci* ma także wbudowany sterownik dyskiety IDE, więc można go wykorzystać do instalacji z urządzeń LS120 lub ZIP.
- **bf2.4**
Jest to eksperymentalny „smak” używający wersji jądra w wersji 2.4. Dostarcza on wsparcia dla nowszego sprzętu, dla którego nie ma wsparcia w pozostałych „smakach”. W jądrze tym dostępne są sterowniki m.in. do większości urządzeń USB, nowszych kontrolerów IDE, niektórych nowych kart sieciowych, czy systemu plików *Ext3* oraz *Reiser*.

Argumenty ładowania

Parametry ładowania jądra Linux zazwyczaj używane są do upewnienia się, że urządzenia peryferyjne działają poprawnie. W większości wypadków jądro może samo wykryć informacje dotyczące urządzeń zewnętrznych. W niektórych wypadkach należy jednak przekazać pewne informacje do jądra na temat przyłączonych urządzeń, których ono samo nie mogłoby wykryć, lub w celu zmiany ustawień tych urządzeń.

W przypadku ładowania systemu instalacyjnego z dyskiety *Rescue Floppy* lub z CD-ROMu, pojawi się zachęta ładowania (*boot prompt*) *boot:*. Jeśli system uruchamiany jest po raz pierwszy, nie należy podawać żadnych argumentów ładowania. Jeśli podczas jego pracy wystąpią błędy, najprawdopodobniej, należy jeszcze raz uruchomić system instalacyjny i podać odpowiednie argumenty ładowania w zależności od błędów, jakie wystąpiły podczas poprzedniej próby. Jeśli natomiast instalacja odbywa się z systemu DOS, należy odpowiednio zmodyfikować plik *install.bat*.

Argument	Opis
mem=<ram><k m>	Podaje ilość całkowitej dostępnej ilości pamięci w systemie, wyrażonej w kilobajtach (k) lub megabajtach (m).
floppy=thinkpad	Stosowane do napędów dyskietek z „odwróconymi DCL”.
hd=<cylindry>, <głowice>,<sektory>	Podaje parametry używanego dysku twardego, to jest ilość cylindrów, głowic i sektorów.
mono	Używa trybu monochromatycznego zamiast kolorowego do wyświetlania.
quiet	System instalacyjny pominie prośby o potwierdzenie i spróbuje samodzielnie wykonać właściwe czynności.
vebose	System instalacyjny będzie zadawał jeszcze więcej pytań niż normalnie.
debug	Generuje dodatkowe komunikaty do dziennika systemowego, w tym wszystkie uruchamiane komendy.

Tab. 3. Najważniejsze argumenty ładowania

Program *dbootstrap*

Program *dbootstrap* jest uruchamiany po załadowaniu systemu instalacyjnego. Jest on odpowiedzialny za początkową konfigurację systemu i instalację „systemu podstawowego”.

Głównym zadaniem *dbootstrap* oraz głównym celem początkowej konfiguracji systemu, jest ustawienie pewnych podstawowych elementów systemu. Przy jego pomocy ładuje się konieczne moduły jądra, czyli sterowniki dołączane do jądra w odpowiednim momencie. Modułami są sterowniki urządzeń pamięci masowych, kart sieciowych, obsługa języków oraz obsługa innych urządzeń zewnętrznych, których obsługa nie została wbudowana w wybrany „smaku” jądra.

Przy pomocy *dbootstrap* wykonuje się też takie czynności jak partycjonowanie i formatowanie dysku, jak i konfigurację urządzeń sieciowych. Konfiguracja tych elementów systemu odbywa się na początku, ponieważ często jest to konieczne do poprawnego działania dalszej części instalacji systemu.

Dbootstrap jest prostą aplikacją z interfejsem znakowym, zaprojektowaną do działania w różnych sytuacjach. Program ten jest łatwy w użyciu. Można w nim poruszać się przy pomocy klawiszy strzałek, klawisza [Enter] oraz [Tab].

Używanie powłoki podczas instalacji

Podczas procesu instalacji, można przenieść się do drugiej konsoli wirtualnej. W tym celu należy nacisnąć kombinację klawiszy [Lewy Alt]+[F2]. Po zmianie konsoli należy nacisnąć klawisz [Enter], co spowoduje uruchomienie powłoki *ash*. W tym momencie system działa z RAM-dysku i jest dostępny tylko ograniczony zestaw narzędzi.

Aby wykonać wszystkie potrzebne czynności instalacyjne, należy używać odpowiednich pozycji w menu instalacyjnym. Powłoka i komendy są udostępnione tylko na wypadek problemów.

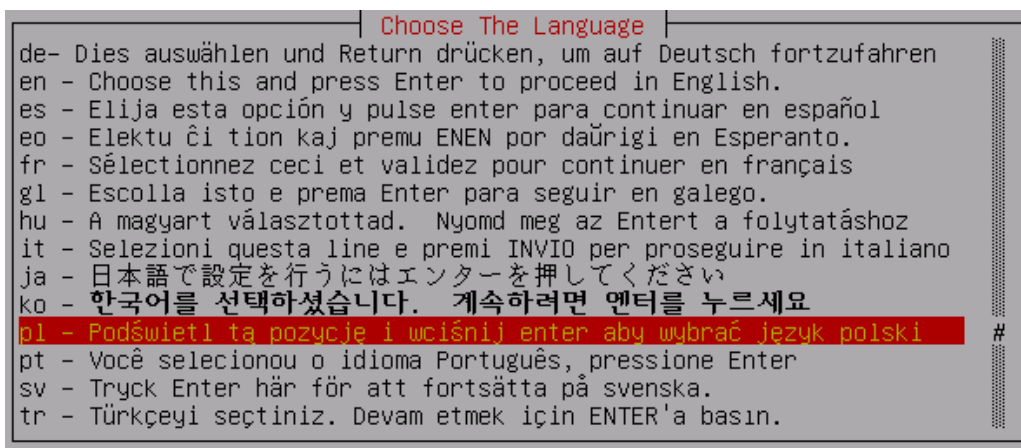
Komunikaty błędów są przekierowywane na trzecią konsolę wirtualną (inaczej *tty3*). Można dostać się na nią naciskając klawisze [Lewy Alt]+[F3]. Komunikaty te można znaleźć w */var/log/messages*, zaś po zakończeniu instalacji plik ten jest kopiowany do */var/log/installer.log* w nowym systemie.

Podczas instalacji systemu podstawowego, komunikaty z procesu rozpakowywania pakietów oraz komunikaty instalatora, można oglądać na konsoli *tty4*. Aby się na nią dostać, należy nacisnąć kombinację klawiszy [Lewy Alt]+[F4], zaś aby powrócić do *dbootstrap* należy użyć kombinacji [Lewy Alt]+[F1].

Wybór języka instalacji

Pierwszym krokiem instalacji jest wybranie języka, w którym będzie kontynuowany proces instalacji systemu.

Odpowiedź na to pytanie, będzie używana do ustawienia domyślnych wartości dla późniejszych ustawień, jak układ klawiatury czy domyślne serwery lustrzane (*mirror server*) dla danej lokalizacji geograficznej.



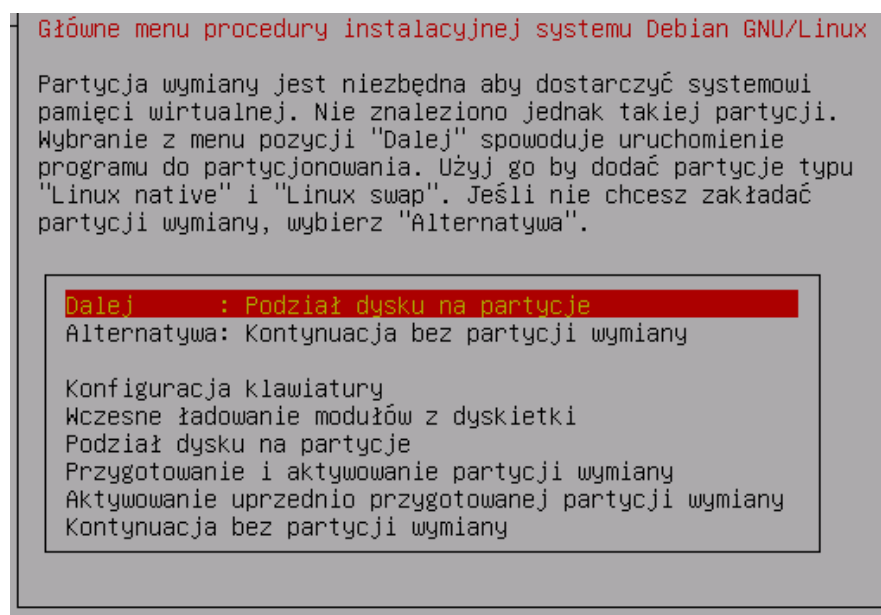
Rys. 4. Wybór języka instalacji

Główne menu procedury instalacyjnej

Program instalacyjny, *dbootstrap*, będzie sprawdzać stan systemu przed każdym krokiem. Dzięki temu można zrestartować instalację, nie tracąc wykonanej już pracy. Jeśli zaistnieje konieczność przerwania instalacji, to przy następnej próbie, konieczne będzie tylko skonfigurowanie klawiatury, ponowne aktywowanie partycji wymiany i ponowne zamontowanie zainicjalizowanych wcześniej partycji. Wszystkie inne czynności, które były wykonane, zostaną zapamiętane i nie trzeba ich będzie powtarzać.

Podczas całej instalacji będzie widoczne główne menu, zatytułowane „Główne menu procedury instalacyjnej systemu Debian GNU/Linux”. Pozycje u góry menu będą zmieniały się, odzwierciedlając postęp w instalacji.

Na pierwszej pozycji w menu instalacyjnym znajduje się czynność, którą powinno się wykonać wnioskując na podstawie tego, co już zostało zrobione. Pozycja ta jest zatytułowana „Dalej”.



Rys. 5. Menu główne procedury instalacyjnej

Konfiguracja klawiatury

W menu wyboru klawiatury należy wybrać klawiaturę odpowiadającą danemu językowi lub podobny typ, jeśli szukany typ nie jest dostępny. Po zainstalowaniu systemu będzie można wybrać klawiaturę spośród większej ilości ustawień (w tym celu należy jako administrator uruchomić program *kbdconfig* po zakończeniu instalacji).

Podział dysku na partycje

Menu „Podział na partycje” przedstawia listę urządzeń dyskowych, które można przepartycjonować i uruchamia program do partycjonowania. Należy utworzyć co najmniej jedną partycję *Linux native* (typ 83) i większości przypadków przynajmniej jedną partycję wymiany *Linux swap* (typ 82).

Dostępna przestrzeń dyskową można użyć jako jedną dużą partycję lub kilka mniejszych, przy czym to drugie rozwiązanie może okazać się bardzo pomocne w niektórych sytuacjach.

Partycja wymiany jest używana jako rodzaj pamięci. Kiedy pamięć RAM nie wystarcza do obsługi uruchomionych aplikacji, część rzadko używanej pamięci może być nagrana na dysk i odczytana z powrotem, gdy będzie potrzebna.

Nazwy dysków i partycji w Linuksie mogą się różnić od nazw w innych systemach operacyjnych. Oto podstawowe przykłady:

- */dev/fd0* – pierwszy napęd dyskietek
- */dev/fd1* – drugi napęd dyskietek
- */dev/hda* – pierwsze urządzenie na pierwszym kontrolerze IDE
- */dev/hdb* – drugie urządzenie na pierwszym kontrolerze IDE
- */dev/hdc* – pierwsze urządzenie na drugim kontrolerze IDE
- */dev/xda* – pierwszy dysk XT
- */dev/xdb* – drugi dysk XT

Linux zwykle używa pojedynczego hierarchicznego systemu plików, zawierającego katalogi i pliki. Każdy katalog może zawierać wiele plików i podkatalogów. Do systemu może być dołączona partycja i będzie ona widoczna we wskazanym pustym katalogu (proces ten nazywa się montowaniem). Każdy katalog zawiera określone typy plików. Ogólna struktura katalogów ma następującą postać:

- */*
Korzeń oznacza punkt startowy hierarchii katalogów.
- */bin*
Katalog z usługowymi programami wykonywalnymi dostępnymi dla wszystkich użytkowników.
- */boot*
Katalog z jądrem systemu i plikami używanymi przez menedżera rozruchu systemu.
- */dev*
Zawiera różne pliki urządzeń, które są interfejsami do różnych części sprzętu.
- */etc*
Katalog z plikami konfiguracyjnymi systemu i poszczególnych programów.
- */home*
Katalog przeznaczony dla danych użytkowników. Każdy użytkownik posiada w tym nim swój podkatalog, którego nazwa odpowiada nazwie użytkownika.
- */lib*
Katalog z bibliotekami systemowymi oraz modułami jądra.
- */mnt*
Używany jako punkt do przyłączenia dodatkowych systemów plików.
- */proc*
Katalog ten, który jest wirtualnym systemem plików, zawiera dane na temat działającego systemu.
- */root*
Katalog domowy użytkownika *root*.
- */sbin*
Programy umożliwiające administrację i konfigurację systemu.
- */tmp*
Katalog dla plików tymczasowych tworzonych przez różne programy.
- */usr*
Ten katalog zawiera wszystkie programy dla użytkowników (*/usr/bin*), biblioteki (*/usr/lib*), dokumentację (*/usr/share/doc*) itp.
- */var*
Katalog dla wszystkich zmieniających się danych, jak poczta elektroniczna, strony WWW, komunikaty systemowe itp.

Do partycjonowania dysku można użyć jednego z dwóch programów: *fdisk* lub *cfdisk*, przy czym domyślnie po wybranie z menu pozycji „Podział dysku na partycje” uruchamia się *cfdisk*, który jest prosty i łatwy w obsłudze. Aby uruchomić *fdisk* należy przejść do drugiej konsoli i wpisać *fdisk* z ewentualnymi argumentami.

```

cfdisk 2.11n

Disk Drive: /dev/hda
Size: 419069952 bytes
Heads: 16 Sectors per Track: 63 Cylinders: 812

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
hda1      Boot      Primary   Linux        349.92
hda2                      Primary   Linux swap   69.16

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Write ]

Toggle bootable flag of the current partition
```

Rys. 6. Przykładowy podział dysku na partycje za pomocą narzędzia *cfdisk*

Przygotowanie i aktywowanie partycji wymiany

Można zainicjalizować i uaktywnić nową partycję lub uaktywnić wcześniej zainicjalizowaną partycję albo w ogóle kontynuować bez tworzenia partycji wymiany. Po wybraniu z menu „Przygotowanie i aktywowanie partycji wymiany”. Domyślnie powinna zostać przedstawiona partycja, która została wcześniej skonfigurowana.

Następnie pojawia się prośba o potwierdzenie, ponieważ inicjalizacja niszczy wszystkie dane uprzednio znajdujące się na tej partycji.

Przygotowanie partycji Linuksa

Można zainicjalizować partycję Linuksa lub zamontować już wcześniej zainicjalizowaną partycję. Natomiast przy pomocy programu *dbootstrap* nie jest możliwe zainstalowanie nowego systemu na starym bez zniszczenia starego systemu.

Po wybraniu z menu pozycji „Przygotowanie partycji Linuksa” nastąpi zainicjalizowanie i zamontowanie pierwszej partycji, która będzie zamontowana jako / (partycja główna).

Następnie system instalacyjny pyta, czy ma sprawdzić twarde dyski. Domyślnie pomija się te testy, ponieważ może zabrać to dużo czasu, a poza tym nowe dyski same wykrywają uszkodzone sektory i radzą sobie z nimi.

Następne okna dialogowe to tylko prośby o potwierdzenie wynikające z tego, że formatowanie partycji niszczy ich zawartość.

Po zamontowaniu partycji głównej, jeśli wcześniej utworzono inne partycje, należy wybrać z menu „Alternatywa” oraz zainicjalizować i zamontować teraz te partycje, podając jednocześnie katalogi, do jakich będą się one odnosić (np. */var*).

Instalacja jądra i modułów systemu operacyjnego

Po wybraniu z menu opcji „Instalacja jądra i modułów sterowników”, wyświetli się menu urządzeń, z których można zainstalować jądro. Można wybrać dowolne z podanych urządzeń – nie trzeba wybierać tego, z którego została uruchomiona instalacja.

Jeśli jądro i moduły mają być zainstalowane przez sieć, można do tego celu użyć opcji „sieć” (HTTP) lub „nfs”. Aby tak zrobić, karta sieciowa podłączona do Internetu, musi być obsługiwana przez standardowe jądro. Po wybraniu opcji „sieć”, należy podać adres internetowy (URL) archiwum Debiana. Wartość domyślna będzie działała dobrze w większości przypadków. Ścieżka zazwyczaj jest taka sama w przypadku wszystkich oficjalnych serwerów lustrzanych Debiana. Można także kazać pobrać pliki przez serwer pośredniczący (proxy).

Konfiguracja modułów – sterowników urządzeń

Aby skonfigurować sterowniki urządzeń, należy wybrać z menu pozycję „Konfiguracja modułów – sterowników urządzeń”. Następnie zostanie uruchomiony program *modconf*, czyli prosty program umożliwiający ładowanie i usuwanie modułów jądra z poszczególnych działów sterowników.

Zalecane jest skonfigurowanie tylko tych urządzeń, które będą niezbędne do przeprowadzenia instalacji, a których jądro nie wykryło samodzielnie.

Niektóre moduły mogą wymagać parametrów. Aby zobaczyć, jakie parametry dotyczą danego modułu, będzie konieczne przeczytanie dokumentacji do danego sterownika.

W każdym momencie po zainstalowaniu systemu, można zmienić konfigurację modułów przy pomocy programu *modconf*.

Konfiguracja sieci

Jeśli system instalacyjny wykryje urządzenie sieciowe, zostanie pokazany krok „Konfiguracja sieci”. Jeśli system nie pozwoli na przeprowadzenie tego kroku, oznacza to, że nie widzi żadnego urządzenia sieciowego.

Jeśli po rozpoczęciu kroku „Konfiguracja sieci” system wykryje więcej niż jedną kartę sieciową, zapyta, którą należy skonfigurować. Podczas instalacji można skonfigurować tylko jedną.

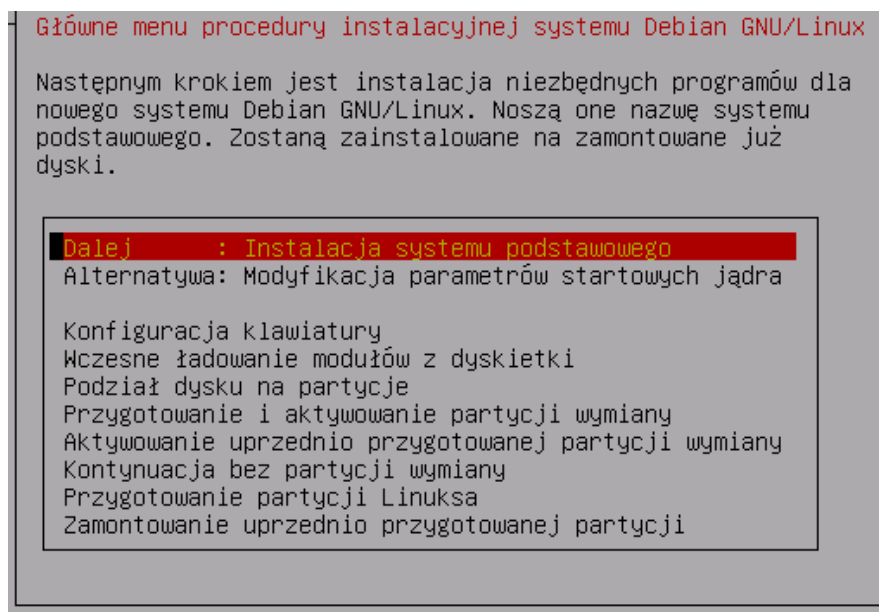
Program instalacyjny następnie pyta, czy do konfiguracji karty sieciowej ma użyć urządzenia BOOTP lub DHCP. Jeśli to możliwe należy wybrać opcję „Tak”. W przeciwnym wypadku, należy ręcznie skonfigurować ustawienia sieciowe.

Ręczna konfiguracja sieci polega na udzieleniu odpowiedzi na kilka pytań, które zada *bootstrap*. Są to pytania, w których wymagane jest podanie takich danych, jak:

- nazwa komputera
- adres IP komputera
- maska sieciowa
- adres bramki
- nazwa domeny
- adres serwera DNS.

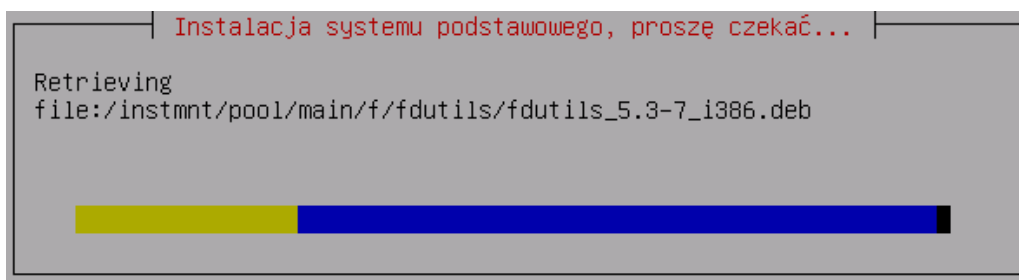
Instalacja systemu podstawowego

Następnym krokiem jest instalacja systemu podstawowego. Jest to minimalny zbiór pakietów, który udostępnia działający, podstawowy, niezależny system. Zajmuje on mniej niż 70 MB.



Rys. 7. Instalacja systemu podstawowego

W przypadku instalacji systemu podstawowego przez sieć, należy zwrócić uwagę na to, że na pasku postępu instalacji przez dłuższy czas może się nic nie zmieniać.



Rys. 8. Postęp pobierania pakietów systemu podstawowego

Przygotowanie systemu do uruchomienia z dysku twardego

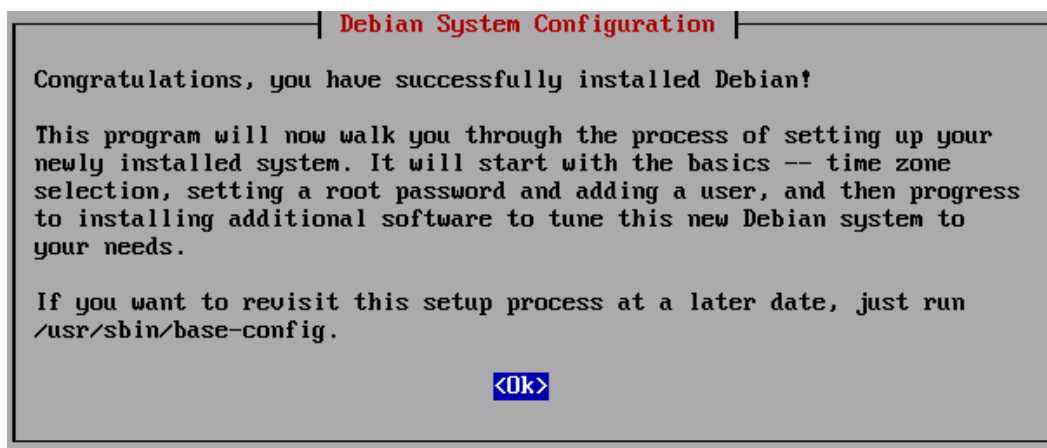
Jeśli zostanie wybrana opcja automatycznego ładowania systemu z dysku twardego, system instalacyjny zapyta o główny sektor ładujący. Jeśli nie jest używany inny program zarządzający ładowaniem, należy odpowiedzieć „Tak”. Standardowym programem ładującym dla architektury i386 jest *lilo*. Jest to skomplikowany program oferujący wiele opcji, takich jak ładowanie systemów DOS, NT i OS/2.

Następne pytanie będzie dotyczyło tego, czy Linux ma być automatycznie z dysku twardego po włączeniu komputera.

Kolejną pozycją w menu jest „Zrestartowanie systemu”. Jeśli wszystko przebiegało pomyślnie, nowy system powinien uruchomić się z dysku twardego.

Początkowa konfiguracja systemu

Po załadowaniu nowego systemu pojawi się prośba o dokończeniu konfiguracji podstawowego systemu i o wybór dodatkowych pakietów do instalacji. Program, który wykonuje te czynności nosi nazwę *base-config*.



Rys. 9. Ekran informujący o pomyślnym zainstalowaniu systemu

Konfiguracja strefy czasowej

Pierwszym etapem konfiguracji systemu po zrestartowaniu, jest wybór strefy czasowej. Najpierw zostanie zadane pytanie o to, czy zegar systemowy jest ustawiony na czas GMT, czy czas lokalny.

Jeśli poza Debianem, na komputerze będzie używany inny system operacyjny (nie UNIX-owy), należy wybrać opcję „Nie”. W przeciwnym wypadku należy wybrać „Tak”.

UNIX zazwyczaj ustawia czas GMT na zegarze systemowym i konwertuje widoczny czas na czas lokalny. Pozwala to systemowi na poprawną zmianę czasu z zimowego na letni i odwrotnie oraz pamiętać o latach przestępnych, a nawet umożliwia użytkownikom zgłoszonym do systemu z innych stref czasowych, samodzielne ustawienie strefy czasowej dla własnego terminala.

Następnie z głównego menu należy wybrać lokalizację geograficzną, podając najpierw kontynent (*Europe*), a potem miasto (*Warsaw*).



Rys. 10. Wybór strefy czasowej

Hasła MD5

Kolejne pytanie dotyczy, czy hasła w systemie mają być szyfrowane przy pomocy algorytmu MD5. Jest to bardziej bezpieczny sposób niż metoda standardowa (zwana „crypt”).

Wartością domyślną jest „Nie”, ale jeśli w systemie nie będzie potrzebna obsługa NIS, a ważne jest bezpieczeństwo, należy wybrać „Tak”.

Obsługa ukrytych haseł

Jeśli wcześniej nie została wybrana obsługa haseł MD5, to system zapyta, czy hasła mają być ukryte (*shadow passwords*). Dzięki temu system jest lepiej zabezpieczony. W przypadku systemu bez ukrytych haseł, hasła są umieszczone (w zaszyfrowanej postaci) w pliku */etc/passwd*, który mogą czytać wszyscy użytkownicy. Ten plik musi być czytelny dla wszystkich, ponieważ zawiera niezbędne dla użytkowników informacje, na przykład przyporządkowuje numerom użytkowników ich nazwy. Dlatego ktoś może odgadnąć hasła poprzez atak typu „brutalnej siły” (*brute force*) polegający na automatycznym teście wszystkich możliwych kombinacji znaków w hasle.

Jeśli hasła są ukryte, to system przechowuje je w pliku */etc/shadow*, który może czytać i modyfikować tylko administrator.

Ustawianie hasła administratora

Konto administratora – *root*, omija wszystkie zabezpieczenia w systemie. Powinno się go używać wyłącznie do czynności administracyjnych i przez jak najkrótszy czas.

Hasło powinno mieć długość przynajmniej 6 znaków i powinno zawierać zarówno duże, jak i małe litery, a także znaki przestankowe. Przy ustawianiu hasła administratora należy zwrócić szczególną uwagę, ponieważ użytkownik tego konta posiada wszystkie uprawnienia.

Tworzenie zwykłego użytkownika

Następnie system pyta, czy ma utworzyć konto zwykłego użytkownika. Powinno ono być zwykłym kontem do użytku codziennego. Nie powinno się używać konta administratora jako konta osobistego.

Nazwa użytkownika nie powinna zawierać polskich znaków diakrytycznych.

Konfiguracja programu apt

Najczęstszym sposobem instalacji pakietów w systemie jest użycie programu *apt-get* z pakietu *apt*. Jednak, aby użyć *apt*, należy go wcześniej skonfigurować, aby wiedział skąd ma pobrać pakiety. Można do tego użyć pomocniczego programu *apt-setup*.

Konfiguracja narzędzia *apt* polega na podaniu miejsc, w których może znaleźć pakiety do instalacji. System poprosi o podanie źródła pakietów: FTP, HTTP, CD-ROM lub lokalny system plików.

Należy pamiętać, że można mieć kilka źródeł *apt*, nawet dla tego samego archiwum Debiana. Program *apt-get* automatycznie wybierze pakiety z najwyższym numerem wersji z wszystkich dostępnych. Jednak nie jest dobrze dodawać zbyt wielu źródeł w sieci, ponieważ wydłuży to czas sprawdzania nowych wersji.

Następnie system zapyta, czy *apt* ma używać uaktualnień związanych z bezpieczeństwem ze strony <http://security.debian.org/>. W przypadku posiadania dostępu do Internetu, należy odpowiedzieć twierdząco na to pytanie.

ZARZĄDZANIE PAKIETAMI

Pakiet to archiwum zawierające programy, dokumentację lub biblioteki. Spełniają one jednak więcej funkcji niż tylko proste archiwum danych. Każdy bit informacji potrzebny do zarządzania oprogramowaniem, które zawiera pakiet, ma swoje właściwe miejsce. Obejmuje to skrypty instalacyjne, rekordy danych potrzebne do odinstalowania itp. W ten sposób procesy instalacji, konfiguracji oraz odinstalowania są spójne dla każdego programu znajdującego się w systemie.

Skomplikowanym elementem zarządzania programami są związki poszczególnych pakietów oprogramowania pomiędzy sobą. Są one znane jako zależności (*dependencies*), gdzie poprawne działanie jednego programu powoduje konflikt z innymi. W systemie powstaje wiele różnego rodzaju zależności, kiedy dany pakiet wymaga konfiguracji systemu we właściwy sposób, zanim będzie możliwa jego poprawna praca. W czasie ręcznego budowania komponentów systemu (z kodu źródłowego), zależności wymagają poszukiwania nowszych wersji różnych części oprogramowania, z których niektóre mają swoje własne zależności.

Dystrybucja Debian dostarcza wszystkie swoje pakiety w formie gotowej do natychmiastowego wykorzystania. System zarządzania pakietami dba o instalację oprogramowania, jego konfigurację itd. System zarządzania pakietami zawarty w dystrybucji Debian zapewnia dla każdego z pakietów następujące ułatwienia:

- spójny interfejs instalacji oprogramowania,
- domyślne konfiguracje lub łatwe w użyciu narzędzia konfiguracyjne,
- ochronę własnej konfiguracji dokonanej przez administratora,
- informacje o zależnościach,
- rekordy potrzebne do odinstalowania.

Wszystkie pakiety, które wymagają konfiguracji są skonfigurowane wstępnie w taki sposób, że dla większości użytkowników są gotowe do wykorzystania bez potrzeby tworzenia własnej konfiguracji.

Program *dpkg* w przeciwieństwie do innych programów zarządzania pakietami nigdy nie zastępuje konfiguracji administratora. Wszystkie dystrybucje oparte na programie *rpm* po prostu zamieniają tę konfigurację.

Informacje o zależnościach

Pakiety dystrybucji Debian zawierają informacje o zależnościach podobnie, jak większość znanych systemów zarządzania pakietami. Jest to niezwykle ważne, ponieważ musi istnieć sposób przekazywania informacji o tym, czy określony pakiet będzie działał po jego zainstalowaniu. W dystrybucji Debian odbywa się to w ten sposób, że są sprawdzane zależności pakietu oprogramowania w stosunku do oprogramowania, które jest już zainstalowane. Program *dpkg* z kolei sprawdza, czy dla instalacji pakietu są spełnione wszystkie warunki wstępne.

System pakietów Debiana używa kilka rodzajów zależności między pakietami, które określają, czy pakiet A może działać niezależnie od obecności w systemie pakietu B. Istnieją następujące typy zależności:

- Pakiet A zależy (*depends*) od pakietu B, jeśli może on działać tylko wtedy, gdy zainstalowany jest pakiet B. W niektórych przypadkach, pakiet A zależy nie tylko od pakietu B, ale od jego wersji.
- Pakiet A zaleca (*recommends*) pakiet B, jeśli opiekun pakietu zdecydował, że większość użytkowników nie będzie chciało pakietu A bez funkcjonalności, którą daje pakiet B.
- Pakiet A proponuje (*suggests*) pakiet B, jeśli pakiet B zawiera pliki związane z funkcjonalnością pakietu A (i zwykle ją zwiększające).
- Pakiet A powoduje konflikt (*conflicts*) z pakietem B, jeśli pakiet A nie może działać, gdy w systemie jest zainstalowany pakiet B. Najczęściej konflikty mają miejsce wtedy, gdy pakiet A zawiera pliki poprawiające pliki z pakietu B.
- Pakiet A zastępuje (*replaces*) pakiet B, gdy pliki zainstalowane przez pakiet B są usuwane i nadpisywane (w niektórych przypadkach) przez pliki z pakietu A.
- Pakiet A dostarcza (*provides*) pakiet B, jeśli wszystkie pliki i funkcjonalność pakietu B jest zawarta w pakiecie A. Dzięki temu mechanizmowi użytkownicy z ograniczoną przestrzenią dyskową pobierają tylko tę część pakietu A, której naprawdę potrzebują.

Budowa pakietów binarnych

Pakiety binarne dla dystrybucji Debian można znaleźć między innymi w ośrodku FTP dystrybucji Debian <ftp://ftp.pl.debian.org/debian>. Zazwyczaj pakiety te mają nazwę w postaci: *pakiet_wersja-wydanie.deb*, gdzie *pakiet* jest nazwą pakietu, *wersja* oznacza jego wersję, a *wydanie* stanowi numer korekty tego pakietu w bieżącej dystrybucji Debiana.

Pakiety binarne to proste, dwupoziomowe archiwa. Pliki *.deb* są archiwami programu *ar* zawierającymi dwa archiwa programu *tar* skompresowane za pomocą programu *gzip*. Jedno z nich zawiera informacje na temat pakietu, natomiast drugie zawiera pliki przeznaczone do umieszczenia w systemie plików w czasie instalacji pakietu.

Budowa pakietów źródłowych

Pakiety źródłowe składają się z kilku elementów. Pierwszy to oryginalne archiwum źródłowe programu *tar* identyczne (w większości przypadków) dla całego strumienia drzewa źródłowego. Jego nazwa to *pakiet-wersja.orig.tar.gz*. Korekty wykonane w kodach źródłowych potrzebne do utworzenia pakietów (jak skrypty kompilacji pakietu oraz poprawki) są umieszczone w pliku *pakiet-wersja_wydanie.diff.gz*. Informacje o pakiecie potrzebne przed rozpakowaniem kodów źródłowych są umieszczone w pliku *pakiet_wersja.dsc*.

Po wydobyciu wszystkich trzech plików z jednego pakietu źródłowego, archiwum może zostać rozpakowane za pomocą polecenia:

```
dpkg -source -x <pakiet.wersja.dsc>
```

Na dysku zostanie utworzony katalog o nazwie *pakiet-wersja*. Ten katalog zawiera standardowe drzewo źródłowe. Jedyne informacje specyficzne dla dystrybucji Debian są umieszczone w podkatalogu *debian/*. Ten katalog z kolei zawiera wszystkie skrypty potrzebne do kompilacji oprogramowania i danych z kodów źródłowych oraz utworzenia dla nich pakietów binarnych.

Pakiety źródłowe nie zawierają informacji o zależnościach. Zawierają je tylko wynikowe pakiety binarne.

Organizacja archiwum lustrzanego dystrybucji

Debian został opracowany dla kilku różnych architektur. Niektóre z nich posiadają dystrybucje stabilne (*stable*). Poniżej tego poziomu, zawsze jest nowa dystrybucja, posiadająca bardziej aktualne oprogramowanie. Tę opracowywaną dystrybucję określa się jako niestabilną (*unstable*). W końcu, kiedy dystrybucja niestabilna jest poddawana specjalnym testom przygotowującym ją do nowego wydania, jej nazwa jest zmieniana na zamrożoną (*frozen*). W każdym momencie można wybrać spośród trzech dystrybucji: *stable*, *frozen* i *unstable*.

Poniżej poziomu stabilności znajduje się kategoria dystrybucji z punktu widzenia licencji. Każdy element oprogramowania wchodzący w skład dystrybucji Debian jest odpowiednio sklasyfikowany i umieszczony w jednej z wymienionych niżej kategorii. Większość oprogramowania wchodzącego w skład systemu Debian należy do katalogu *main*, wszystko co znajduje się w tej części jest całkowicie zgodne z zasadami DFSG (*Debian Free Software Guidelines*). Wszystko, co jest udostępniane nieodpłatnie, ale posiada zależności z pakietami komercyjnymi, jest umieszczone w dystrybucji nazwanej *contrib*. Wreszcie oprogramowanie, które nie spełnia reguł DFSG jest umieszczone pod etykietą *non-free*. Całkowicie oddzielnie od pozostałych występuje jedna, specjalna dystrybucja. Jest nią dystrybucja *non-US*. W tej kategorii pakiety mogą być darmowe lub komercyjne, ale amerykańskie prawo eksportowe zabrania ich eksportu.

Program dpkg

Program *dpkg* jest prostym narzędziem wiersza poleceń, który potrafi wykonać podstawowe operacje dotyczące pojedynczych pakietów. Nie potrafi obsługiwać pakietów, które znajdują się w różnych miejscach, a tylko te, które znajdują się w lokalnym systemie plików.

Polecenie *dpkg* można uruchomić z 2 poziomów. Wiele opcji informacyjnych o pakietach jest dostępnych dla nieuprzywilejowanych użytkowników. Dla zadań takich, jak instalacja czy kasowanie pakietów, wymagane są przywileje administratora.

Aby poinformować program *dpkg*, jaka operacja ma być wykonana, należy użyć jedną z opcji (zwanymi flagami działania). Ogólna postać polecenia *dpkg* ma postać:

```
dpkg -<flaga> <pakiety>
```

Opcje programu *dpkg*

Rodzaj opcji	Flaga	Działanie
Instalacyjne i kasujące	install	Rozpakowuje i instaluje wymieniony pakiet
	unpack	Rozpakowuje wymieniony pakiet
	configure	Konfiguruje wymieniony zainstalowany pakiet
	remove	Usuwa wymieniony zainstalowany pakiety
	purge	Usuwa wymieniony pakiet wraz z plikami konfiguracyjnymi
Zarządzania pakietami	clear-avail	Czyści informacje o liście dostępnych pakietów
	avail	Dodaje informacje o danym pakiecie do listy dostępnych pakietów
	update-avail	Zastępuje stare informacje o liście pakietów na nową listę
	merge-avail	Dodaje całe drzewo do listy dostępnych pakietów
Informacji o pakietach	status	Wyświetla status wymienionego pakietu
	print-avail	Wyświetla opis wymienionego pakietu
	listfiles	Wyświetla listę plików zainstalowanych z wymienionego pakietu
	list	Wyświetla status, numer wersji i krótki opis danego pakietu
	search	Wyświetla katalog i pakiet źródłowy, z którego pochodzi dany plik
	audit	Szuka uszkodzeń w zainstalowanych pakietach
Wymuszające	downgrade	Instaluje pakiet starszy niż obecnie zainstalowany
	configure-any	Konfiguruje wszystkie, mimo iż nie zostały skonfigurowane inne pakiety, od których zależą te pierwsze
	remove-reinstreq	Usuwa pakiet, nawet jeśli został on oznaczony jako wymagany w systemie
	remove-essential	Usuwa dane pakiety, choć są one niezbędne dla poprawnego działania systemu
Różne	print-architecture	Wyświetla architekturę maszyny
	print-gnu-build-architecture	Wyświetla architekturę maszyny w formacie GNU
	license	Wyświetla informacje o prawach autorskich
	version	Wyświetla wersję programu <i>dpkg</i>
	help	Wyświetla listę opcji i ich opis programu <i>dpkg</i>

Tab. 4. Najważniejsze flagi programu *dpkg*

Program dselect

Program *dselect* jest jednym z systemów zarządzania pakietami o największych możliwościach. Zawiera on bardzo rozbudowany system pomocy, zwięzły mechanizm wyświetlania i wyboru pakietów, a także oferuje wiele metod instalacji. Do wad programu można zaliczyć złożoność systemu, a także brak wyszukanego interfejsu użytkownika.

Menu główne

Po uruchomieniu programu *dselect*, wyświetlane jest proste menu, które zawiera siedem pozycji:

- **Dostęp**
Pozwala na wybranie metody dostępu do plików z pakietami.
- **Aktualizacja**
Pozwala na uaktualnienie listy dostępnych pakietów.
- **Wybór**
Pozwala na wybranie pakietów do zainstalowania lub do usunięcia.
- **Instalacja**
Pozwala na zainstalowanie wybranych pakietów.
- **Konfiguracja**
Pozwala na skonfigurowanie zainstalowanych pakietów
- **Skasowanie**
Pozwala na usunięcie wybranych pakietów.
- **Wyjście**
Wychodzi z programu.

```
Debian GNU/Linux `dselect`, nakładka na program zarządzania pakietami.
* 0. [D]ostęp      Wybór metody dostępu.
1. [A]ktualizacj   Zaktualizowanie informacji o dostępnych pakietach.
2. [W]ybór        Wybór pakietów instalowanych w systemie
3. [I]nstalacja    Instalacja i uaktualnienie wybranych pakietów.
4. [K]onfiguracj   Konfiguracja pakietów, które pozostały nieskonfigurowane.
5. [S]kasowanie    Skasowanie niechcianych pakietów.
6. [W]yjście      Wyjście z dselect.

Użyj ^P oraz ^N, klawiszy kursora, pierwszych liter lub cyfr;
Wciśnij ENTER dla potwierdzenia wyboru.  ^L przerysowanie ekranu.

Wersja 1.6.15 (i386). Copyright 1994-1996 Ian Jackson.
Niniejszy program jest oprogramowaniem wolnodostępnym, rozpowszechnianym
na warunkach Powszechnej Licencji Publicznej GNU wersji 2-giej tej Licencji
lub którejś z późniejszych wersji. Brak JAKIEJKOLWIEK gwarancji.
Więcej szczegółów zobacz w dselect --licence.
```

Rys. 11. Główne menu programu *dselect*

Wybór metody dostępu

Aby wybrać metodę dostępu, należy użyć klawiszy strzałek do podświetlenia pozycji menu [D]ostęp i naciskając [Enter]. Pojawi się wtedy ekran, na którym wyświetli się lista metod dostępu. Należy wybrać jedną z nich.

Z zasady, metody dostępu programu *dselect* zawsze instalują pakiety w sposób posortowany, bez zwracania uwagi na zależności. W wyniku tego, czasami powstają błędy. Dzieje się tak w przypadku, gdy program *dselect* próbuje zainstalować pakiety, dla których zależności nie zostały spełnione. Po napotkaniu pewnej liczby błędów program *dselect* przerywa działanie. Z tego powodu czasami konieczne jest kilkakrotne wznowianie instalacji pakietów, zanim uda się ją zakończyć.

Zaleca się skorzystanie z metody *apt*, która jest najbardziej wszechstronna. Zapewnia ona równoległą obsługę instalacji z sieci oraz obsługę zależności, która umożliwia dokonywanie uaktualnień i instalacji bez zwracania uwagi, czy wszystkie zależności są spełnione.

Wybór pakietów

Przeglądarka listy pakietów wyświetla w usystematyzowany sposób listę wszystkich pakietów dostępnych do instalacji. Przeglądarka pakietów obsługuje zarówno bazę danych dostępności pakietów, jak też bazę danych stanu systemu. Baza danych stanu systemu przechowuje informacje na temat bieżącego profilu systemu, to jest kontroluje, jakie pakiety są zainstalowane, czy którykolwiek ze skryptów instalacyjnych nie działa itp. Baza danych zawiera również zaległy profil systemowy (*pending system profile*), który zawiera informacje o żądanych zmianach do instalacji oprogramowania. Baza danych systemu jest przechowywana w pliku */var/lib/dpkg/status*.

```
dselect - lista pakietów (wg dostępności)      wybór:+=/- szczegóły:v pomoc:?
EIDM Pri Sekcja Pakiet      Zainst.      Dostępne      Opis
---
___ Opc games      gnome-gnomet <brak>      1.2.0-helix A tetris clone.
___ Opc games      gnome-gnomin <brak>      1.2.0-helix Classic find the mines in the
___ Opc games      gnome-gnotra <brak>      1.2.0-helix A game based on Tetravex.
___ Opc games      gnome-gtali <brak>      1.2.0-helix Gnome version of Yahtzee Dice
___ Opc games      gnome-gturin <brak>      1.2.0-helix Turing game
___ Opc interpre gnome-guile <brak>      1.0.1.cvs.1 Guile-Gtk scheme interpreter
___ Opc graphics gnome-gv <brak>      0.95-helix3 GNOME PostScript/PDF viewer
___ Opc x11         gnome-help <brak>      1.2.4-helix GNOME help browser
___ Opc x11         gnome-help-d <brak>      1.2.4-helix GNOME help browser data
___ Opc games      gnome-iagno <brak>      1.2.0-helix Gnome version of Othello (Rev
___ Opc x11         gnome-iconed <brak>      1.2.0-helix Icon editor for the GNOME Des
___ Opc x11         gnome-libs-d <brak>      1.2.11-ximi Data for Gnome libraries
___ Opc games      gnome-mahjon <brak>      1.2.0-helix Classic Chinese Tile Game (fr
gnome-guile nie zainstalowany ; wyczyszczony (był: wyczyszczony). Opcjonalne
gnome-guile - Guile-Gtk scheme interpreter (part of Gnome)

Gnome is the "GNU Network Object Model Environment"

It is a project to build a complete, user-friendly desktop based entirely on
free software.

This package contains the guile-gtk and gnomeg scheme interpreters (which
contain hooks for the Gtk widget library and the Gnome application framework)

opis pakietu gnome-guile
```

Rys. 12. Przeglądarka listy pakietów programu *dselect*

Górna połowa ekranu przeglądarki listy pakietów zawiera rekordy opisu pakietów – po jednym w każdym wierszu. Każdy pakiet ma swoją nazwę, krótki opis zawartego w nim oprogramowania oraz informacje na temat bieżącego stanu instalacji, a także priorytet oprogramowania i przypisaną mu kategorię. Dolna połowa ekranu przedstawia bardziej szczegółowe informacje na temat aktualnie wybranego pakietu (lub kategorii pakietów). Pierwszy wiersz jest streszczeniem i powinien dawać podstawową informację na temat przeznaczenia oprogramowania. Pozostała część tego opisu jest bardziej szczegółowa. Jeżeli nie wszystkie informacje mogą się zmieścić w tej części ekranu, można wcisnąć *d* w celu przewijania w dół oraz *u* w celu przewijania w górę. Wiersz podziału dwóch części ekranu wyświetla nazwę pakietu, kilka informacji na temat stanu oraz określenie poziomu ważności dla użytkownika.

Kolumna EIOM w programie *dselect* jest odczytem informacji z profilu systemowego. Każda z tych czterech liter reprezentuje własną kolumnę:

- **E – flaga błędu** (*error flage*)

Jeśli pakiet ma zaznaczoną tę kolumnę, oznacza to, że instalacja nie powiodła się. Jeżeli w tej kolumnie znajduje się litera R, oznacza to potrzebę ponownego zainstalowania pakietu.

- **I – stan instalacji** (*installed state*)

Ta kolumna reprezentuje bieżący stan instalacji. Jeżeli jest tam wyświetlony znak podkreślenia, oznacza to, że pakiet nie został zainstalowany. Gwiazdka mówi, że pakiet jest zainstalowany całkowicie, a myślnik wskazuje, że istnieją pliki konfiguracyjne z poprzedniej instalacji tego pakietu. Litera U informuje, że pakiet został rozpakowany, ale ciągle wymaga konfiguracji. Litera C oznacza, że w czasie konfiguracji wystąpił błąd, zaś litera I, że pakiet nie mógł być właściwie rozpakowany.

- **O – poprzedni znacznik** (*old mark*)

Ta kolumna pokazuje informacje pochodzące z zaległego profilu systemowego, zanim został uruchomiony program *dselect*. Jest to pomocne z tego względu, że można odnieść się do poprzednich ustawień nawet wtedy, gdy zostanie zmieniona bieżąca konfiguracja.

- **M – znacznik** (*mark*)

Kolumna ta pokazuje informacje pochodzące z zaległego profilu systemowego bez uwzględnienia zmian wykonanych w bieżącej sesji programu *dselect*. Za każdym razem, kiedy uruchamiany jest program *dselect*, kolumny poprzedniego znacznika (O) i znacznika (M) są identyczne. W przypadku zaznaczenia nowych pakietów do zainstalowania lub do usunięcia, zmiany te są odzwierciedlane właśnie w tej kolumnie. W momencie wyjścia z przeglądarki kolumna znacznika jest zapisywana do zaległego profilu systemowego. Celem takiego sposobu prezentacji zmian jest możliwość łatwego cofnięcia zmian dokonanych przez pomyłkę.

Wybór oprogramowania, które należy zainstalować jest zadaniem trudnym. Dlatego pakiety są klasyfikowane według rodzaju funkcji, jakie spełniają oraz poziomu ważności dla przeciętnego użytkownika. Przeglądarka listy pakietów wykorzystuje do wyświetlania tych informacji kolumny **Pri** (priorytet) oraz **Section** (kategoria).

Kategoria	Opis
admin	Narzędzia administracji systemu i sieci
base	Oprogramowanie stanowiące minimum dla właściwej pracy systemu
comm	Oprogramowanie szeregowego portu komunikacyjnego
devel	Narzędzia programistyczne
doc	Dokumentacja
editors	Różnego rodzaju edytory tekstowe
games	Gry
graphics	Edytory multimedialne oraz przeglądarki
hamradio	Programy dla operatorów amatorskich sieci radiowych
interpreters	Języki skryptowe
libs	Różnego rodzaju biblioteki dla języka C/C++
mail	Programy do odczytywania poczty, serwery pocztowe, filtry
math	Narzędzia matematyczne: kalkulatory naukowe, edytory równań
misc	Różne narzędzia nie pasujące do innych kategorii
net	Aplikacje sieciowe, serwery i narzędzia
news	Programy do odczytywania list dyskusyjnych, serwery itp.
oldlibs	Biblioteki potrzebne dla zachowania zgodności ze starszymi systemami
oterosfs	Narzędzia potrzebne do współdziałania z innymi systemami operacyjnymi
shells	Powłoki
sound	Edytory różnych formatów dźwięku, odtwarzacze dźwięku, kodery itp.
tex	Popularny program do składu drukarskiego
text	Inne narzędzia do składu
utils	Narzędzia programowe do wykonywania określonych zadań
web	Narzędzia potrzebne do WWW
x11	Aplikacje graficzne dla systemu X Window

Tab. 5. Kategorie pakietów używane przez program *dselect*

Domyślna hierarchia (poczynając od niezbędnych, kończąc na dodatkowych) priorytetów pakietów przedstawia się następująco:

- required
- important
- standard
- optional
- extra.

Często próba zainstalowania lub usunięcia jednego pakietu narusza środowisko pracy innego. Niektóre programy dla poprawnej pracy wymagają dostępności do innych i z tej przyczyny powstają zależności pomiędzy pakietami. W efekcie, czasem nie można zainstalować jednego pakietu, bez zainstalowania drugiego. Kiedy zostanie napotkany taki konflikt, pojawi się ekran rozwiązywania zależności.

```

dselect - lista konfliktów                               wybór:+=/- szczegóły:v pomoc:?
EIDM Pri Sekcja Pakiet Opis
_* Opc interpre gnome-guile Guile-Gtk scheme interpreter (part of G
_* Opc libs libguilegtk0 Libraries for Guile-Gtk scheme interpreter (part of G
_* Opc graphics gdk-implib1 Gdk-Imlib is an imaging library for use with gtk
_* Opc libs libart2 The Gnome canvas widget
_* Opc libs libaudiofile The Audiofile Library
_* Opc libs libesd0 Enlightened Sound Daemon - Shared libraries
_* Opc libs libesd-alsa0 Enlightened Sound Daemon (ALSA) - Shared libraries
_* Opc libs libgnome32 The Gnome libraries
_* Opc libs libgnomesupp The Gnome libraries (Support libraries)
_* Opc libs libgnomeui32 The Gnome libraries (User Interface)
_* Opc libs libguile6 `libguile.so.6' shared libraries for Guile1.3.
_* Opc libs libgnorba27 Gnome CORBA services
_* Opc libs liborbit0 Libraries for ORBit - a CORBA ORB
gnome-guile nie zainstalowany ; instalacja (był: wyczyszczony). Opcjonalne
gnome-guile zależy od libguilegtk0 (= 1.0.1.cvs.19991112-2)
gnome-guile zależy od gdk-implib1 (>= 1.7)
gnome-guile zależy od libart2 (>= 1.0.56-2)
gnome-guile zależy od libaudiofile0
gnome-guile zależy od libesd0 (>= 0.2.16) lub libesd-alsa0 (>= 0.2.16)
gnome-guile zależy od libgnome32 (>= 1.0.56-2)
gnome-guile zależy od libgnomesupport0 (>= 1.0.56-2)
gnome-guile zależy od libgnomeui32 (>= 1.0.56-2)
gnome-guile zależy od libguile6 (>= 1.3.4-2)

relacje dotyczące pakietu gnome-guile

```

Rys. 13. Ekran rozwiązywania zależności programu *dselect*

Istnieją dwa sposoby wyjścia z dowolnych okien dialogowych w przeglądarce. Można albo sprawdzić zależności i rozwiązać wszystkie problemy (wcisnięcie [Enter] w dowolnym z okien dialogowych), albo po prostu opuścić dialog bez sprawdzania tych zależności (wcisnięcie klawisza q). Zalecane jest stosowanie pierwszego sposobu.

Instalacja, konfiguracja i usuwanie pakietów

Kiedy zostanie wybrana opcja [I]nstalacja z menu głównego programu *dselect*, spowoduje to automatyczne wyszukiwanie pakietów zaznaczonych do instalacji lub do uaktualnień i próbę pobrania ich z nośnika wybranego za pomocą polecenia [D]ostęp. Następnie program *dselect* rozpakuje te pakiety i skonfiguruje wszystkie za jednym razem. Niektóre z pakietów powodują wyświetlanie pewnych pytań, ale większość tego nie wymaga.

Jeżeli ostatnim komunikatem, jaki otrzymaliśmy w programie *dselect*, jest informacja, że program *dpkg* nie zdołał zainstalować pewnych pakietów, uruchomienie konfiguracji często rozwiąże ten problem. W tej czynności następuje ponowna próba konfiguracji tych z nich, których automatyczna konfiguracja w fazie instalacji nie powiodła się.

Dowolne pakiety zaznaczone do usunięcia lub wyczyszczenia są odpowiednio obsługiwane po wybraniu polecenia [S]kasowanie z głównego menu programu *dselect*.

Program apt

Ponieważ program *apt* nie posiada mechanizmu przeglądania i wyboru pakietów (wszystko odbywa się z wiersza poleceń), jest często wykorzystywany w połączeniu z programem *dselect*, gdzie jest wymieniony jako metoda dostępu.

Główną cechą programu *apt* jest obsługa zależności. Kiedy zostanie wybrany pakiet do instalacji, *apt* automatycznie instaluje go wraz z innymi, potrzebnymi do uwzględnienia wszystkich zależności pakietami. Dodatkowo, program *apt* instaluje pakiety w kolejności dyktowanej przez zależności, unikając problemów, właściwych dla programu *dselect*.

W zasadzie, kiedy jeden pakiet zależy od drugiego, wymaga on, żeby ten drugi był skonfigurowany w pierwszej kolejności. Każdy menedżer pakietów instalujący dużą liczbę pakietów w tym samym czasie powinien porządkować je zgodnie z zależnościami w taki sposób, aby uniknąć błędów instalacji. W przeszłości metody stosowane przez program *dselect* nie były tak inteligentne. Zamiast tego wykonywały one instalację pakietów w wybranej przez siebie kolejności bez zwracania uwagi na zależności. Powstawało masę błędów, które czasem nakładały się na siebie i w efekcie program *dselect* przerywał pracę nie dokończając instalacji. Taki sposób pracy zmusza administratora systemu do ponownego uruchamiania faz konfiguracji i instalacji, zanim system będzie mógł działać poprawnie.

Taka sytuacja nie występuje, jeżeli używany jest program *apt*. Posiada on funkcję rozpoznawania kolejności, w jakiej pakiety powinny być zainstalowane. W każdym przypadku potrzebne jest tylko jedno uruchomienie programu *apt* w celu instalacji lub uaktualnienia dowolnej liczby pakietów. Jest to rozwiązanie, którego nie ma w innych systemach operacyjnych ani też w innych dystrybucjach systemu Linux.

Konfigurowanie programu apt

Wstępna konfiguracja systemu *apt* odbywa się podczas automatycznej wstępnej konfiguracji systemu.

Program *apt* potrafi skorzystać z dowolnego ośrodka WWW lub FTP. Źródła te są przechowywane w pliku */etc/apt/sources.list*. Każdy wiersz pliku */etc/apt/sources.list* programu *apt* jest nazwą archiwum, gdzie znajdują się pakiety. Są one w postaci:

```
deb <URI> <rodzaj stabilności> <kategoria>
```

gdzie URI (*Universal Resource Indicator*) oznacza miejsce, skąd należy ściągnąć pakiety, rodzaj stabilności określa kategorię stabilności, a kategoria oznacza rodzaj kategorii (spośród *main*, *contrib*, *non-free* oraz *non-us*).

Program *apt*, dla rozszerzenia pasma, dokonuje równoległego pobierania plików, kiedykolwiek jest to możliwe. Czasem może wystąpić sytuacja, że w dwóch serwerach lustrzanych będą występować różne wersje tego samego pakietu. Program *apt* przydziela pierwszemu serwerowi najwyższy priorytet. W przypadku różnic wersji, zawsze jest pobierana wersja z serwera wymienionego jako pierwszy.

Używanie programów z rodziny *apt*

Na rodzinę *apt* składa się kilka programów. Najczęściej używanymi są jednak *apt-get* oraz *apt-cache*.

Program	Polecenia	Działanie
apt-get	install	Pobiera i instaluje wszystkie pakiety wymienione w wierszu poleceń razem ze wszystkimi pakietami potrzebnymi do spełnienia zależności.
	remove	Usuwa zainstalowane pakiety wymienione w wierszu poleceń razem ze wszystkimi pakietami potrzebnymi do spełnienia zależności.
	update	Pobiera wszystkie bazy danych pakietów z każdego dostępnego źródła, modyfikując informacje przechowywane lokalnie tak, aby odpowiadały informacjom występującym w źródłach.
	upgrade	Próbuje zainstalować najbardziej aktualne wersje każdego z pakietów w systemie. To polecenie próbuje wykonać wszystkie czynności bez zmiany stanu instalacji żadnego z pakietów, niezależnie od tego, czy wymagają tego zależności.
	dist-upgrade	Działa podobnie jak <i>upgrade</i> , ale zmienia stan instalacji dodatkowych pakietów, jeśli wymagają tego zależności. Jest to zalecane polecenie dla kompletnego uaktualniania systemu.
	dselect-upgrade	Czyta bazę danych stanu systemu i próbuje uaktualnić system zgodnie z profilem określonym w przeglądarce listy pakietów. W tym trybie <i>apt</i> instaluje pakiety w kolejności wymaganej przez zależności, ale nie zmienia stanu instalacji żadnego z pakietów.
	check	Uaktualnia dane o zbiorze pakietów i sprawdza uszkodzone zależności.
apt-cache	search	Przeszukuje lokalną bazę pakietów w celu odnalezienia pakietów pasujących do podanego wzorca.
	show	Wyświetla szczegółowe informacje o podanym pakiecie.
	stats	Wyświetla statystykę dotyczącą lokalnego bufora pakietów i ich wzajemnych zależności.
	depends	Wyświetla zależności dla podanego pakietu.
	policy	Wyświetla informacje o priorytecie poszczególnych źródeł pakietów. Jeśli w wierszu poleceń podana jest nazwa pakietu, wyświetla informacje o priorytecie danego pakietu.

Tab. 6. Programy z rodziny *apt* i ich najważniejsze polecenia

4. KONFIGURACJA SYSTEMU

Wstępna konfiguracja systemu odbywa się w połączeniu z procesem instalacji. Istnieje jednak wiele rzeczy, które należy skonfigurować już po zakończeniu procesu instalacji, aby system mógł właściwie pracować.

Do podstawowych rzeczy, które wymagają konfiguracji należy konfiguracja procesu *init* i systemu awaryjnego UPS oraz konfiguracja sieci.

PROCES INIT

Proces *init* służy do wykonywania podstawowych, powtarzających się działań w czasie każdego startu systemu. Program *init* posiada profile programowe, które można skonfigurować, nazywane poziomami startu (*run levels*). Poziom startu może być uważany za określony stan oprogramowania systemowego, w którym znajdują się uruchomione programy w danym czasie. Istnieje kilka poziomów startu, które można skonfigurować dla własnych potrzeb. Każdy poziom startu wyznacza całkowicie odmienny zbiór programów.

Domyślnym poziomem startu jest 2., w którym wszystko, co w systemie Debian jest uznawane za istotne, jest uruchamiane domyślnie. Poprzez ręczną konfigurację poziomów startu w określonych celach, można łatwo przeskakiwać pomiędzy określonymi zastosowaniami systemu.

Program *init* jest skonfigurowany wstępnie dla ściśle określonych zachowań w kilku poziomach startu. Wszystkie one odnoszą się do ważnych zadań.

- **Poziom startu 0**

Zamyka wszystkie programy i zatrzymuje pracę komputera. Jeżeli system ma wkompiłowaną obsługę zaawansowanego zarządzania energią, to komputer wyłączy się automatycznie.

- **Poziom startu 1**

Jest zarezerwowany dla trybu jednostanowiskowego. W tym poziomie, nie pracuje nic oprócz pojedynczej powłoki *root*. Nie są zamontowane żadne systemy plików, oprócz systemu *root*. Ten poziom startu jest generalnie wykorzystywany w celach odtwarzania systemu.

- **Poziom startu 2**

Może być konfigurowany i nie ma żadnego zarezerwowanego zastosowania, ale jest to domyślny poziom startu systemu Debian.

- **Poziomy startu 3, 4 i 5**

W systemie Debian nie ma określonego sposobu konfiguracji poziomów startu 3, 4 i 5. Są one otwarte dla celów dostosowania do własnych potrzeb.

- **Poziom startu 6**

Jest bardzo podobny do poziomowi 0, ale zamiast zatrzymania systemu, powoduje on jego ponowne uruchomienie.

Konfiguracja procesu *init*

Oprócz wszystkich skryptów wykorzystywanych przez proces *init*, istnieje jeden plik konfiguracyjny, który steruje większością jego podstawowych zachowań. Plik */etc/inittab* określa pierwszy skrypt, który powinien być uruchomiony, zanim proces *init* przejdzie do jakiegokolwiek poziomu startu. Określa także poziom startu, do którego proces *init* ma przejść po wykonaniu tego skryptu. Następnie określa, jakie polecenia powinny być wykonane w celu uruchomienia poszczególnych poziomów startu.

Główną funkcją pliku */etc/inittab* jest opis tego, które programy powinny być uruchomione w czasie ładowania oraz normalnej pracy w danych poziomach startu. Każdy z poziomów startu może być całkowicie zdefiniowany w pliku */etc/inittab*, ale system Debian posiada w tym celu znacznie bardziej rozbudowany mechanizm o nazwie *sysvinit*.

Plik */etc/inittab* posiada jeden podstawowy typ dyrektyw, który określa wiersz polecenia, działania do wykonania oraz to, w jakim poziomie startu to polecenia powinno być aktywne. Ogólnie format tych wierszy jest następujący:

```
<id>:<poziomy startu>:<działanie>:<polecenie>
```

gdzie *id* jest nazwą, **poziomy startu** to ciąg liczb (odpowiadających poziomom startu), **działanie** opisuje, kiedy należy wykonać polecenie, natomiast **polecenie** określa właściwe polecenie do wykonania (wraz z opcjami).

Flaga	Działanie
respawn	Uruchomienie polecenia i monitorowanie jego wykonania. Kiedy proces zakończy się, następuje ponowne wykonanie polecenia.
wait	Uruchomienie polecenia w każdym z określonych poziomów startu. Proces <i>init</i> oczekuje na zakończenie procesu.
once	Uruchomienie procesu raz w każdym z określonych poziomów startu. Można stosować to polecenie do wykonania rzutu pliku <i>/var/spool/messages</i> na konsolę wirtualną.
boot	Uruchomienie polecenia w czasie ładowania systemu. We fładze tej postaci, poziomy startu są ignorowane.
bootwait	Uruchomienie polecenia podczas ładowania systemu. Proces <i>init</i> oczekuje na zakończenie procesu, po czym wznowia działanie.
off	Wyłączenie polecenia we wszystkich poziomach startu systemu.
initdefault	Informuje, który poziom startu powinien być uruchomiony w czasie ładowania systemu. Pola <poziomy startu> są ignorowane.
powerwait	Uruchomienie polecenia w przypadku awarii zasilania. Proces <i>init</i> czeka na zakończenie tego polecenia.
powerfail	Uruchomienie polecenia w przypadku awarii zasilania. Proces <i>init</i> nie czeka na zakończenie tego polecenia.
powerokwait	Uruchomienie, kiedy zasilanie wraca do normalnego stanu. Proces <i>init</i> wstrzymuje działanie do momentu zakończenia działania tego polecenia.
powerfailnow	Uruchomienie polecenia w przypadku spadku mocy baterii urządzenia UPS.
ctrlaltdel	Uruchomienie określonego polecenia po wciśnięciu klawiszy [Ctrl]+[Alt]+[Del].
kbdrequest	Przyporządkowanie specjalnego działania do określonych klawiszy. W przypadku systemu Debian jest to [Alt]+[↑].

Tab. 7. Flagi działań w pliku */etc/inittab*

Sposób organizacji pliku */etc/inittab* w systemie Debian jest taki, że większa część definicji poszczególnych poziomów startu jest przeniesiona poza ten plik do hierarchii startów. Jedynymi programami, które są uruchamiane bezpośrednio z pliku */etc/inittab*, są programy *getty*, wykorzystywane do uruchamiania komunikatów logowania na urządzeniach wirtualnych terminali. Dzieje się tak dlatego, że wymagają one specjalnej obsługi, która byłaby znacznie trudniejsza do osiągnięcia poza skryptem *inittab*.

Domyślna konfiguracja procesu *init* w systemie Debian prowadzi do ponownego uruchomienia w przypadku naciśnięcia kombinacji [Ctrl]+[Alt]+[Del], zamknięcia systemu w przypadku problemów z zasilaniem oraz uruchomienia programów *getty* we wszystkich wielodostępnych poziomach startu (poziomy startu od 2 do 5).

Podtrzymywanie zasilania (UPS)

W rezultacie problemów z zasilaniem może powstać wiele błędów w każdym komputerze. Ze względu na sposób przechowywania plików i ich zapisywania na dysk w systemie Debian, jest bardzo ważne, aby za każdym razem system był poprawnie zamykany. Jeżeli tak nie jest, może nastąpić awaria dysku. Dodatkowo skoki napięcia oraz jego wahania mogą zniszczyć inne części systemu.

Najlepszym zabezpieczeniem przeciwko problemom z zasilaniem jest urządzenie o nazwie UPS (*Uninterruptible Power Supply*). Są to urządzenia, które mają baterie o dużej pojemności, które pozwalają na kontynuowanie pracy komputera przez krótki czas po zaniku zasilania. Dodatkowo posiadają możliwości sprawdzania zasilania, filtrowania skoków i spadków napięcia.

Większość urządzeń UPS ma również dodatkową przydatną funkcję: zdolność przekazywania informacji do komputera i ostrzegania go o problemach z zasilaniem. Zwykle odbywa się to poprzez przewód szeregowy podłączony z jednej strony do urządzenia UPS, a z drugiej do portu szeregowego komputera. Większość modeli potrafi ostrzec system o awariach zasilania i warunkach wyczerpywania się baterii, a także alarmować system, kiedy zasilanie jest przywrócone.

Zarządzanie UPS w systemie Debian

System Debian zarządza zdarzeniami UPS poprzez *init* – główny proces systemu. Oddzielny proces monitoruje port szeregowy, do którego jest podłączone urządzenie UPS, oczekując na zdarzenie lub otrzymanie danych. W przypadku pojawienia się zdarzenia krytycznego, proces wykorzystuje sygnał SIGPWR w celu zaalarmowania procesu *init* o zmianie statusu urządzenia UPS i zapisuje bieżący status do pliku */etc/ups-status*. Proces *init* sprawdza ten status i uruchamia odpowiedni skrypt: *powerwait* w przypadku pierwszej utraty zasilania, *powerfailnow*, kiedy zaczyna się wyczerpywać bateria oraz skrypt *powerokwait*, kiedy zasilanie powraca do normy.

Te skrypty są zazwyczaj konfigurowane do pracy niemal w taki sam sposób we wszystkich systemach. W sytuacji *powerwait* jest inicjowane zamykanie systemu w ciągu ustalonej liczby minut, zabezpiecza to przed kompletnym zamknięciem systemu

w przypadku chwilowego zaniku zasilania, chroniąc jednocześnie przed całkowitym wyczerpaniem baterii w czasie przedłużającego się zaniku zasilania. Skrypt *powerfailnow* zwykle rozpoczyna proces zamykania systemu natychmiast. Skrypt *powerokwait* zazwyczaj powoduje anulowanie zaplanowanego zamknięcia systemu.

Drzewo dowiązań symbolicznych rc

Podstawowym katalogiem konfiguracji procesu *init* w systemie Debian jest */etc/init.d*. Ten katalog zawiera skrypty, które uruchamiają i zatrzymują ważne programy. Dla uproszczenia zostało ustalone, że wszystkie będą obsługiwały dokładnie jeden argument, jest nim argument **start** lub **stop**.

Jest to bardzo przydatne narzędzie dla użytkownika, ponieważ daje możliwość uruchomienia i zamknięcia skomplikowanych programów, które są dostarczone z systemem Debian.

Są to mechanizmy, dzięki którym można dostroić poziom startu do własnych potrzeb. Dzięki nim program */etc/init.d/rc* może doprowadzić system do stanu, jaki został zaplanowany w danym poziomie startu. W każdym z katalogów */etc/rc<0-6>.d* (jeden dla każdego trybu wielodostępowego) jest grupa dowiązań symbolicznych wskazujących na skrypty w katalogu poziomu startu w logicznej kolejności. Najpierw są uruchamiane wszystkie skrypty, których nazwa rozpoczyna się literą *K* z argumentem **stop** (w porządku numerycznym i alfabetycznym), następnie wykonywane są skrypty, których nazwa rozpoczyna się literą *S* z argumentem **start** (w takim samym porządku).

Adaptacja poziomów startu

Dostosowanie poziomów startu pozwala na dostrojenie procesu startowego, dając większą kontrolę nad tym, co należy uruchomić, a czego nie, i w jakich sytuacjach. Może to przydać się na przykład do konfiguracji zachowania systemu w przypadku awarii zasilania.

Aby dodać wpis uruchamiający lub kończący działanie danego programu należy dokonać wpisu:

```
ln -s /etc/init.d/<skrypt> /etc/rc<0-6>.d/<s|k><0-99><skrypt>
```

Na przykład, aby do piątego poziomu startowego dodać uruchamianie demona Samby jako ostatniego z kolei, należy wykonać polecenie:

```
ln -s /etc/init.d/samba /etc/rc5.d/s99samba
```

KONFIGURACJA SIECI

W systemie Debian podstawowa konfiguracja jest wykonywana w czasie instalacji, a dokładniej w czasie konfigurowania systemu bazowego. Tak jak w innych systemach UNIX, wszystkie dane konfiguracyjne są zapisywane w plikach tekstowych w katalogu */etc*.

Ważną rzeczą jest fakt, że w systemach UNIX inaczej niż w systemach operacyjnych firmy Microsoft, zmianę konfiguracji można przeprowadzić „w locie”. Oznacza to, że większość parametrów można zmienić w czasie działania systemu, bez potrzeby jego ponownego uruchamiania. Ułatwia to eksperymentowanie lub rozwiązywanie problemów konfiguracyjnych.

Pliki konfiguracyjne

Wszystkie pliki konfiguracyjne mogą być modyfikowane w czasie działania systemu. Modyfikacje większości z nich odniosą skutek natychmiast, bez konieczności uruchamiania lub zatrzymywania żadnego z demonów. Większość z plików konfiguracyjnych akceptuje komentarze, które rozpoczynają się od znaku #.

Ważniejsze pliki konfiguracji sieci w systemie Linux to:

- */etc/hostname*
- */etc/hosts*
- */etc/networks*
- */etc/ethers*
- */etc/protocols*
- */etc/host.conf*
- */etc/nsswitch.conf*
- */etc/resolv.conf*
- */etc/network/interfaces*

Plik */etc/hostname*

Plik */etc/hostname* zawiera zazwyczaj tylko jeden wiersz z nazwą hosta. Plik ten jest wykorzystywany w czasie ładowania systemu w celu ustawiania nazwy hosta.

```
# Przykładowy plik /etc/hostname
```

```
#Nazwa hosta  
fraxinus
```

Plik */etc/hosts*

Plik */etc/hosts* zawiera odwzorowanie pomiędzy adresami IP a bardziej przyjaznymi dla użytkownika nazwami hostów. Adresy IP zostały opracowane z myślą o ułatwieniu trasowania pakietów IP w sieci przez komputery, ale dla ludzi są one trudne do zapamiętania.

Plik */etc/hosts* oprócz nazw hostów może zawierać także aliasy do nich. Odniesienie się do nazwy aliasu ma taki sam efekt, co odniesienie się do nazwy hosta.

```
# Przykładowy plik /etc/hosts

#Numer IP      Nazwa hosta      Aliasy
127.0.0.1      localhost
192.168.1.1    gate             router
192.168.1.10   merkury
192.168.1.11   mars
192.168.1.12   jupiter
192.168.1.13   saturn
81.21.195.180  enigma
81.21.195.159  fraxinus
```

Plik */etc/networks*

Plik */etc/networks* ma funkcję zbliżoną do funkcji pliku */etc/hosts*. Jest prostą bazą danych nazw i adresów sieci.

W przypadku używania programu */sbin/route* jeśli punkt docelowy jest siecią, a sieć ta znajduje się w pliku */etc/networks*, wtedy polecenie *route* zamiast adresu IP sieci wyświetli jej nazwę.

```
# Przykładowy plik /etc/networks

#Nazwa sieci      Adres sieci
loopnet           127.0.0.0
localnet          192.168.1.0
tknet             81.21.195.128
cyfronet          149.156.4.64
```

Plik */etc/ethers*

Plik */etc/ethers* zawiera numery MAC kart sieciowych i przypisane do nich adresy IP lub nazwy hostów, jeśli znajdują się one w bazie nazw.

Każde urządzenie sieciowe posiada niepowtarzalny adres fizyczny, inaczej zwany adresem sprzętowym, który dla sieci Ethernet ma długość 48 bitów. Zakodowany jest zwykle w karcie sieciowej. Adres ten podawany zazwyczaj w zapisie szesnastkowym oddzielonym dwukropkami znany jest jako adres sterowania dostępem do nośnika (*MAC*, *Media Access Control*).

Plik */etc/ethers* wykorzystywany jest przez protokół RARP. Protokół ten umożliwia komputerom rozprzestrzenianie ich adresu sprzętowego (MAC), oczekując aż demon serwera zarządzający adresami w sieci lokalnej odpowie, podając adres IP. Dzięki temu protokołowi komputery bezdyskowe oraz inne urządzenia sieciowe mogą otrzymywać własne adresy IP.

```
# Przykładowy plik /etc/ethers

#Adres sprzetowy  Adres IP lub nazwa hosta
00:00:00:00:00:00 komputer1
00:00:00:00:00:00 komputer2
00:00:00:00:00:00 terminal1
00:00:00:00:00:00 terminal2
```

Plik /etc/protocols

Plik */etc/protocols* zawiera informacje przyporządkowujące nazwom protokołów odpowiednie numery. Jest wykorzystywany przez różne programy pozwalając na podawanie nazw protokołów zamiast numerów.

Plik ten opisuje różne, dostępne w podsystemie TCP/IP, protokoły internetowe DARPA. Zawarte w tym pliku numery protokołów pojawiają się w polu **protocol** każdego nagłówka IP. Zmiany w pliku */etc/protocols* mogą być przyczyną powstawania nieprawidłowych pakietów IP. Numery protokołów i ich nazwy ustalane są przez DDN Network Information Center.

Ten plik może być rozprowadzany przez sieć przy użyciu ogólnosieciowej usługi nazewnictwa, takiej jak NIS czy BIND.

Fragment pliku /etc/protocols

#Protokół	Numer	Alias	Komentarz
ip	0	IP	#Internet Protocol
icmp	1	ICMP	#Internet Control Message Protocol
igmp	2	IGMP	#Internet Group Management
ggp	3	GGP	#Gateway-Gateway Protocol
ipencap	4	IP-ENCAP	#IP encapsulated in IP
st	5	ST	#ST datagram mode
tcp	6	TCP	#Transmission Control Protocol
egp	8	EGP	#Exterior Gateway Protocol
igp	9	IGP	#Interior Gateway Protocol
pup	12	PUP	#PARC Universal Packet Protocol
udp	17	UDP	#User Datagram Protocol
hmp	20	HMP	#Host Monitoring Protocol
xns-idp	22	XNS-IDP	#Xerox NS IDP
rdp	27	RDP	#"Reliable datagram" protocol
iso-tp4	29	ISO-TP4	#ISO Transport Protocol class 4
xtp	36	XTP	#Xpress Transfer Protocol
ddp	37	DDP	#Datagram Delivery Protocol
idpr-cmtp	38	IDPR-CMTP	#IDPR Control Message Transport
idrp	45	IDRP	#Inter-Domain Routing Protocol
rsvp	46	RSVP	#Reservation Protocol
gre	47	GRE	#General Routing Encapsulation
skip	57	SKIP	#SKIP
rspf	73	RSPF	#Radio Shortest Path First
iso-ip	80	ISO-IP	#ISO Internet Protocol
vmtp	81	VMTP	#Versatile Message Transport
ospf	89	OSPF	#Open Shortest Path First IGP
larp	91	LARP	#Locus Address Resolution Protocol
mtp	92	MTP	#Multicast Transport Protocol
ipip	94	IPIP	#IP-within-IP Encapsulation Protocol
etherip	97	ETHERIP	#Ethernet-within-IP Encapsulation
encap	98	ENCAP	#Encapsulation Header
pim	103	PIM	#Protocol Independent Multicast

Plik /etc/host.conf

Plik */etc/host.conf* określa kolejność poszukiwania nazwy hosta w różnych systemach poszukiwania nazw (plik */etc/hosts*, DNS, NIS). Każdy wiersz pliku */etc/host.conf* składa się z dyrektywy wraz z parametrami.

Dyrektywa	Funkcja
order	Określa porządek przeszukiwania w określonych systemach. Jako parametry podaje się dowolną kombinację metod wyszukiwania oddzielonych przecinkami. Obsługiwane metody wyszukiwania to <code>bind</code> , <code>hosts</code> oraz <code>nis</code> znaczące odpowiednio DNS, plik <code>/etc/hosts</code> oraz NIS.
trim	Określa domenę, która będzie odrzucona z nazwy hosta w czasie dokonywania przekształcenia adresu IP na nazwę hosta przez system DNS. Dyrektywę <code>trim</code> można wymienić wiele razy dla wielu domen. Dyrektywa <code>trim</code> nie dotyczy przeszukiwania za pomocą pliku <code>/etc/hosts</code> oraz NIS. Należy zadbać o to, aby w pliku <code>/etc/hosts</code> oraz tabelach NIS hosty były wymienione prawidłowo (z pełną domeną lub bez domeny).
multi	Decyduje, czy zapytanie do systemu nazw będzie zwracało tylko jeden wynik, czy też jest dopuszczalne zwracanie kilku wyników. Parametry to <code>on</code> , co oznacza, że jest dopuszczalne kilka wyników lub <code>off</code> , co oznacza, że zawsze będzie zwracany jeden wynik. Domyślną wartością jest <code>off</code> .
nospoof	Włącza lub wyłącza funkcję bezpieczeństwa polegającą na ochronie przed „zwodzeniem adresów” (<i>spoofing</i>). Jeżeli <code>nospoof</code> jest ustawione na <code>on</code> , to po każdej operacji zamiany nazwy na adres IP, będzie wykonana odwrotna operacja: zamiana adresu IP na nazwę. Operacja nie powiedzie się, jeżeli nazwy nie będą zgodne. Domyślną wartością jest <code>off</code> .
alert	Jeżeli dyrektywa <code>nospoof</code> jest ustawiona na wartość <code>on</code> , dyrektywa <code>alert</code> steruje funkcją raportowania prób zwodzenia adresów poprzez funkcję dzienników systemowych (<code>syslog</code>). Domyślną wartością jest <code>off</code> .
reorder	Jeżeli jest ustawione na <code>on</code> , wszystkie wyszukiwania będą wykonywane w odwrotnej kolejności. Tak więc hosty w tej samej podsieci będą zwracane jako pierwsze. Domyślną wartością jest <code>off</code> .

Tab. 8. Format pliku `/etc/host.conf`

```
# Przykładowy plik /etc/host.conf

order hosts,bind
multi on
nospoof on
```

Plik `/etc/nsswitch.conf`

Wersja 2. standardowej biblioteki GNU oferuje wydajniejszy i bardziej elastyczny mechanizm, który zastępuje plik `/etc/host.conf`. Pojęcie usługi nazewnicznej zostało rozszerzone tak, że zawiera wiele różnych informacji.

Plik `/etc/nsswitch` został utworzony przez firmę Sun Microsystems w celu zarządzania kolejnością wyszukiwania kilku plików konfiguracyjnych w systemie. Plik ten pozwala administratorowi na systemu skonfigurować szereg różnych baz danych.

Każdy wiersz pliku `/etc/nsswitch.conf`, który nie jest komentarzem, to słowo kluczowe, po którym następuje dwukropek i lista metod w porządku, w jakim będą one stosowane. Każde słowo kluczowe to nazwa pliku w katalogu `/etc`, który jest sterowany plikiem `/etc/nsswitch.conf`.

Słowo kluczowe	Funkcja
aliases	aliasy pocztowe
passwd	użytkownicy systemu
group	grupy użytkowników
shadow	hasła typu <i>shadow</i>
hosts	nazwy hostów i adresy IP
networks	nazwy sieci i ich adresy
protocols	protokoły sieciowe
services	numery portów i nazwy usług
ethers	adresy Ethernet
rpc	nazwy RPC oraz adresy
netgroup	grupy sieciowe

Tab. 9. Dopuszczalne słowa kluczowe w pliku */etc/nsswitch.conf*

Metoda	Znaczenie
files	Poprawna dla wszystkich słów kluczowych z wyjątkiem netgroup . Poszukiwanie rekordu w skojarzonym pliku <i>/etc</i> .
db	Poprawna dla wszystkich słów kluczowych z wyjątkiem netgroup . Wyszukiwanie rekordów w skojarzonej bazie danych w katalogu <i>/var/db</i> . Jest to użyteczne dla bardzo dużych plików. Aby utworzyć te pliki ze standardowych plików <i>/etc</i> należy uruchomić polecenie make w katalogu <i>/var/db</i> .
compat	Tryb zgodności – poprawne dla plików <i>passwd</i> , <i>group</i> oraz <i>shadow</i> . W tym trybie poszukiwania są wykonywane najpierw w odpowiednim pliku <i>/etc</i> . Aby wykonać wyszukiwanie w bazie danych NIS, należy w określonym pliku dodać wiersz, gdzie pierwotne pole (nazwa użytkownika lub grupy) jest oznaczona znakiem +, po którym następuje odpowiednia liczba dwukropków (6 dla pliku <i>/etc/passwd</i> , 3 dla pliku <i>/etc/group</i> , 8 dla pliku <i>/etc/shadow</i>).
dns	Prawidłowa tylko dla słowa kluczowego hosts . Wyszukiwanie odbywa się poprzez system DNS według konfiguracji w pliku <i>/etc/resolv.conf</i> .
nis	Prawidłowa dla wszystkich słów kluczowych. Wyszukiwanie odbywa się na serwerze NIS, jeżeli NIS jest aktywny.
<[!]STATUS=akcja>	<p>Steruje akcją usługi wyszukiwania nazw. STATUS może przyjmować wartości:</p> <ul style="list-style-type: none"> SUCCESS (operacja zakończyła się sukcesem) NOTFOUND (rekordu nie znaleziono) UNAVAIL (wybrana usługa była niedostępna) TRYAGAIN (usługa czasowo niedostępna, należy spróbować jeszcze raz). <p>Akcja może zaś przyjąć wartości:</p> <ul style="list-style-type: none"> return (zaprzestanie szukania i zwrócenie bieżącego statusu) continue (kontynuowanie z następnym elementem w danym wierszu). <p>Opcjonalny znak wykrzyknika oznacza „nie”.</p>

Tab. 10. Lista dopuszczalnych metod w pliku */etc/nsswitch.conf*

```
# Przykładowy plik /etc/nsswitch.conf

passwd:      compat
group:       compat
shadow:      compat

hosts:       dns [!UNAVAIL=return] files
#Szukanie hosta za pomocą systemu DNS. Jeżeli tylko zwrócony status
#nie oznacza niedostępności, resolver zwraca to, co znalazł. Jeżeli
#próba zapytania DNS zwróciła status niedostępności, resolver próbuje
#użyć lokalnego pliku /etc/hosts.

networks:    files
protocols:   files
services:    files
ethers:      files
rpc:         files
netgroup:    files
```

Plik /etc/resolv.conf

Plik */etc/resolv.conf* służy do konfiguracji klienta DNS. Zawiera on porządek przeszukiwania nazw domen oraz adresy serwerów DNS. Każdy wiersz powinien zawierać słowo kluczowe i jeden lub więcej parametrów oddzielonych spacjami.

Słowo kluczowe	Znaczenie
nameserver	Pojedynczy parametr tego słowa kluczowego określa adres IP serwera DNS. Można umieścić kilka wierszy ze słowem kluczowym nameserver , każdy z pojedynczym adresem IP. Serwery nazw będą odpytywane w kolejności, w jakiej są wymienione w pliku. Serwery nazw, które są wymienione jako kolejne, będą odpytane tylko wtedy, kiedy pierwszy nie odpowie.
domain	Pojedynczy parametr określa nazwę domeny hosta. Ta nazwa jest wykorzystywana przez niektóre programy, takie jak np. system poczty elektronicznej, a także w czasie tworzenia zapytania DNS o hosta bez wymienionej domeny (bez kropek w nazwie). Jeżeli nie wymieniono nazwy domeny, będzie wykorzystywana nazwa hosta, z której usunięto wszystko przed pierwszą kropką.
search	Parametry określają porządek przeszukiwania nazw domen. Jeżeli występuje zapytanie o hosta bez wymienionej domeny, host będzie kolejno wyszukiwany we wszystkich domenach wymienionych jako parametry słowa kluczowego search . Słowa kluczowe domain oraz search wzajemnie się wykluczają. Jeżeli wymienimy oba, wykorzystywane będzie to, które jest wymienione jako drugie.
sortlist	Umożliwia sortowanie zwróconych nazw domen w określonym porządku. Parametry tego słowa kluczowego są określone przez pary sieć – maska sieciowa, co umożliwia wprowadzenie dowolnych porządków sortowania.

Tab. 11. Lista dopuszczalnych słów kluczowych w pliku */etc/resolv.conf*

```
# Przykładowy plik /etc/resolv.conf

search fraxinus.tk.krakow.pl tk.krakow.pl
nameserver 81.21.195.180
nameserver 81.21.195.190
sortlist 81.21.195.180/255.255.255.192 81.21.195.190/255.255.255.192
```

Plik */etc/network/interfaces*

Debian korzysta z pliku */etc/network/interfaces* do konfiguracji interfejsów sieciowych. W systemie Linux, wszystkie interfejsy sieciowe mają nazwy złożone z nazwy sterownika (interfejsu), po którym następuje liczba.

Interfejs	Typ urządzenia
lo	pętla zwrotna (loopback)
eth	ethernet
ppp	protokół punkt-punkt (PPP)
slip	protokół SLIP (IP przez łącze szeregowe)
plip	protokół PLIP (IP przez łącze równoległe)
tunl	tunel sieciowy
sit	tunel IPv6 poprzez IP

Tab. 12. Niektóre z nazw sterowników sieciowych

Interfejsy są numerowane począwszy od 0 w porządku odnajdywania ich przez jądro systemu lub, jeżeli sterowniki są ładowane jako moduły, w pliku */etc/modules.conf*. Domyślnie jądro Linuksa odnajduje tylko jedną sieć. W przypadku występowania więcej niż jednej karty sieciowej, należy odpowiednio zmodyfikować plik */etc/modules.conf* (jeśli sterowniki są ładowane jako moduły) lub */etc/lilo.conf* (jeżeli sterowniki do karty sieciowej są wkompilowane w jądro).

```
# Zmodyfikowany fragment pliku /etc/modules.conf
alias eth0 8139too
alias eth1 8139too

# Zmodyfikowany fragment pliku /etc/lilo.conf
#append="ether=IRQ,I/O,parametry,nazwa"
append="ether=0,0,eth1"
```

Plik */etc/network/interfaces* rozpoczyna się, od dyrektywy, po której następują dalsze parametry.

Dyrektywa	Znaczenie
auto	Identyfikuje interfejsy sieciowe, które powinny być uruchomione automatycznie w czasie startu systemu. Nazwy interfejsów powinny znajdować się w tej samej linii, co dyrektywa auto .
mapping	Automatycznie tłumaczy nazwę fizycznego interfejsu na logiczny interfejs określony przez dyrektywę iface . Każde mapowanie musi określać skrypt, który będzie uruchamiany, kiedy stosowane jest mapowanie. W dodatku, mapowanie może określać dowolny numer parametrów map , które będą przekazywane do skryptu.
iface	Definiuje interfejsy sieciowe, których nazwa jest określona po dyrektywie iface .

Tab. 13. Funkcje dyrektyw w pliku */etc/network/interfaces*

Nazwa interfejsu jest związany rodziną adresów, których dany interfejs używa. Generalnie dla sieci TCP/IP jest to `inet`, ale może to być `ipx` dla sieci IPX/SPX lub `inet6` dla sieci opartej o wersję 6 protokołu IP.

Opcja	Działanie
<code>up <polecenie></code>	Uruchamia polecenie po włączeniu interfejsu. Opcja ta może być podana wielokrotnie dla pojedynczego interfejsu. W takim przypadku, polecenia będą wykonywane według kolejności. Jeśli któraś z nich się nie powiedzie, żadna z pozostałych nie zostanie wykonana, ale interfejs zostanie skonfigurowany.
<code>pre-up <polecenie></code>	Uruchamia polecenie przed włączeniem interfejsu. Opcja ta może być podana wielokrotnie dla pojedynczego interfejsu. W takim przypadku, polecenia będą wykonywane według kolejności. Jeśli któraś z nich się nie powiedzie, żadna z pozostałych nie zostanie wykonana i interfejs nie będzie skonfigurowany.
<code>down <polecenie></code>	Uruchamia polecenie przed wyłączeniem interfejsu. Opcja ta może być podana wielokrotnie dla pojedynczego interfejsu. W takim przypadku, polecenia będą wykonywane według kolejności. Jeśli któraś z nich się nie powiedzie, żadna z pozostałych nie zostanie wykonana i interfejs nie będzie zdekongfigurowany.
<code>post-down <polecenie></code>	Uruchamia polecenie po wyłączeniu interfejsu. Opcja ta może być podana wielokrotnie dla pojedynczego interfejsu. W takim przypadku, polecenia będą wykonywane według kolejności. Jeśli któraś z nich się nie powiedzie, żadna z pozostałych nie zostanie wykonana, ale interfejs zostanie zdekongfigurowany.

Tab. 14. Standardowe opcje w pliku `/etc/network/interfaces`

Metoda	Opis	Opcja	Działanie
<code>loopback</code>	Określa interfejs pętli zwrotnej	–	
<code>static</code>	Określa interfejsy ethernetowe ze statycznym przydziałem adresów IP	<code>address</code>	Przypisuje dany adres IP
		<code>netmask</code>	Przypisuje maskę sieciową
		<code>broadcast</code>	Przypisuje adres rozgłoszeniowy
		<code>network</code>	Przypisuje adres sieci
		<code>gateway</code>	Przypisuje adres domyślnej bramy
		<code>pointopoint</code>	Przypisuje adres końcowego punktu
<code>dhcp</code>	Nabywa adres poprzez klienta protokołu DHCP	<code>hostname</code>	Określa nazwę hosta
		<code>leasetime</code>	Preferowany czas dzierżawy (w sekundach)
<code>ppp</code>	Konfiguruje interfejs PPP	<code>provider</code>	Używa podanej nazwy (z pliku <code>/etc/ppp/peers</code>) jako dostawcy

Tab. 15. Najczęściej stosowane metody i ich opcje w pliku `/etc/network/interfaces`

```
# Przykładowy plik /etc/network/interfaces

auto lo
iface lo inet loopback
#Interfejs petli zwrotnej

auto eth0
iface eth0 inet static
    address 81.21.195.159
    netmask 255.255.255.192
    network 81.21.195.128
    broadcast 81.21.195.191
    gateway 81.21.195.129
#Interfejs podstawowy polaczenia z Internetem

auto eth1
iface eth1 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    broadcast 192.168.1.255
#Interfejs podstawowy dla sieci lokalnej
```

Programy konfiguracyjne

Pliki opisane poprzednio służą do konfiguracji ogólnych parametrów sieci. Chociaż większość nich może być zmieniana dynamicznie, poprzez edycję odpowiedniego pliku, niektóre parametry sieci należy modyfikować za pomocą odpowiednich programów. Do takich parametrów należą adres IP hosta oraz tabela trasowania.

Program *ifconfig*

Program */sbin/ifconfig* jest wykorzystywany do konfiguracji interfejsów sieciowych hosta. Polecenie *ifconfig* ma postać:

```
ifconfig <interfejs> <adres IP> [netmask <maska sieci>]
[broadcast <adres rozgłoszeniowy>]
```

Program *ifconfig* w tej postaci może być stosowany jedynie przez administratora (użytkownika *root*). Parametry *netmask* oraz *broadcast* są opcjonalne. Jeżeli zostaną pominięte, polecenie *ifconfig* pobierze te wartości z domyślnej klasy adresu IP. Parametry te powinny zostać wymienione, jeżeli wydzielamy podsieci. Polecenie będzie próbowało działać według podanych adresów. Nie nastąpi sprawdzenie, czy adres rozgłoszeniowy odpowiada podanemu adresowi IP oraz masce sieci. Polecenie ładuje odpowiedni sterownik sieci i konfiguruje interfejs.

Aby sprawdzić status interfejsu sieciowego, należy użyć polecenia:

```
ifconfig <interfejs>
```

W tej formie polecenie */sbin/ifconfig* może być uruchamiane przez dowolnego użytkownika. W wyniku otrzymamy wszystkie opcje konfiguracyjne interfejsu obejmujące adres IP, maskę podsieci, adres rozgłoszeniowy oraz fizyczny adres sprzętowy. Wyświetlone będą także informacje na temat statusu interfejsu oraz inne statystyki i informacje jak maksymalna jednostka transmisji, adres I/O karty sieciowej oraz numer IRQ, a także liczba otrzymanych pakietów, pakietów wysłanych oraz liczba kolizji.

Nieraz zachodzi sytuacja, że pojedynczy interfejs sieciowy ma posiadać kilka adresów IP. Linux posiada funkcję aliasów sieciowych, która umożliwia zrealizowanie tej opcji. Jeżeli działające jądro ma włączoną obsługę aliasów, stworzenie aliasu sprowadza się do wprowadzenia polecenia *ifconfig*. Wystarczy jedynie do nazwy interfejsu dodać dwukropek oraz numer aliasu:

```
ifconfig eth1:0 192.168.1.254
```

Polecenie to tworzy alias **eth1:0** dla interfejsu ethernetowego **eth1** z podanymi parametrami.

Aby system automatycznie tworzył alias podczas ładowania systemu, należy dodać to polecenie do pliku */etc/init.d/network* lub też odpowiednio zmodyfikować plik */etc/network/interfaces*.

Opcja	Funkcja
up	Włączenie interfejsu.
down	Wyłączenie interfejsu.
[-]arp	Włączenie lub wyłączenie korzystania z protokołu ARP.
[-]allmulti	Włączenie lub wyłączenie korzystania z trybu pracy <i>promiscious</i> , w którym urządzenie może być zmuszone do odbierania wszelkich pakietów, a nie tylko tych adresowanych bezpośrednio do niego.
mtu <ilość bitów>	Ustawienie wielkości parametru MTU (rozmiar największego pakietu, który może być przesyłany przez interfejs).
netmask <maska sieci>	Ustawienie wartości maski sieci, do której jest podłączony interfejs.
irq <nr przerwania>	Ustawienie wartości przerwania IRQ, z którego powinno korzystać dane urządzenie.
[-]broadcast <adres broadcast>	Włączenie lub wyłączenie odbierania pakietów skierowanych na podany adres rozgłoszeniowy.
[-]pointopoint <adres>	Ustawienie adresu komputera na drugim końcu połączenia point-to-point obsługiwanego przez ten interfejs (dla protokołów jak <i>slip</i> czy <i>ppp</i>).
hw <adres sprzętowy>	Określenie adresu sprzętowego (dla ograniczonego rodzaju urządzeń).

Tab. 16. Ważniejsze opcje polecenia *ifconfig*

Program route

Polecenie */sbin/route* jest wykorzystywane do przetwarzania tabeli trasowania jądra. Ta tabela jest wykorzystywana przez jądro w celu określenia, co należy zrobić z każdym pakietem wychodzącym z hosta – czy wysłać go bezpośrednio do hosta docelowego czy do routera i poprzez jaki interfejs sieciowy.

Polecenia *route* ma postać:

```
route [opcje] [polecenie [parametry]]
```

Najprostsza forma polecenia (bez opcji i poleceń) wyświetla tabelę trasowania. W tej formie polecenie *route* może być uruchamiane przez dowolnego użytkownika.

Pole	Opis
Destination	Określa punkt docelowy trasy. Jeżeli w pliku <i>/etc/hosts</i> lub w pliku <i>/etc/networks</i> istnieje odpowiedni zapis, nazwa będzie w określony sposób zamieniona. Specjalna nazwa default określa domyślny ruter.
Gateway	Określa ruter, przez który będą przesyłane pakiety do danego celu. Gwiazdka oznacza, że pakiety będą bezpośrednio wysyłane do hosta docelowego.
Genmask	Określa skojarzoną z trasą maskę sieci. Maska sieci będzie dotyczyć wartości w kolumnie Destination .
Flags	Może mieć kilka wartości. Najpopularniejsze flagi to: <ul style="list-style-type: none"> ▪ U – trasa włączona (<i>up</i>) ▪ H – trasa statyczna do określonego hosta ▪ G – pakiety będą przesyłane do hosta docelowego poprzez ruter.
Metric	Określa dystans do celu. Jest ona wykorzystywana przez niektóre demony trasowania do obliczania najlepszej trasy do hosta docelowego.
Ref	W systemach UNIX określa liczbę odnośników do tej trasy. Nie jest ona wykorzystywana przez jądro systemu Linux.
Use	Określa liczbę poszukiwań danej trasy przez jądro.
Iface	Wyświetla nazwę interfejsu, przez który będą przechodzić pakiety skierowane do tej trasy.

Tab. 17. Pola polecenia *route*

Polecenie *route* można także zastosować do dodawania i usuwania tras z tabeli trasowania. Wykonuje się to za pomocą poleceń:

```
route add|del [-net|-host] <cel> [gw <ruter>] [netmask <maska>]
[dev <interfejs>]
```

gdzie **add** oraz **del** oznaczają odpowiednio dodanie oraz usunięcie trasy, opcje **-net** oraz **-host** wskazują, czy operacje będą wykonywane z trasą sieciową, czy z trasą hosta, parametr **<cel>** to nazwa hosta lub sieci docelowego adresu IP, lub też słowem kluczowym **default** w celu skonfigurowania domyślnej trasy, parametr **gw <ruter>** określa, który ruter ma być zastosowany dla danej trasy, parametr **netmask** określa maskę sieci dla trasy (dotyczy adresu **<cel>**), natomiast parametr **dev <interfejs>** określa interfejs, przez który będą przesyłane pakiety do wskazanego celu.

Najczęściej polecenie *route* jest wykorzystywane do wykonania operacji z trasami sieciowymi (tymi, które prowadzą do zdalnej sieci), ale czasem trzeba dodać trasy do określonych hostów. Jest to konieczne, jeżeli host jest połączony bezpośrednio poprzez łącze punkt-punkt (protokół PPP) – na przykład poprzez modem lub kabel szeregowy.

Programy testujące

Po skonfigurowaniu protokołu TCP/IP problemy pojawiają się rzadko. Urządzenia sieciowe ulegają uszkodzeniom, kable się rozłączają, połączenia są zrywane. Pojawiają się także problemy podczas sekwencji inicjującej komputera lokalnego.

Program ping

Najbardziej podstawowym narzędziem pomocnym przy lokalizacji problemów jest polecenie *ping*. Program *ping* wysyła pakiety do innego komputera i oczekuje na odpowiedź. Jest on bardzo użyteczny do sprawdzania, czy komputery w sieci mogą wzajemnie nawiązać połączenie. Jeśli oba widzą się przy zastosowaniu polecenia *ping*, powinny być w stanie wymieniać między sobą każdy rodzaj danych. Polecenie to używa protokołu ICMP (*Internet Control Message Protocol*). Protokół ten jest nadrzędnym dla IP i jest zaprojektowany do kontroli komunikatów, przydaje się do takich zastosowań jak rutowanie i kontrola nad osiągalnością informacji.

Podstawową metodą użycia *ping* jest wywołanie go z podaniem nazwy lub adresu szukanego komputera. W takim przypadku program *ping* wysyła pakiety co sekundę do momentu wciśnięcia [Ctrl]+[C]. Wtedy wyświetla statystykę ze swojej pracy. W statystyce, oprócz liczby wysłanych i odebranych pakietów, można zobaczyć minimalny, średni i maksymalny czas pomiędzy wysłaniem i odebraniem pakietu, co pozwala zaobserwować, jak sieć jest obciążona.

Opcja	Znaczenie
-c <liczba>	Powoduje wysłanie tylko podanej liczby pakietów.
-n	Program wyświetla adresy IP zamiast nazw domenowych. Przydatne, gdy serwer DNS odpowiada bardzo wolno.
-R	Program zapisuje informacje o drodze pakietu. Wymusza zapisywanie w przesyłanym pakiecie adresów IP komputerów, przez jakie ten pakiet przechodzi. Dzięki temu można sprawdzić, jaką drogą przechodzi pakiet. Liczba rejestrowanych adresów jest ograniczona do dziewięciu.
-t	Program zmienia domyślną wartość <i>ttl</i> (<i>Time-To-Live</i>). Wartość pola czasu życia pakietów jest zmniejszana o jeden przez każdy ruter napotkany w sieci. Gdy osiągnie wartość 0 pakiet jest porzucany. Program <i>ping</i> używa domyślnej wartości TTL 255 (wartość maksymalna), jednak wiele innych programów używa mniejszych wartości (zwykle 30 lub 60). Oznacza to, że host widoczny przy wykorzystaniu programu <i>ping</i> może się stać nieosiągalny w przypadku użycia innych programów.
-q	Program wyświetla tylko statystykę.
-v	Program wyświetla informacje o wszystkich odebranych pakietach, nie tylko o odpowiedziach na wysłane przez <i>ping</i> .

Tab. 18. Ważniejsze opcje polecenia *ping*

Program traceroute

Program *traceroute* jest podstawowym narzędziem do diagnozowania problemów z sieciami TCP/IP. Wysyła on pakiety UDP z coraz większymi wartościami parametru TTL i obserwuje informacje zwracane przez bramki podczas porzucania pakietów. Umożliwia to obserwację drogi pakietów od komputera lokalnego do docelowego.

Program *traceroute* wysyła trzy pakiety dla każdej wartości TTL i wyświetla czas oczekiwania na odpowiedź dla każdego z nich, co pozwala zlokalizować miejsca, które są „wąskimi gardłami” systemu.

Polecenie *traceroute* zwykle jest wykorzystywane w ten sposób, jak polecenie *ping* – jako parametr należy podać nazwę hosta docelowego.

Program tcpdump

Program *tcpdump* działa na zasadzie podsłuchiwania pakietów, które są przesyłane w sieci (bez względu na to, do którego hosta są adresowane). Pozwala to na zbieranie wszystkich lub tylko wybranych informacji o ruchu w sieci, może je wyświetlić lub przeprowadzić proste analizy zebranych danych.

Program *tcpdump* działa, ustawiając kartę sieciową w tzw. „tryb *promiscuous*”. Zwykle karty sieciowe widzą tylko pakiety adresowane bezpośrednio do nich. Jednak w trybie tym widzi wszystkie pakiety przesyłane w sieci i przekazuje dane o nich do systemu operacyjnego. System operacyjny z kolei przekazuje je do programu *tcpdump*, a ten może je filtrować, wyświetlać bądź przechowywać. Aby móc zmodyfikować ustawienia karty sieciowej, program *tcpdump* musi być uruchamiany przez użytkownika z prawami administratora.

Uruchomione bez żadnych parametrów polecenie, otrzymuje się informacje o wszystkich pakietach przesyłanych w danej sieci. Program przerywa pracę po wciśnięciu [Ctrl]+[C].

Program *tcpdump* domyślnie zamienia adresy IP na nazwy domen, a numery portów na nazwy usług. Często nie ma potrzeby oglądać wszystkich parametrów. Parametrem polecenia *tcpdump* może być wyrażenie filtrujące, określające typ pakietów, jakie mają być wyświetlane. Wyrażenia filtrujące składają się z jednego lub więcej wyrażeń prostych połączonych operatorami *and*, *or* oraz *not*. Wyrażenia proste składają się z kwalifikatora poprzedzonego identyfikatorem. Kwalifikator składa się z jednego lub więcej słów kluczowych, zaś identyfikator określa wartość, jaką musi mieć dane pole, aby było przepuszczone przez filtr.

Kwalifikator	Filtrowane pole
src host	Adres IP hosta, który wysłał pakiet
dst host	Adres IP hosta, będącego adresatem pakietu
host	Adres IP hosta, który wysłał lub jest odbiorcą pakietu
src port	Numer portu, z którego wysłano pakiet
dst port	Numer portu będącego adresatem pakietu
port	Numer portu, do którego pakiet jest kierowany lub, z którego był wysłany
tcp,udp,icmp	Typ protokołu wykorzystywanego przez pakiet

Tab. 19. Najczęściej wykorzystywane kwalifikatory w poleceniu *tcpdump*

Opcja	Znaczenie
-c <liczba>	Wymusza zakończenie działania programu po otrzymaniu określonej liczby pakietów.
-i <interfejs>	Program będzie nasłuchiwał na wskazanym interfejsie. Domyślnie nasłuchiwany jest pierwszy interfejs znaleziony po pętli zwrotnej.
-n	Adresy IP i numery portów nie będą zamieniane na odpowiadające im nazwy komputerów i usług.
-N	Wyświetla tylko nazwę komputera (bez pełnej nazwy domeny).
-r <plik>	Odczytuje pakiety zapisane wcześniej przez wykorzystanie opcji -w.
-s <liczba>	Powoduje przechwycenie określonej liczby bajtów z każdego pakietu. Domyślnie jest to 68 bajtów, co jest wystarczające dla pakietów IP, ICMP, TCP i UDP. Jednak przy użyciu niektórych protokołów (np. DNS) domyślna długość spowoduje utratę części informacji.
-v	Wyświetla dodatkowe informacje o każdym pakiecie.
-vv	Wyświetla jeszcze więcej informacji o każdym pakiecie.
-w <plik>	Zapisuje pakiety do wskazanego pliku.
-x	Wyświetla pakiety w postaci heksadecymalnej.

Tab. 20. Najbardziej użyteczne opcje programu *tcpdump*

Program netstat

Program *netstat* wyświetla status wszystkich usług sieciowych TCP/IP. Polecenie *netstat* daje szczegółowy opis portów, pakietów oraz innych zagadnień dotyczących sieci.

Polecenie *netstat* bez opcji wyświetla listę wszystkich podłączonych gniazd. Informacje o każdym gnieździe obejmuje:

- Protokół (*tcp* lub *udp*)
- Liczbę bajtów w kolejkach nadawczych i odbiorczych (bajtów, które nie zostały odczytane przez procesy lokalne lub takich, których odbiór nie został potwierdzony przez procesy zdalne)
- Adresy hosta lokalnego i zdalnego (adres hosta zdalnego dla gniazd, które są w stanie LISTEN są wyświetlane jako *.*)
- Stan gniazda

Stan gniazda	Znaczenie
SYN_	Adres IP hosta, który wysłał pakiet
WAIT	Adres IP hosta, będącego adresatem pakietu
ESTABLISHED	Adres IP hosta, który wysłał lub jest odbiorcą pakietu
LISTEN	Numer portu, z którego wysłano pakiet
CLOSED	Typ protokołu wykorzystywanego przez pakiet

Tab. 21. Stany gniazd w poleceniu *netstat* i ich znaczenie

Dodatkowo opcja *-e* powoduje wyświetlenie informacji o użytkowniku, który aktualnie wykorzystuje gniazdo.

5. SERWERY USŁUGOWE

Większość serwerów usług TCP/IP jest realizowanych jako oddzielne demony. Demony (serwery) to procesy, które są uruchomione w tle i wykonują określone operacje w określonych predefiniowanych momentach czasu lub w odpowiedzi na określone zdarzenia. Demon jest cały czas aktywny w czasie, gdy system jest uruchomiony, do momentu, kiedy nie zostanie zatrzymany.

Demony są zazwyczaj uruchamiane w czasie ładowania systemu albo bezpośrednio przez administratora. W ten sposób uruchamia się wiele programów systemowych.

Mimo, że demony sieciowe mają wiele różnych funkcji, większość z nich posiada kilka cech wspólnych:

- Nazwy ich kończą się zazwyczaj na literę *d* (od słowa *daemon*).
- Ich odpowiedzią na sygnał HUP (*kill -HUP*) jest ponowne wczytanie plików konfiguracyjnych.
- Zwykle są uruchamiane w czasie ładowania systemu poprzez skrypty w katalogu */etc/init.d*.
- Kiedy otrzymują żądanie, tworzą kopię samego siebie w celu obsługi tego żądania. Z tego względu w danym czasie może istnieć kilka kopii danego demona pracujących równocześnie.

SUPERSERWER INETD

Program *inetd* jest demonem, który uruchamia się podczas ładowania systemu. Odczytuje on plik konfiguracyjny */etc/inetd.conf*, informujący o tym, których gniazd należy słuchać i jakie uruchomić programy po nawiązaniu połączenia dla każdego z tych gniazd. Obsługuje on tworzenie gniazd, słuchanie do momentu uzyskania połączenia, utworzenie nowego procesu w celu obsługi tego połączenia i przekazanie do tego procesu połączeń do gniazda jako standardowego wejścia oraz standardowego wyjścia.

Istnieje jednak jedna wada uruchamiania serwerów poprzez demona *inetd*. Procedura uruchamiania serwera trwa dłużej. Dzieje się tak, ponieważ w przypadku oddzielnych demonów, proces serwera jest ciągle aktywny i stale działa, natomiast w przypadku *inetd*, musi on załadować proces serwera za każdym uruchomieniem.

Konfiguracja demona inetd

Zasadniczo demon *inetd* jest konfigurowany poprzez plik */etc/inetd.conf*, jednak nie jest on jedynym plikiem konfiguracyjnym potrzebnym do działania *inetd*.

Plik /etc/services

Plik */etc/services* zawiera odwzorowanie pomiędzy numerami portów a nazwami usług. Jest to wykorzystywane przez niektóre programy systemowe. W pliku */etc/services* można umieścić aliasy. Wpisuje się je po numerach portów.

Fragment pliku /etc/services

#Usługa	Port/protokół	Alias
tcpmux	1/tcp	
echo	7/tcp	
echo	7/udp	
discard	9/tcp	sink null
discard	9/udp	sink null
systat	11/tcp	users
daytime	13/tcp	
daytime	13/udp	
netstat	15/tcp	
qotd	17/tcp	quote
msh	18/tcp	
msh	18/udp	
chargen	19/tcp	ttytst source
chargen	19/udp	ttytst source
ftp-data	20/tcp	
ftp	21/tcp	
fsp	21/udp	fspd
ssh	22/tcp	
ssh	22/udp	
telnet	23/tcp	
smtp	25/tcp	mail
time	37/tcp	timserver
time	37/udp	timserver
rlp	39/udp	resource
nameserver	42/tcp	name
whois	43/tcp	nicname

Plik `/etc/inetd.conf`

Plik `/etc/inetd.conf` służy do konfigurowania demona *inetd*. Każdy wiersz tego pliku ma następujący format:

```
<usługa> <typ gniazda> <protokół> [no]wait [.<max>]  
<użytkownik[.grupa]> <program serwera> <argumenty>
```

gdzie **usługa** jest nazwą usługi zaczerpniętą z pliku `/etc/services`, **typ gniazda** jest zwykle określany jako **stream** (gdy usługa jest strumieniowa) lub **dgram** (gdy usługa opiera się o datagramy), **protokół** jest poprawną nazwą protokołu z pliku `/etc/protocols`, zapis **wait** odnosi się do serwerów jednowątkowych (uruchamiających tylko jeden wątek, który sekwencyjnie przyjmuje wszystkie pakiety, i w końcu przechodzi w stan oczekiwania), opcjonalne pole **max** określa maksymalną liczbę procesów, które mogą być utworzone w ciągu 60 sekund, **użytkownik[.grupa]** określa nazwę użytkownika i, opcjonalnie, nazwę grupy, która powinna być wykorzystywana przez serwer, **program serwera** jest pełną ścieżką dostępu wykonywalnego programu (niektóre podstawowe usługi wymagają w tym miejscu słowa kluczowego **internal**), zaś **argumenty** to lista argumentów programu serwera (w większości przypadków pierwszy element tej listy powinien być nazwą programu serwera).

Przykładowy plik `/etc/inetd.conf`

#Usługa	Typ	Protokół	User	Program	Argumenty
discard	stream	tcp nowait	root	internal	
discard	dgram	udp wait	root	internal	
daytime	stream	tcp nowait	root	internal	
time	stream	tcp nowait	root	internal	
swat	stream	tcp nowait.400	root	/usr/sbin/tcpd	/usr/sbin/swat

Pliki `/etc/hosts.allow` i `/etc/hosts.deny`

Program *tcpd* dostarcza mechanizmów rejestracji i sterowaniem dostępem do usług, do ochrony których został skonfigurowany. W momencie uruchamiania przez demon *inetd* usługi, demon odczytuje swoje dwa pliki konfiguracyjne, zawierające zasady dostępu i albo zezwala, albo odmawia dostępu do usługi, którą ochrania.

Najpierw odczytywany jest plik `/etc/hosts.allow`, zawierający zasady dopuszczania połączeń, a następnie `/etc/hosts.deny`, zawierający zasady odrzucania. Domyślnie połączenia są akceptowane, a reguły w pliku `/etc/hosts.allow` są ważniejsze od reguł zdefiniowanych w pliku `/etc/hosts.deny`.

Format plików `/etc/hosts.allow` i `/etc/hosts.deny` jest następujący:

```
<lista usług>: <lista komputerów> [:<polecenie>]
```

gdzie **lista usług** jest listą nazw programów (oddzielonych przecinkami) obsługi chronionej, do której ma zastosowanie dana reguła, **lista komputerów** jest rozdzieloną przecinkami listą nazw komputerów, adresów IP lub jednym z określeń: **ALL** (wszystkie komputery), **LOCAL** (komputery z sieci lokalnej), **PARANOID** (komputery, których nazwa nie odpowiada ich adresowi), **EXCEPT** (wyjątki od podanej reguły), zaś **polecenie** to ścieżka dostępu do programu, który będzie wykonany, gdy reguła zostanie dopasowana.

Przykładowy plik /etc/hosts.allow

ALL: LOCAL

#Umożliwia dostęp do wszystkich usług wszystkim komputerom z sieci
#lokalnej.

Przykładowy plik /etc/hosts.deny

ALL: PARANOID

#Blokuje dostęp do wszystkich usług hostom o podejrzanych nazwach

swat: ALL

#Blokuje dostęp do usługi swat wszystkim komputerom.

#Dostęp do usługi swat będzie zatem możliwy tylko dla komputerów

#z sieci lokalnej, które mają w pliku /etc/hosts.allow umożliwiony

#dostęp do wszystkich usług.

SERWER DHCP

Protokół DHCP

Protokół dynamicznej konfiguracji hostów (DHCP, Dynamic Host Configuration Protocol) umożliwia przydzielanie żądającym hostom adresów IP z dostępnej puli. Może też dostarczać im inne informacje, w tym adres IP routera i serwerów DNS oraz nazwy domen domyślnych. Serwer DHCP może oszczędnie wydzielać dowolny podzakres adresów, ukrywając nawet przed hostami określone adresy IP.

Wszystkie sesje z serwerem rozpoczyna klient (w fazie rozruchu wstępnego), jako podstawowy element komunikacji w tym układzie. Protokół DHCP ma następujące możliwości:

- obsługuje przydziały dynamiczne
- obsługuje przydziały statyczne
- dzierżawi adresy IP
- obsługuje dzierżawy *trwale (persistent)*
- obsługuje dzierżawy *wygasłe (expired)*.

Główne obowiązki protokołu DHCP polegają na obsłudze dwóch elementów – *puli* (dostępnych adresów IP) i *dzierżaw* (przydzielonych adresów IP).

Serwer DHCP przydziela adresy IP za pomocą dzierżaw. Polegają one na przydzieleniu adresu IP dla konkretnego komputera na określony czas. Po wygaśnięciu dzierżawy, tj. osiągnięciu limitu czasu przed ponownym podłączeniem, klient musi postarać się o nowy adres. Żądanie klienta może dotyczyć konkretnego adresu IP. Jeśli jest dostępny, serwer najpewniej go przyzna. W ten sposób w środowiskach dynamicznych, można też obsługiwać dzierżawy trwałe.

Typowa konfiguracja serwera DHCP umożliwia dostarczanie przez niego klientom wszystkich parametrów niezbędnych do pracy w sieci. Jednak parametry te, wprowadzone statycznie na klientach, będą miały pierwszeństwo przed przypisywanymi przez serwer DHCP.

Konfiguracja serwera DHCP

Demon DHCP służy do przydzielania maszyną w sieci adresów IP może to robić na 2 sposoby:

- **Przydzielanie dynamiczne**

Komputer logujący się do sieci dostaje pierwszy wolny numer IP z puli określonej w pliku konfiguracyjnym. Zdarza się więc że dany host za każdym razem ma inny adres IP.

- **Przydzielanie statyczne**

Jeśli komputer logujący się do sieci jest określony w pliku konfiguracyjnym dostaje on zawsze taki sam numer IP, albo pierwszy wolny numer z puli.

Do poprawnej pracy demon ten potrzebuje pliku konfiguracyjnego */etc/dhcpd.conf*. Serwer DHCP oferuje usługę pojedynczym hostom poprzez statyczne przepisywania adresów, a całym podsieciom hostów poprzez dynamiczne przypisywanie adresów. Instrukcja *host* definiuje podstawowe parametry dla pojedynczego klienta. Instrukcja *subnet* deklaruje, że system dostarcza usługi DHCP dla podanej sieci.

Parametry pliku konfiguracyjnego

Parametry pliku konfiguracyjnego serwera DHCP – */etc/dhcpd.conf* kontrolują działanie serwera i protokołu DHCP.

Parametr	Funkcja
default-lease-time	Definiuje czas dzierżawy adresu w sekundach, jeśli klient w żądaniu nie określił czasu dzierżawy.
filename	Definiuje ścieżkę do pliku startowego dla klientów bez dysków twardych.
fixed-address	Przypisuje trwały adres IP do hosta jako część instrukcji <i>host</i> . Kilka adresów może być przypisanych klientowi, który ładuje system w kilku podsieciach.
hardware ethernet	Definiuje adres ethernetowy klienta. Parametr ten musi być częścią instrukcji <i>host</i> , która używa adresu Ethernet do przywiązania informacji hosta do określonego klienta.
max-lease-time	Definiuje maksymalny czas dzierżawy w sekundach niezależnie od czasu dzierżawy, żadanego przez klienta.
next-server	Definiuje nazwę hosta serwera, z którego ma być wgrany plik startowy. Ma to znaczenie tylko dla urządzeń bez dysków, które ładują system z serwera.
range	Definiuje zakres adresów dostępnych dla dynamicznego przypisywania.
server-name	Definiuje nazwę hosta serwera DHCP.
use-host-decl-names	Informuje serwer DHCP, aby wysłać do klienta nazwę dostarczoną w instrukcji <i>host</i> jako nazwę hosta.
use-lease-addr-for-default-route	Wysyła klientowi jego własny adres jako domyślną trasę zamiast prawdziwej domyślnej trasy. Wymusza to na kliencie używanie ARP dla wszystkich adresów.

Tab. 22. Ważniejsze parametry i ich funkcje w pliku */etc/dhcpd.conf*

Opcje podstawowe

Instrukcje opcji w serwerze DHCP obejmują wszystkie opcje konfiguracyjne DHCP zdefiniowane przez dokumenty RFC.

Podstawowe opcje definiują takie elementy, jak: adres, maska podsieci, adres domyślnego routera czy serwery DNS.

Opcja	Funkcja
option broadcast-address	Definiuje adres rozgłoszeniowy.
option dhcp-client-idetifier	Definiuje łańcuch używany do identyfikacji klientów DHCP w miejsce adresu sprzętowego.
option domain-name	Definiuje domenę nazw.
option domain-name-servers	Określa adresy serwerów DNS.
option host-name	Definiuje nazwę hosta dla klienta.
option irc-server	Określa adresy serwerów <i>Internet Relay Chat</i> (IRC).
option lpr-servers	Określa adresy serwerów wydruku.
option netbios-dd-server	Określa adresy serwerów dystrybucji datagramów NetBIOS.
option netbios-name-servers	Określa adresy IP serwerów nazw NetBIOS.
option netbios-node-type	Definiuje typ węzła NetBIOS dla klienta. Typ 1 oznacza B-węzeł NetBIOS, 2 oznacza P-węzeł, 4 oznacza M-węzeł, 8 oznacza H-węzeł.
option netbios-scope	Definiuje NetBIOS przez parametr zakresu TCP/IP.
option ntp-server	Określa adresy serwerów <i>Network Time Protocol</i> (NTP).
option pop-server	Określa adresy serwerów POP3.
option routers	Definiuje domyślną bramę sieciową.
option smtp-server	Określa adresy serwerów pocztowych SMTP.
option subnet-mask	Definiuje maskę podsieci.
option time-offset	Definiuje przesunięcie względem czasu Greenwich dla danej strefy czasowej.
option time-servers	Określa serwery czasu.

Tab. 23. Opcje podstawowe i ich funkcje w pliku */etc/dhcpd.conf*

Opcje dostrajające

Opcje dostrajające służą do dostrajania protokołu TCP/IP.

Opcja	Funkcja
option arp-cache-timeout	Definiuje, ile sekund rekordy są buforowane przez ARP.
option default ip-ttl	Definiuje domyślny czas życia dla wychodzących datagramów.
option default-tcp-ttl	Definiuje domyślny czas życia dla segmentów TCP.
option interface-mtu	Definiuje maksymalną jednostkę transmisji (MTU).
option tcp-keepalive-garbage	Określa, czy komunikaty TCP sygnalizujące aktywność (<i>keepalive messages</i>) powinny zawierać oktet bezużytecznych danych.

Tab. 24. Opcje dostrajające i ich funkcje w pliku */etc/dhcpd.conf*

Opcje routingu

Opcje routingu są związane z wyborem tras pakietów.

Opcja	Funkcja
option ip-forwarding	Informuje klienta, czy ma włączyć przekazywanie IP.
option non-local-source-routing	Informuje klienta, czy ma włączyć routing nielokalnego źródła.
option policy-filter	Określa, które pakiety pochodzące od par adres/maska sieciowa będą dopuszczalne dla klienta.
option static-routes	Definiuje listę statycznych tras dla klienta.

Tab. 25. Opcje routingu i ich funkcje w pliku */etc/dhcpd.conf*

Plik */etc/dhcpd.conf*

```
# Przykładowy plik konfiguracyjny /etc/dhcpd.conf

option domain-name "fraxinus.tk.krakow.pl";
option domain-name-servers 192.168.1.1;
option subnet-mask 255.255.255.0;
default-lease-time 21600;
max-lease-time 86400;
subnet 192.168.1.0

netmask 255.255.255.0 {
range 192.168.1.20 192.168.1.39;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
}

host merkury {
hardware ethernet 00:C0:CA:31:26:CC;
fixed-address 192.168.1.10;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
}

host wenus {
hardware ethernet 00:00:00:00:00:00;
fixed-address 192.168.1.11;
option broadcast-address 192.168.1.255
option routers 192.168.1.1;
}

host mars {
hardware ethernet 00:00:00:00:00:00;
fixed-address 192.168.1.20;
option broadcast-address 192.168.1.255
option routers 192.168.1.1;
}
```

SERWER SAMBY

Samba to pakiet narzędzi umożliwiających współdzielenie zasobów, takich jak drukarki i pliki, w sieci. Samba korzysta z protokołu *Server Message Block (SMB)*, wspólnego produktu Microsoftu i IBM-u.

Wiele systemów operacyjnych, w tym Windows i OS/2, używa SMB do komunikacji sieciowej między klientami i serwerami. Samba umożliwia uniksowym serwerom porozumiewanie się za pomocą tego samego protokołu, którego używają systemy Microsoftu. Zatem uniksowy komputer z Sambą może udawać serwer w sieci Microsoftu i udostępniać odpowiednie usługi.

Samba jest szczególnie atrakcyjna z następujących powodów:

- Używa tego samego protokołu, z którego standardowo korzystają systemy operacyjne IBM-u i Microsoftu, począwszy od DOS-a 3.0. Oznacza to, że rozumieją go niemal wszystkie komputery z systemami operacyjnymi Microsoftu i nie ma potrzeby instalowania w nich dodatkowego oprogramowania klienckiego.
- Działa na różnych platformach, w tym większości odmian Uniksa, OS/2 i NetWare. Dzięki temu jeden program na serwerze może zapewnić dostęp do plików i drukarek komputerom z różnymi systemami operacyjnymi.
- Administrowanie Sambą jest scentralizowane. Chcąc zainstalować lub uaktualnić oprogramowanie klienckie, nie trzeba tego robić na każdym komputerze z osobna.
- Samba jest kompletnym rozwiązaniem dla sieci lokalnych każdej wielkości.

Samba zapewnia przezroczyste środowisko sieciowe, które udostępnia użytkownikom wszystkie zasoby potrzebne do pracy. Po skonfigurowaniu Samba pozwala na:

- dostarczanie uniksowych plików klientom działającym pod kontrolą innych systemów operacyjnych i odwrotnie
- udostępnianie drukarek sieciowych klientom
- świadczenie usług nazewniczych (rozgłoszeniowych i WINS)
- przeglądanie zasobów sieciowych przez klientów
- tworzenie grup roboczych lub domen
- wymuszanie uwierzytelniania nazw użytkowników i haseł klientów.

Składniki pakietu Samba

Na pakiet Samba, obsługujący protokół SMB, składa się wiele komponentów. Głównymi składnikami są dwa uniksowe demony, które udostępniają współdzielone zasoby – zwane udziałami – sieciowym klientom SMB.

W czasie wstępnej konfiguracji pakietu Samba, należy zdecydować czy demony te będą uruchamiane przez system jako osobne serwery w czasie startu systemu, czy też będą uruchamiane poprzez superserwer *inetd*.

Komponent	Funkcja
smbd	Umożliwia dostęp do plików i drukarek klientom usługi SMB.
nmbd	Obsługuje zarządzanie nazwami i ich przeglądanie przez NETBIOS. Może też działać w trybie interaktywnym z zapytaniami do innych demonów serwisów nazw.
smbsh	Umożliwia korzystanie ze zdalnych udziałów podłączonych do serwera SMB jako dodatkowego zamontowanego katalogu <i>/smb</i> .
smbclient	Oferuje interfejs podobny do klienta FTP. Umożliwia on pobieranie, wysyłanie plików, pobieranie informacji o katalogach i korzystanie z drukarek podłączonych do serwerów SMB.
testparm	Umożliwia sprawdzenie pliku konfiguracyjnego <i>/etc/samba/smb.conf</i> .
testprns	Umożliwia sprawdzenie drukarek opisanych w pliku <i>/etc/printcap</i> .
smbstatus	Umożliwia wyświetlenie połączeń do serwera smbd .
nmblookup	Umożliwia wysyłanie zapytań o nazwy NETBIOS w celu otrzymania adresu IP maszyny.
smbpasswd	Oferuje zmianę zakodowanych haseł SMB, a w trybie administratora także do dodawania i usuwania użytkowników do serwera SMB (plik <i>/etc/samba/smbpasswd</i>).

Tab. 26. Składniki pakietu Samba i ich zadania

Konfiguracja Samby

Kluczem do konfigurowania Samby jest plik konfiguracyjny */etc/samba/smb.conf*. Opcje tego pliku definiują reakcje Samby na otaczającą ją sieć, zaczynając od prostych praw dostępu, a kończąc na zaszyfrowanych połączeniach i domenach NT.

Struktura pliku */etc/samba/smb.conf* składa się z kolejnych sekcji posiadających nazwy. Każda sekcja zaczyna się od jej nazwy, umieszczonej w nawiasach kwadratowych. Wszystkie zdefiniowane sekcje, z wyjątkiem sekcji **[globals]**, będą dostępne jako udział dyskowy lub drukarka każdego klienta łączącego się z serwerem Samby.

Pozostałe linie zawierają indywidualne opcje konfiguracyjne odnoszące się do określonego udziału. Opcje te rozciągają się do początku następnej sekcji albo do końca pliku. Wszystkie opcje konfiguracyjne mają format:

```
<opcja> = <wartość>
```

Opcje w pliku */etc/samba/smb.conf* ustawia się przez przypisanie im wartości.

Zmienne

Samba przechowuje obszerny zbiór zmiennych określających charakterystykę serwera i połączonych z nim klientów. Każda zmienna zaczyna się od znaku procentu, po którym następuje pojedyncza duża lub mała litera. Zmiennych można używać tylko po prawej stronie opcji konfiguracyjnej.

Rodzaj zmiennej	Zmienna	Definicja
Zmienne klienta	%a	Architektura klienta (na przykład Samba, WinNT, Win95)
	%l	Adres IP klienta
	%m	NetBIOS-owa nazwa klienta
	%M	Nazwa DNS klienta
Zmienne użytkownika	%u	Bieżąca uniksowa nazwa użytkownika
	%U	Żądana nazwa użytkownika
	%H	Katalog macierzysty %u
	%g	Podstawowa grupa %g
	%G	Podstawowa grupa %G
Zmienne udziału	%S	Nazwa bieżącego udziału
	%P	Katalog główny bieżącego udziału
	%p	Ścieżka do głównego katalogu udziału używana przez program montujący, jeśli różni się od %P
Zmienne serwera	%L	Nazwa NetBIOS-owa serwera Samby
	%h	Nazwa DNS serwera Samby
	%d	Bieżący identyfikator procesu serwera
	%N	Serwer katalogów macierzystych, ustalony na podstawie mapy programu montującego
	%v	Wersja Samby
Pozostałe zmienne	%T	Bieżący czas i data
	%R	Wynegocjowany wariant protokołu SMB

Tab. 27. Zmienne Samby

Sekcje specjalne

Sekcje specjalne pełnią ważne funkcje i różnią się od pozostałych sekcji. Do sekcji specjalnych należą:

- sekcja [globals]
- sekcja [homes]
- sekcja [printers].

Sekcja [globals] pojawia się w niemal każdym pliku konfiguracyjnym Samby, mimo że jej definiowanie nie jest obowiązkowe. Każda opcja zdefiniowana w tej opcji pliku będzie odnosić się do wszystkich pozostałych udziałów – tak, jakby zawartość sekcji została skopiowana do samego udziału. Jeśli jednak w innej sekcji opcja została zdefiniowana ponownie, nowa wartość będzie miała pierwszeństwo przed wartością z sekcji [globals].

Jeśli klient stara się połączyć z udziałem, który nie figuruje w pliku konfiguracyjnym, Samba spróbuje odszukać sekcję [homes]. Jeśli taka sekcja istnieje, niezidentyfikowana nazwa udziału zostanie uznana za uniksową nazwę użytkownika,

a Samba sprawdzi, czy taka nazwa występuje w bazie haseł serwera. Jeśli tak jest w istocie, Samba założy, że klient jest uniksowym użytkownikiem, próbującym połączyć się ze swoim katalogiem macierzystym w serwerze.

Trzecia sekcja specjalna nosi nazwę `[printers]` i przypomina sekcję `[homes]`. Jeśli klient próbuje się połączyć z udziałem, który nie jest zdefiniowany w pliku konfiguracyjnym, a jego nazwy nie można znaleźć w pliku haseł, Samba sprawdza, czy nie chodzi o udział drukarki. Odczytuje w tym celu plik parametrów drukarek (zwykle `/etc/printcap`) i sprawdza, czy występuje w nim nazwa udziału. Jeśli tak jest, Samba tworzy udział o nazwie drukarki.

Podobnie jak w przypadku sekcji `[homes]`, oznacza to, że nie trzeba tworzyć w pliku konfiguracyjnym udziału dla każdej z systemowych drukarek. Jeśli Samba zostanie odpowiednio poinformowana, będzie ona odwoływać się do uniksowego rejestru drukarek i udostępniać je klientom. Istnieje jednak oczywiste ograniczenie: jeśli konto użytkownika i drukarka nosi taką samą nazwę, Samba zawsze znajdzie najpierw konto użytkownika, nawet wtedy, gdy klient próbuje połączyć się z drukarką.

Opcje konfiguracyjne

Opcje w pliku konfiguracyjnym Samby dzielą się na dwie kategorie: globalne i dotyczące udziałów. Przynależność do kategorii warunkuje miejsca, w których może pojawić się dana opcja.

▪ Opcje globalne

Opcje globalne mogą występować tylko w sekcji `[globals]` i nigdzie indziej. Są to z reguły opcje, które wpływają na zachowanie samego serwera Samby, a nie udostępnionych przez niego udziałów.

▪ Opcje udziałów

Opcje udziałów mogą występować w poszczególnych udziałach albo w sekcji `[globals]`. Jeśli są umieszczone w sekcji `[globals]`, wówczas definiują domyślne zachowanie wszystkich udziałów, chyba że udział przypisze danej opcji nową wartość.

Kategoria	Funkcja
Wartości logiczne	Są to po prostu wartości typu „tak-nie”. Można je reprezentować za pomocą symboli: <code>yes</code> , <code>no</code> , <code>true</code> , <code>false</code> , <code>0</code> , <code>1</code> .
Wartości liczbowe	Liczby dziesiętne, szesnastkowe lub ósemkowe. Standardowa składnia <code>0xnn</code> oznacza liczbę szesnastkową, a <code>0nnn</code> – ósemkową.
Łańcuchy	Łańcuchy znaków takie jak nazwy plików lub użytkowników, w których wielkość liter jest rozróżnialna.
Listy wyliczane	Skończone listy znanych wartości. Wartość logiczna jest w istocie listą wyliczaną tylko o dwóch wartościach.

Tab. 28. Kategorie wartości opcji konfiguracyjnych

Opcje pliku konfiguracyjnego

Pliki konfiguracyjne wcale nie muszą być statyczne. Można nakazać Sambie dołączenie, a nawet zastąpienie opcji konfiguracyjnych w trakcie ich przetwarzania.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
config file	Łańcuch (pełna nazwa ze ścieżką)	Określa położenie pliku konfiguracyjnego, który należy wczytać zamiast bieżącego. Jeśli określony plik konfiguracyjny nie istnieje, opcja zostanie zignorowana, a Samba skonfiguruje się na podstawie bieżącego pliku.	Brak	Globalny
include	Łańcuch (pełna nazwa ze ścieżką)	Dołącza docelowy plik do bieżącego pliku konfiguracyjnego. Jeśli wskazany plik konfiguracyjny nie istnieje, opcja zostanie zignorowana. Opcja nie rozpoznaje zmiennych %u, %p ani %s, ponieważ zmienne te nie są jeszcze ustawione w momencie odczytywania pliku.	Brak	Globalny
copy	Łańcuch (nazwa udziału)	Umożliwia powielenie opcji innego udziału w bieżącym udziale. Wskazany udział musi występować w pliku konfiguracyjnym wcześniej niż ten, do którego kopiowane są opcje. Wszystkie opcje w udziale, będą miały pierwszeństwo przed powielanymi opcjami niezależnie od tego, czy występują przed czy za tą dyrektywą.	Brak	Udział

Tab. 29. Opcje pliku konfiguracyjnego

Opcje konfiguracyjne serwera

Opcje konfiguracyjne serwera określają parametry samego serwera Samby.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
netbios name	Łańcuch	Ustawia NetBIOS-ową nazwę serwera. Najczęściej służy do przypisania Sambie nazwy NetBIOS-owej różnej od bieżącej nazwy DNS lub przeniesienie usług SMB z uszkodzonego komputera.	Nazwa hosta DNS	Globalny
server string	Łańcuch	Ustawia tekst opisujący serwer Samby.	Samba %v	Globalny
workgroup	Łańcuch	Ustawia NetBIOS-ową nazwę grupy roboczej, w której serwer będzie ogłaszał swoją obecność.	WORKGROUP	Udział

Tab. 30. Opcje konfiguracyjne serwera

Opcje konfiguracyjne udziałów dyskowych

Opcje konfiguracyjne udziałów dyskowych określają jakie udziały i z jakimi prawami dla różnych klientów będą dostępne.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
path (directory)	Łańcuch (pełna nazwa ze ścieżką)	Ustawia ścieżkę do głównego uniksowego katalogu, który będzie udostępniony jako udział dyskowy albo używany jako katalog buforowy drukarki. Można wybrać dowolny katalog w serwerze Samby pod warunkiem, że użytkownik łączącego się procesu ma prawo do odczytu i zapisu w tym katalogu. Jeśli ścieżka odnosi się do drukarki, powinna wskazywać na katalog tymczasowy, w którym można zapisać pliki przed ich wysłaniem do bufora docelowej drukarki.	/tmp	Udział
guest ok (public)	Wartość logiczna	Umożliwia lub uniemożliwia „gościnny” dostęp do udziału. W przypadku zmiany domyślnej wartości, w celu uzyskania dostępu do udziału, nie trzeba będzie podawać nazwy użytkownika ani hasła. Kiedy użytkownik połączy się z serwerem, jego prawa dostępu będą równoważne prawom wyznaczonego użytkownika-gościa. Domyślne konto użytkownika to nobody .	no	Udział
guest account	Łańcuch (nazwa użytkownika)	Określa nazwę nieuprzywilejowanego konta uniksowego, które będzie wykorzystywane podczas drukowania i udostępniania udziałów oznaczonych opcją guest ok .	Brak	Udział
comment	Łańcuch	Ustawia komentarz, który będzie wysyłany do klientów próbujących przejrzeć zawartość danego zasobu.	Brak	Udział
volume	Łańcuch	Umożliwia określenie nazwy wolumenu udziału zgłaszanej przez Sambę.	Nazwa udziału	Udział
read only	Wartość logiczna	Określa, czy dany zasób jest przeznaczony tylko do odczytu.	yes	Udział
writeable (write ok)	Wartość logiczna	Określa, czy możliwy jest zapis w danym zasobie. Opcje read only i writeable to dwa sposoby na określenie tego samego, choć z przeciwnej pozycji. W przypadku użycia którejś z tych opcji więcej niż raz, Samba przyjmuje ostatnią zdefiniowaną wartość.	no	Udział

Tab. 31. Podstawowe opcje konfiguracyjne udziałów

Opcje sieciowe

Opcje sieciowe są szczególnie istotne, gdy Samba działa w serwerze połączonym z różnymi podsieciami lub gdy ważnym elementem jest wdrożenie polityki bezpieczeństwa w danej podsieci.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
hosts allow (allow hosts)	Łańcuch (lista nazw hostów)	Określa komputery, które mogą korzystać z udziałów w serwerze Samby. Należy pamiętać, by jawnie zezwolić na korzystanie z udziałów adresowi pętli zwrotnej, gdyż można w ten sposób uniemożliwić serwerowi Samby komunikacji z samym sobą. W tej można używać: nazwy hostów; adresów IP; nazw domen, które można oddzielić od nazw hostów; grup sieciowych, których nazwy rozpoczynają się od znaku @ (systemy używające NIS); podsieci, które kończą się kropką; słowa kluczowe ALL (zezwala na dostęp wszystkim klientom) i EXCEPT (wyjątki od reguł).	Brak	Udział
hosts deny (deny hosts)	Łańcuch (lista nazw hostów)	Określa komputery, które nie mają zezwolenia na dostęp do udziałów. Do określenia klientów można użyć tego samego, co w opcji hosts allow . Jeśli zostanie umieszczona w sekcji [globals] , będzie ona miała pierwszeństwo przed opcjami hosts deny zdefiniowanymi w udziałach.	Brak	Udział
interfaces	Łańcuch (kombinacje adres IP/maska sieciowa)	Wymienia interfejsy sieciowe, które Samba będzie rozpoznawać i przez które będzie odpowiadać. Jeśli opcja nie jest ustawiona, Samba w trakcie uruchamiania wyszukuje podstawowy interfejs serwera i konfiguruje się do obsługi tej jednej podsieci.	Zależna od systemu	Globalny
bind interfaces only	Wartość logiczna	Sprawia, że procesy <i>smbd</i> i <i>nmbd</i> obsługują żądania tylko z tych podsieci, które są wymienione w opcji interfaces .	no	Globalny
socket address	Łańcuch (adres IP)	Określa, pod którymi adresami podanymi w opcji interfaces Samba będzie czekać na połączenia.	Brak	Globalny

Tab. 32. Opcje konfiguracji sieci

Opcje serwerów wirtualnych

Serwery wirtualne tworzą iluzję obecności wielu serwerów NetBIOS-u w sieci, choć w rzeczywistości jest tylko jeden taki serwer. Uzyskanie takiego efektu nie jest trudne: komputer po prostu rozgłasza więcej niż jedną NetBIOS-ową nazwę w połączeniu ze swoim adresem IP. Metoda ta przynosi wymierne korzyści.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
netbios aliases	Lista nazw NetBIOS-owych	Uaktywnia podane dodatkowe nazwy NetBIOS-owe, na które powinien reagować serwer, używane do tworzenia „wirtualnych” serwerów Samby. Każda podana nazwa NetBIOS-owa zostanie wyświetlona przez komputer przeglądający zasoby sieci. Kiedy jednak klient spróbuje nawiązać połączenie, połączy się z tym samym serwerem Samby.	Brak	Globalny

Tab. 33. Opcje konfiguracji wirtualnych serwerów

Opcje konfiguracji rejestrowania

Czasem trzeba sprawdzić, co właściwie robi Samba, zwłaszcza wtedy, gdy przeprowadzi ona nieoczekiwaną operację lub w ogóle przestanie działać. Aby uzyskać niezbędne informacje, należy przejrzeć pliki dziennika Samby i sprawdzić, czemu wykonała dane działanie.

Samba zawiera sześć opcji umożliwiających określenie sposobu i miejsca rejestrowania informacji. Każda z nich ma zasięg globalny i nie może pojawić się w definicji udziału.

Jeśli oprócz lub zamiast standardowego dziennika Samby, zamierza się używać systemowego programu rejestrującego (*syslogd*), można skorzystać z przeznaczonych do tego opcji. W tym celu należy upewnić się, że Samba została odpowiednio skompilowana. Następnie trzeba zmodyfikować plik */etc/syslog.conf* tak, aby komunikaty Samby były akceptowane przez *syslogd*. Jeśli takowa nie istnieje, to należy do pliku */etc/syslog.conf* dopisać następującą linię:

```
daemon.* /var/log/daemon.log
```

Dzięki temu wszystkie komunikaty od demonów systemowych będą zapisywane w pliku */var/log/daemon.log*. Tam również trafią komunikaty Samby.

Później można zdecydować, czy komunikaty będą wysyłane tylko do programu rejestrującego, czy też będą używały zarówno standardowego dziennika Samby oraz demona *syslogd*.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
log file	Łańcuch (nazwa pliku wraz z pełną ścieżką)	Ustawia nazwę i położenie pliku dziennika używanego przez Sambę. Rozpoznaje standardowe zmienne. Domyślne położenie pliku dziennika można zmienić za pomocą opcji linii polecenia -l podczas uruchamiania obu demonów, która jednak nie ma pierwszeństwa przed opcją log file.	/var/log/samba/ /log.%m	Globalny
log level (debug level)	Wartość liczbowa (0-10)	Określa ilość komunikatów informacyjnych i diagnostycznych wysyłanych do pliku dziennika. Zwykle nadaje się jej wartość 0 lub 1. Poziomy powyżej 3 zapisują informacje przydatne głównie dla programistów i znacznie spowalniają serwer.	1	Globalny
max log size	Wartość liczbowa (rozmiar w KB)	Ustawia maksymalny rozmiar pliku dziennika. Kiedy dziennik przekroczy ten rozmiar, otrzyma rozszerzenie <i>.old</i> i utworzony zostanie nowy plik dziennika. Jeśli istnieje już plik o przyrostku <i>.old</i> zostanie on usunięty.	5000	Globalny
debug timestamp (timestamp logs)	Wartość logiczna	Określa, czy w plikach dziennika umieszczać znaczniki czasowe.	yes	Globalny
syslog	Wartość liczbowa (0-10)	Ustawia poziom komunikatów wysyłanych do programu <i>syslogd</i> . Komunikaty o poziomie niższym od podanej liczby będą wysyłane do rejestratora systemowego, natomiast informacje o poziomie równym lub większym od wartości opcji <i>syslog</i> będą nadal zapisywane w dziennikach Samby.	1	Globalny
syslog only	Wartość logiczna	Określa, czy do zapisywania komunikatów używany jest tylko rejestrator systemowy. W takim wypadku w standardowych dziennikach Samby nie są zapisywane żadne dane.	no	Globalny

Tab. 34. Opcje konfiguracji rejestrowania

Opcje przeglądania

Dzięki przeglądaniu można stwierdzić, jakie serwery i udziały są obecnie dostępne w sieci. Jeden z komputerów w każdej podsieci przechowuje listę wszystkich aktywnych komputerów. Lista ta nosi nazwę listy przeglądania, a przechowujący ją serwer jest nazywany główną przeglądarką lokalną. W miarę dołączania i odłączania komputerów, główna przeglądarka lokalna uaktualnia informacje na liście przeglądania i udostępnia je żądającym tego komputerom.

Komputer staje się główną przeglądarką lokalną w wyniku wyborów ogłaszanych w lokalnej podsieci. Wybory przeglądarki mogą zostać ogłoszone w dowolnym momencie. Samba może dowolnie fałszować wybory, na przykład po to, aby zawsze zostawać główną przeglądarką lokalną lub nigdy nie przejmować tej funkcji.

Każdy komputer biorący udział w wyborach rozgłasza informacje o sobie. Informacje te obejmują:

- wersję używanego protokołu elekcyjnego
- system operacyjny komputera
- czas, od którego komputer jest w sieci
- nazwę hosta.

System operacyjny	Wartość
Windows XP Server	64
Windows XP Professional	16
Windows 2000 Server	32
Windows 2000 Professional	16
Windows NT Server 4.0	32
Windows NT Server 3.51	32
Windows NT Workstation 4.0	16
Windows NT Workstation 3.51	16
Windows ME	1
Windows 98	1
Windows 95	1
Windows 3.1 for Workgroups	1

Tab. 35. Wartości przypisywane systemom operacyjnym w wyborach przeglądarki

Główna przeglądarka domeny rozpowszechnia listy przeglądania między wszystkimi podsieciami w grupie roboczej. Każda główna przeglądarka lokalna okresowo synchronizuje swoją listę przeglądania z główną przeglądarką domeny. Podczas synchronizacji, przeglądarka lokalna przekazuje przeglądarce domeny informacje o wszystkich serwerach, których ta nie ma na swojej liście przeglądania i odwrotnie.

Inaczej niż w przypadku przeglądarek lokalnych, nie przeprowadza się wyborów w celu wyłonienia głównej przeglądarki domeny. Musi wyznaczyć ją administrator.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
announce as	NT, WfW lub Win95	Ustawia system operacyjny, za który będzie podawać się Samba.	NT	Globalny
announce version	Wartość liczbowa	Ustawia wersję systemu operacyjnego ogłaszaną przez Sambę.	4.2	Globalny
browseable	Wartość logiczna	Umożliwia wyświetlanie udziału na liście zasobów komputera.	yes	Udział
browse list	Wartość logiczna	Określa, czy serwer Samby będzie udostępniał listę przeglądania.	yes	Globalny
auto services (preload)	Łańcuch (lista udziałów)	Określa listę udziałów, które będą zawsze obecne na liście przeglądania. Najczęściej używa się jej do ogłaszania konkretnych udziałów.	Brak	Globalny
default service (default)	Łańcuch (nazwa udziału)	Określa udział (usługę), który zostanie udostępniony, jeśli klient zażąda dostępu do udziału nie zdefiniowanego w pliku konfiguracyjnym. Nazwy udziału nie umieszcza tutaj się w nawiasach kwadratowych.	Brak	Globalny
local master	Wartość logiczna	Określa, czy Samba ma próbować zostać główną przeglądarką lokalną.	yes	Globalny
lm announce	yes, no lub auto	Określa, czy Samba ma wysyłać ogłoszenia LAN Managera o goście. Takich ogłoszeń mogą wymagać starsze klienty (na przykład OS/2).	auto	Globalny
lm interval	Wartość liczbowa	Określa częstotliwość (w sekundach), z jaką będą ponawiane ogłoszenia LAN Managera.	60	Globalny
preferred master	Wartość logiczna	Określa, czy Samba ma użyć bitu preferowanej przeglądarki głównej, próbując zostać główną przeglądarką lokalną. Dzięki temu komputer uzyskuje wyższą preferencję od komputerów o tym samym poziomie systemu operacyjnego.	no	Globalny
domain master	Wartość logiczna	Określa, czy Samba ma próbować zostać główną przeglądarką domeny w swojej grupie roboczej.	no	Globalny
os level	Wartość liczbowa	Ustawia poziom systemu operacyjnego Samby podczas wyboru głównej przeglądarki lokalnej.	0	Globalny
remote browse sync	Łańcuch (adresy IP)	Wymienia serwery Samby, z którymi należy synchronizować listy przeglądania.	Brak	Globalny
remote announce	Łańcuch (adres/grupa robocza)	Wymienia podsieci i grupy robocze, do których należy wysłać ukierunkowane rozgłoszenia, aby umieścić serwer Samby na listach przeglądania.	Brak	Globalny

Tab. 36. Opcje konfiguracji przeglądania

Opcje systemu plików

Jednym z najpoważniejszych zadań stojących przed Sambą jest korygowanie różnic między uniksowymi i nieunikсовymi systemami plików. Chodzi tu o obsługę dowiązań symbolicznych i plików ukrytych.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
unix realname	Wartość logiczna	Udostępnia klientowi pełną nazwę uniksowego użytkownika.	no	Globalny
dont descend	Łańcuch (lista katalogów)	Określa listę katalogów, których zawartość powinna być niewidoczna dla klientów.	Brak	Udział
follow symlinks	Wartość logiczna	Określa, czy Samba ma blokować używanie dowiązań symbolicznych.	yes	Udział
getwd cache	Wartość logiczna	Określa, czy Samba będzie używać lokalnego bufora dla uniksowego wywołania systemowego zwracającego nazwę bieżącego katalogu roboczego.	yes	Globalny
wide links	Wartość logiczna	Określa, czy Samba będzie podążać za dowiązaniem na zewnątrz udziału. Dotyczy to wszystkich plików i katalogów, jeśli tylko użytkownik ma odpowiednie prawa dostępu.	yes	Udział
hide dot files	Wartość logiczna	Określa, czy uniksowe pliki ukryte będą traktowane jako ukryte także w Windows.	yes	Udział
hide files	Łańcuch (lista plików)	Określa wzorce nazw katalogów i plików, które przez klientów będą traktowane jako ukryte. Każdy wpis na liście musi zaczynać się, kończyć lub być oddzielony od innego wpisu znakiem ukośnika (/). Można używać gwiazdek, reprezentujących 0 lub więcej znaków oraz znaków zapytania, reprezentujących dokładnie jeden znak.	Brak	Udział
veto files	Łańcuch (lista plików)	Określa wzorce nazw katalogów i plików, które nigdy nie będą wyświetlane dla klientów. Opcja ta ma taką samą składnię, jak opcja hide files.	Brak	Udział
delete veto files	Wartość logiczna	Określa, czy podczas usuwania katalogu zawierającego zawetowane pliki, mają zostać usunięte zawetowane pliki.	no	Udział

Tab. 37. Opcje konfiguracji systemu plików

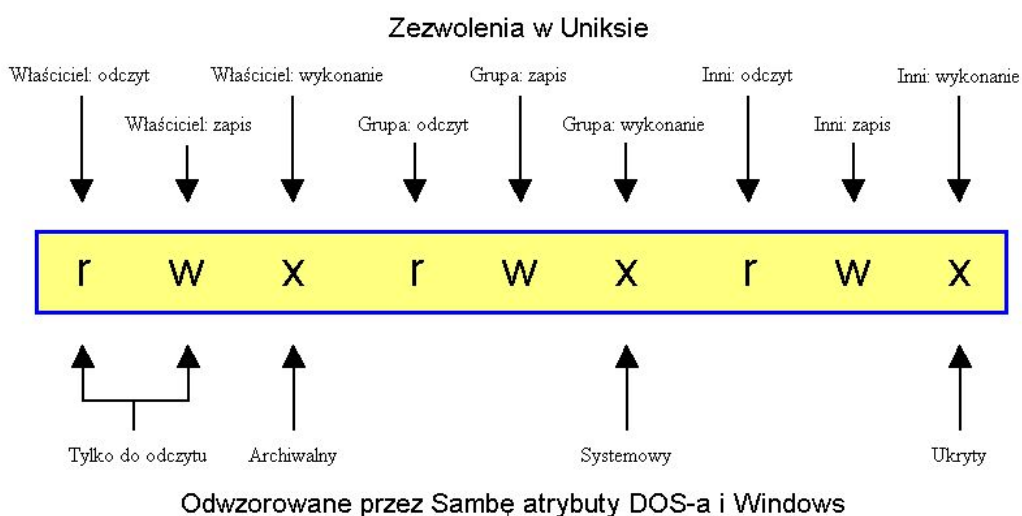
Opcje praw dostępu do plików i katalogów

DOS nigdy nie miał być wielodostępnym, sieciowym systemem operacyjnym, natomiast Unix był zaprojektowany w ten sposób od samego początku. W obsłudze ich systemów plików występuje więc sporo rozbieżności i luk. Jedną z największych różnic między Uniksem i DOS-em jest obsługa praw dostępu do plików.

Wszystkie pliki Uniksa mają zezwolenia na odczyt, zapis i wykonanie dla trzech kategorii użytkowników: właściciela, grupy i pozostałych użytkowników.

Windows dysponuje natomiast czterema bitami, które przypisuje wszystkim plikom atrybuty:

- **tylko do odczytu**
użytkownik może odczytać zawartość pliku, ale nie może nic w nim zapisać
- **systemowy**
spełnia specjalną funkcję w systemie operacyjnym
- **ukryty**
niewidoczny dla użytkownika, chyba że ten jawnie zażąda jego wyświetlenia
- **archiwalny**
plik został zmodyfikowany od czasu sporządzenia jego dosowej kopii zapasowej



Rys. 14. Znaczenie praw dostępu do pliku w Uniksie i Sambie

Samba może zachowywać dosowe atrybuty plików, wykorzystując bity wykonywalności pliku uniksowego. Bity odczytu i zapisu dla grupy i innych użytkowników, nie przekładają się bezpośrednio na atrybuty DOS-a, ale zachowują swoje pierwotne uniksowe znaczenie w serwerze Samby.

Kolejną cechą Uniksa, której brakuje DOS-owi, jest możliwość usunięcia z zapisywalnego katalogu pliku przeznaczonego tylko do odczytu. Jeśli uniksowy katalog jest zapisywalny, można usuwać z niego pliki przeznaczone tylko do odczytu. Dzięki temu użytkownicy mogą usuwać dowolne pliki ze swoich katalogów, nawet wtedy, gdy umieścił je tam ktoś inny.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
map archive	Wartość logiczna	Zachowuje dosowy atrybut archiwalny w bicie wykonywalności dla użytkownika (0100).	yes	Udział
map system	Wartość logiczna	Zachowuje dosowy atrybut systemowy w bicie wykonywalności dla grupy (0010).	no	Udział
map hidden	Wartość logiczna	Zachowuje dosowy atrybut ukrycia w bicie wykonywalności dla pozostałych użytkowników (0001).	no	Udział
create mask (create mode)	Wartość liczbowa	Ustawia maksymalną wartość zezwoleń dla plików tworzonych przez Sambę. Argumentem tej opcji jest ósemkowa liczba określająca zezwolenia, które mogą zostać ustawione podczas tworzenia pliku przez klienta.	0744	Udział
directory mask (directory mode)	Wartość liczbowa	Ustawia maksymalną wartość zezwoleń dla katalogów tworzonych przez Sambę. Ma taką samą składnię, jak opcja create mask.	0755	Udział
force create mode	Wartość liczbowa	Wymusza ustawienie określonych zezwoleń (bitowa suma logiczna OR) dla plików tworzonych przez Sambę. Opcja ta jest zawsze uwzględniana po opcjach map archive, map system, map hidden i create mask.	0000	Udział
force directory mode	Wartość liczbowa	Wymusza ustawienie określonych zezwoleń (bitowa suma logiczna OR) dla katalogów tworzonych przez Sambę. Można używać tej opcji dokładnie tak samo, jak opcji force create mode, aby w razie potrzeby dodawać zezwolenia grupowe lub inne.	0000	Udział
force group (group)	Łańcuch (nazwa grupy)	Ustawia obowiązującą grupę, która po uwierzytelnieniu klienta będzie używana we wszystkich połączeniach z udziałem.	Brak	Udział
force user	Łańcuch (nazwa użytkownika)	Ustawia obowiązującą nazwę użytkownika, który po uwierzytelnieniu klienta będzie używany we wszystkich połączeniach z udziałem.	Brak	Udział
delete readonly	Wartość logiczna	Pozwala użytkownikowi usunąć z zapisywalnego katalogu pliki przeznaczone tylko do odczytu.	no	Udział

Tab. 38. Opcje praw dostępu do plików i katalogów

Opcje przekształcania nazw

W systemach DOS i Windows 3.1 każda nazwa pliku może składać się co najwyżej z ośmiu dużych liter, kropki i kolejnych trzech dużych liter (*format 8.3*). Samba musi zachować zgodność wstecz z klientami sieciowymi, które przechowują pliki tylko w formacie 8.3. Samba musi stosować specjalną metodę tłumaczenia długich nazw plików na nazwy 8.3, w taki sposób, aby podobne nazwy nie powodowały kolizji.

Samba przekształca długą nazwę pliku na nazwę w formacie 8.3 w następujący sposób:

- Jeśli pierwotna nazwa pliku nie zaczyna się od kropki, najwyżej pięć pierwszych znaków alfanumerycznych występujących przed ostatnią kropką (jeśli kropka występuje w nazwie) jest przekształcanych na duże litery. Stanowią one pierwsze pięć znaków przekształconej nazwy 8.3.
- Jeśli pierwotna nazwa pliku zaczyna się od kropki, kropka jest usuwana, a najwyżej pięć pierwszych znaków alfanumerycznych występujących przed ostatnią kropką (jeśli kropka występuje w nazwie) jest przekształcanych na duże litery. Stanowią one pierwsze pięć znaków przekształconej nazwy 8.3.
- Za tymi znakami umieszczany jest specjalny znak przekształcenia: domyślnie jest to tylda (~), choć Samba pozwala na użycie innego znaku.
- Podstawowa część długiej nazwy pliku przed ostatnią kropką jest zamieniana za pomocą funkcji mieszającej na dwuznakowy kod; w razie potrzeby wykorzystywane są części nazwy znajdujące się za ostatnią kropką. Ten dwuznakowy kod jest dołączony do nazwy 8.3 za znakiem przekształcenia.
- Pierwsze trzy znaki pierwotnej nazwy pliku znajdujące się za ostatnią kropką (jeśli kropka występuje w nazwie) są zamieniane na duże litery i dołączane do przekształconej nazwy jako rozszerzenie. Jeśli pierwotna nazwa pliku zaczynała się od kropki, zamiast rozszerzenia zostaną użyte trzy znaki podkreślenia.

Dzięki tym regułom Windows for Workgroups będzie rozróżniać pliki na potrzeby użytkowników, którzy korzystają z tego systemu operacyjnego.

Opcje konfiguracji przekształcania przydają się właściwie tylko do współpracy z najstarszymi klientami. Jeśli opcje te mają być użyte najlepiej ustawić je, bez zakłócania pracy innych klientów, za pomocą dyrektywy **include**.

Systemy operacyjne mogą inaczej reprezentować nazwy plików, a inaczej je ustalać. Rodzina systemów operacyjnych Windows 95/98/NT ustala nazwy plików bez uwzględniania wielkości liter, choć reprezentuje je z rozróżnieniem małych i dużych liter. Systemy uniksowe zawsze ustalają nazwy plików z rozróżnieniem wielkości liter.

Samba pozwala na dostosowanie przekształcania nazw do potrzeb użytkownika, w tym na kontrolowanie rozróżniania wielkości liter, na określenie znaku używanego do tworzenia przekształconych nazw oraz na ręczne odwzorowywanie nazw plików między dwoma formatami.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
case sensitive (casesignames)	Wartość logiczna	Określa, czy Samba powinna zachowywać wielkość liter podczas ustalania nazw w konkretnym udziale.	no	Udział
default case	upper lub lower	Określa domyślną wielkość liter, której Samba użyje podczas tworzenia pliku w imieniu klienta.	lower	Udział
preserve case	Wartość logiczna	Definiuje, czy w nazwach plików tworzonych przez Sambę na zlecenie klienta będą występowały litery o wielkości określonej przez system operacyjny klienta.	yes	Udział
short preserve case	Wartość logiczna	Określa, czy w nazwach plików formatu 8.3 tworzonych przez Sambę na zlecenie klienta będą występowały litery o wielkości określonej przez system operacyjny klienta.	yes	Udział
mangle case	Wartość logiczna	Informuje Sambę, czy należy przekształcać nazwy plików nie składające się w całości z liter o wielkości określonej opcją default case.	no	Udział
mangled names	Wartość logiczna	Określa, czy Samba będzie przekształcać nazwy plików w swoich udziałach na potrzeby klientów posługujących się formatem 8.3.	yes	Udział
mangling char	Łańcuch (pojedynczy znak)	Określa znak używany podczas przekształcania nazw plików na format 8.3.	~	Udział
mangled stack	Wartość liczbowa	Określa liczbę przekształconych nazw przechowywanych na lokalnym stosie przekształcania. Stos ten można wykorzystać do przywracania pierwotnej postaci przekształconych nazw plików.	50	Globalny
mangled map	Łańcuch (lista wzorców)	Pozwala na określenie własnych wzorców przekształcania nazw plików między dwoma formatami. Jest to przydatne wtedy, gdy domyślny algorytm przekształca nazwy nieprawidłowo lub na format, którego klient nie może rozpoznać. Poszczególne wzorce oddziela się znakami odstępu.	Brak	Udział

Tab. 39. Opcje przekształcania nazw

Opcje konfiguracji blokad

Jednoczesne zapisy w jednym pliku są w każdym systemie operacyjnym rzeczą niepożądaną. Aby temu zapobiec, większość systemów używa blokad, które gwarantują, że w danej chwili w pliku może pisać tylko jeden proces. Jeśli inny proces będzie próbował zapisać coś w zablokowanym już pliku, system operacyjny zgłosi błąd i proces będzie musiał poczekać na zwolnienie blokady.

Samba obsługuje standardowe żądania blokady DOS-a i systemu plików NT w trybie odmowy (*deny-mode*). Ponadto Samba obsługuje nowy mechanizm blokowania, nazywany blokadą oportunistyczną.

Dzięki blokadom oportunistycznym klient może poinformować serwer Samby, że nie tylko będzie jedynym uprawnionym do pisania w pliku, ale że będzie buforował wszystkie zmiany lokalne (a nie w serwerze Samby) w celu przyspieszenia dostępu do pliku. Kiedy Samba wie, że plik został oportunistycznie zablokowany przez klienta, zaznacza swoją wersję tego pliku jako obłożoną blokadą i czeka, aż klient zakończy operacje na pliku i odeśle jego ostateczną wersję w celu wzajemnej synchronizacji. Jeśli inny klient zażąda dostępu do tego pliku, zanim pierwszy klient zakończy pracę, Samba może wysłać żądanie przerwania blokady do pierwszego klienta. Jest to informacja dla klienta, że powinien zaprzestać lokalnego buforowania i zwrócić informacje o bieżącym stanie pliku, aby nowy klient mógł go użyć. Blokada oportunistyczna nie jest jednak zamiennikiem standardowej blokady w trybie odmowy.

Systemy Windows potrafią ze sobą współpracować tak, aby uniknąć nadpisania zmian dokonanych w pliku przez innego klienta. Jeśli jednak do pliku przechowywanego w serwerze Samby uzyska dostęp proces uniksowy, nie będzie on wiedział o oportunistycznym blokowaniu używanym w Windows i naruszy tę blokadę. Może się zatem zdarzyć, że ktoś uruchomi uniksowy proces czytający lub piszący w pliku, z którego korzysta również Windows, co spowoduje uszkodzenie danych.

Najprostsze blokady udostępniane przez Sambę to blokady w trybie odmowy, tak zwane tryby współdzielenia, z których korzystają programy, takie jak edytor tekstów, aby uniknąć przypadkowego nadpisania plików.

Blokada	Opis
DENY_NONE	Nie odrzuca innych żądań dostępu do pliku.
DENY_ALL	Odrzuca wszystkie żądania otwarcia pliku.
DENY_READ	Odrzuca wszystkie żądania otwarcia pliku w trybie tylko do odczytu.
DENY_WRITE	Odrzuca wszystkie żądania otwarcia pliku w trybie tylko do zapisu.
DENY_DOS	Jeśli plik jest otwarty tylko do odczytu, inni mogą czytać plik, ale nie mogą w nim pisać. Jeśli plik jest otwarty do zapisu, inni w ogóle nie mogą otworzyć pliku.

Tab. 40. Blokady Samby w trybie odmowy

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
share modes	Wartość logiczna	Określa, czy mają być użyte blokady w trybie odmowy.	yes	Udział
locking	Wartość logiczna	Określa, czy należy włączyć blokowanie zakresów bajtów po stronie serwera za pomocą zwykłych uniksowych blokad doradczych. Zapobiega to nadpisaniu zablokowanego zakresu bajtów przez poprawnie działające procesy uniksowe.	yes	Udział
strict locking	Wartość logiczna	Określa, czy przy każdym dostępie do pewnego zakresu bajtów Samba ma sprawdzać, czy nie jest na niego nałożona blokada.	no	Udział
oplocks	Wartość logiczna	Określa, czy mają być włączone blokady oportunistyczne.	yes	Udział
kernel oplocks	Wartość logiczna	Określa, czy jądro systemu operacyjnego, na którym pracuje Samba, obsługuje blokady oportunistyczne.	yes	Globalny
fake oplocks	Wartość logiczna	Określa, czy demony Samby mają udawać, że przyznają blokady oportunistycznie. Opcja ta jest przestarzała, ponieważ Samba obsługuje blokady oportunistyczne.	no	Udział
blocking locks	Wartość logiczna	Określa, czy Samba ma pozwolić zaczekać klientowi żądającemu blokady na jej przyznanie (tzw. blokady wstrzymujące).	yes	Udział
veto oplocks files	Łańcuch (lista nazw plików)	Definiuje listę plików, na które nigdy nie będą nakładane blokady oportunistyczne. Każdy wpis na liście musi zaczynać się, kończyć lub być oddzielony od innego wpisu znakiem ukośnika (/). Można używać gwiazdek, reprezentujących 0 lub więcej znaków oraz znaków zapytania, reprezentujących dokładnie jeden znak.	Brak	Udział
lock directory	Łańcuch (pełna ścieżka dostępu)	Określa położenie katalogu, w którym Samba przechowuje pliki blokad SMB, listy przeglądania, plik pamięci dzielonej oraz baza danych WINS.	/var/run/samba	Globalny

Tab. 41. Opcje konfiguracji blokad zwykłych i oportunistycznych

Opcje kontroli dostępu

Ze względów bezpieczeństwa, często będzie zachodziła potrzeba ograniczania dostępu do konkretnych udziałów.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
admin users	Łańcuch (lista nazw użytkowników)	Określa listę użytkowników, którzy mogą wykonywać operacje z przywilejami administratora. Oznacza to, że mogą zmodyfikować lub usunąć pliki innych użytkowników, niezależnie od obowiązujących praw dostępu. Opcja ta pozwala także na zarządzanie udziałami.	Brak	Udział
valid users	Łańcuch (lista nazw użytkowników)	Określa listę użytkowników, którzy mogą łączyć się z udziałem.	Brak	Udział
invalid users	Łańcuch (lista nazw użytkowników)	Określa listę użytkowników, którzy nie mogą łączyć się z udziałem. Każdy użytkownik lub grupa z listy tej opcji, zawsze spotka się z odmową dostępu do udziału, nawet jeśli figuruje na liście valid users.	Brak	Udział
read list	Łańcuch (lista nazw użytkowników)	Podaje listę użytkowników, którzy mogą tylko odczytywać pliki z udziału. Opcja ta ma pierwszeństwo przed innymi uprawnieniami przyznawanymi przez Sambę, a także przed prawami dostępu w systemie uniksowym i uniemożliwia zapisywanie plików w udziale.	Brak	Udział
write list	Łańcuch (lista nazw użytkowników)	Podaje listę użytkowników, którzy mogą zapisywać pliki w udziale przeznaczonym tylko do odczytu.	Brak	Udział
max connections	Wartość liczbowa	Określa maksymalną liczbę jednoczesnych połączeń z udziałem.	0	Udział
guest only	Wartość logiczna	Określa, czy dany udział ma zezwalać tylko na dostęp gościnny. Udział, do którego stosuje się ta opcja, musi zawierać wpis guest only = yes.	no	Udział
guest account	Łańcuch (nazwa konta)	Określa uniksowe konto, które będzie używane do dostępu gościnnego.	nobody	Udział

Tab. 42. Opcje kontroli dostępu do udziału

Opcje nazw użytkowników

Opcje nazw użytkowników są stosowane przez Sambę do korygowania niezgodności między nazwami użytkowników w systemie uniksowym i Windows.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
username map	Łańcuch (pełna ścieżka dostępu do pliku)	Ustawia nazwę pliku z odwzorowaniami nazw użytkowników. Odwzorowania są stosowane w celu przekształcenia nazwy użytkowników klientów (mogą one zawierać do 255 znaków) na nawy uniksowe (zazwyczaj do 8 znaków). Każdy wpis w pliku odwzorowań powinien składać się z uniksowej nazwy użytkownika, znaku równości i jednej lub wielu nazw użytkowników klientów SMB, oddzielonych znakami odstępu.	Brak	Globalny
username level	Wartość liczbowa	Określa liczbę liter, które będą zamieniane w nazwie użytkownika na duże litery podczas próby połączenia z serwerem.	0	Globalny

Tab. 43. Opcje nazw użytkowników

Opcje uwierzytelniania

Każdy użytkownik, który próbuje połączyć się z udziałem nie dopuszczającym gościnnego dostępu, musi podać hasło, aby pomyślnie nawiązać połączenie. Operacje, które Samba przeprowadza na tym hasle zależą od opcji uwierzytelniania.

Istnieją 4 poziomy zabezpieczeń, które Samba może obsługiwać w swojej podsieci:

- **Zabezpieczenia na poziomie udziału**

Każdy udział w grupie roboczej jest chroniony oddzielnym hasłem. Każdy, kto zna hasło dostępu do udziału, może z niego korzystać.

- **Zabezpieczenia na poziomie użytkownika**

Każdy udział w grupie roboczej jest skonfigurowany tak, aby pozwalać na dostęp tylko określonym użytkownikom. Po nawiązaniu połączenia z zasobem, serwer Samby weryfikuje użytkowników i ich hasła zanim przyzna im dostęp do udziału.

- **Zabezpieczenia na poziomie serwera**

Samba używa oddzielnego serwera SMB, który zatwierdza użytkowników i ich hasła przed przyznaniem dostępu do udziału.

- **Zabezpieczenia na poziomie domeny**

Samba staje się członkiem domeny Windows i uwierzytelnia klienty za pośrednictwem podstawowego kontrolera domeny. Po uwierzytelnieniu użytkownik otrzymuje specjalny żeton, który umożliwia mu korzystanie ze wszystkich udziałów, do których ma odpowiednie uprawnienia.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
security	domain, server, share lub user	Określa poziom bezpieczeństwa serwera Samby.	user	Globalny

Tab. 44. Opcje uwierzytelniania

Opcje konfiguracji haseł

Hasła wysyłane przez poszczególne klienty mogą być albo zaszyfrowane, albo niezaszyfrowane. Zaszyfrowane hasła są oczywiście dużo bezpieczniejsze. Niezaszyfrowane hasło można łatwo odczytać za pomocą programu przechwytyjącego pakiety. Szyfrowanie haseł jest zależne od systemu operacyjnego klienta łączącego się z serwerem Samby.

Jeśli Samba obsługuje zaszyfrowane hasła, wówczas przechowuje je w pliku o nazwie *smbpasswd*. Klient również przechowuje zaszyfrowane hasło użytkownika w swoim systemie. Oba systemy automatycznie szyfrują ustawione lub zmienione hasło za pomocą tego samego algorytmu.

Kiedy klient zażąda dostępu do serwera SMB obsługującego zaszyfrowane hasła (na przykład Samba lub Windows NT), dwa komputery prowadzą negocjacje następująco:

- Klient próbuje wynegocjować protokół z serwerem.
- Serwer określa protokół i zaznacza, że obsługuje zaszyfrowane hasła. W tym momencie przesyła losowo wygenerowany, 8-bitowy łańcuch wyzwania.
- Klient używa łańcucha wyzwania jako klucza do wtórnego zaszyfrowania swojego hasła, korzystając z algorytmu zdefiniowanego przez wynegocjowany protokół. Następnie wysyła wynik do serwera.
- Serwer robi to samo z zaszyfrowanym hasłem przechowywanym w swojej bazie danych. Jeśli wyniki są zgodne, oznacza to, że hasła są równoważne, więc użytkownik zostaje uwierzytelniony.

Przed rozpoczęciem korzystania z zaszyfrowanych haseł, należy utworzyć wpis dla każdego uniksowego użytkownika w pliku *smbpasswd*. Istnieje kilka sposobów na dodanie nowego wpisu do pliku *smbpasswd*. Zalecanym sposobem jest użycie programu *smbpasswd*:

```
smbpasswd -a <użytkownik>
```

Spowoduje to dodanie wpisu do pliku */etc/samba/smbpasswd* użytkownika, którego nazwa została podana jako parametr.

Programu *smbpasswd* można użyć też do zmiany zaszyfrowanego hasła w pliku */etc/samba/smbpasswd*. Program *smbpasswd* prosi o wprowadzenie starego hasła, a następnie o dwukrotne wpisanie nowego.

Kiedy w bazie danych znajdują się hasła użytkowników, będą oni mogli łączyć się z udziałami, używając zaszyfrowanych haseł.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
encrypt passwords	Wartość logiczna	Włącza obsługę zaszyfrowanych haseł. Popularnym sposobem jest dołączenie tej opcji za pomocą opcji include, dzięki której można utworzyć odrębne pliki konfiguracyjne, które mogą być wczytywane na podstawie np. systemu operacyjnego.	no	Globalny
unix password sync	Wartość logiczna	Określa, czy Samba ma uaktualniać standardową bazę haseł, kiedy użytkownik zmieni swoje zaszyfrowane hasło.	no	Globalny
passwd chat	Łańcuch (polecenia wymiany danych)	Ustawia sekwencję poleceń, które będą wysyłane do programu zmieniającego hasło. W wartości tej opcji mogą być używane m.in. zmienne: <ul style="list-style-type: none"> %O – stare hasło użytkownika %n – nowe hasło użytkownika. 	*old*password* %o\n *new*password* %n\n *new*password* %n\n *changed*	Globalny
passwd chat debug	Wartość logiczna	Wysyła komunikaty diagnostyczne dotyczące procesu zmiany hasła do plików dziennika z poziomem równym 100.	no	Globalny
passwd program	Łańcuch (polecenie Uniksa)	Określa program, który będzie uaktualniał standardowy plik haseł, kiedy Samba uaktualnia plik haseł zaszyfrowanych.	/bin/passwd %u	Globalny
password level	Wartość liczbową	Ustawia liczbę dużych liter permutowanych podczas dopasowywania hasła przesłanego przez klienta.	Brak	Globalny
update encrypted	Wartość logiczna	Określa, czy Samba ma uaktualniać plik zaszyfrowanych haseł, kiedy klient połączy się z udziałem za pomocą jawnego hasła.	no	Globalny
null passwords	Wartość logiczna	Określa, czy Samba ma zezwalać na dostęp użytkownikom mającym puste hasło.	no	Globalny
smb passwd file	Łańcuch (ścieżka dostępu)	Określa nazwę pliku z zaszyfrowanymi hasłami.	/etc/samba/smbpasswd	Globalny
hosts equiv	Łańcuch (ścieżka dostępu)	Określa nazwę pliku zawierającego nazwy hostów i użytkowników, którzy mogą się łączyć bez podawania hasła	Brak	Globalny
use rhosts	Łańcuch (ścieżka dostępu)	Określa nazwę pliku .rhosts, który pozwala użytkownikom na łączenie się bez podawania hasła.	Brak	Globalny

Tab. 45. Opcje konfiguracji haseł

Opcje logowania domenowego

W tradycyjnych grupach roboczych Windows 95/98 akceptuje wszystkie wpisywane podczas logowania nazwy użytkownika i hasła. W Windows 95/98 nie ma autoryzowanych użytkowników. Kiedy loguje się nowy użytkownik, system operacyjny prosi go o wprowadzenie nowego hasła i od tego momentu uwierzytelnia go za pomocą tego hasła. Windows 95/98 używa wprowadzonego hasła tylko wtedy, kiedy klient próbuje połączyć się z udziałem.

Logowania domenowe przypominają natomiast uniksowe mechanizmy bezpieczeństwa. Aby zalogować się w domenę, należy podać poprawną nazwę użytkownika i hasło, które są następnie weryfikowane w bazie haseł podstawowego kontrolera domeny. Jeśli hasło jest błędne, użytkownik jest o tym natychmiast informowany i nie może zalogować się w domenę. Po pomyślnym zalogowaniu w domenę, nie ma potrzeby ponownego uwierzytelniania się w udziale. Mówiąc ściślej, podstawowy kontroler domeny przyznaje klientowi żeton, który upoważnia do korzystania ze wszystkich udziałów bez ponownego kontaktowania się z kontrolerem.

Wszystkie klienty Windows NT łączące się z podstawowym kontrolerem domeny korzystają z kont zaufania (*trust accounts*). Konta te pozwalają komputerowi na logowanie się w samym kontrolerze domeny (a nie w jednym z jego udziałów), co oznacza, że podstawowy kontroler domeny może zaufać wszystkim przyszłym połączeniom nawiązywanym przez użytkowników tego klienta. Praktycznie rzecz biorąc, konto zaufania jest równoważne kontu użytkownika. W rzeczywistości używa się standardowych uniksowych kont użytkownika do emulowania kont zaufania w serwerze Samby. Nazwa używana przy logowaniu się na koncie zaufania to nazwa komputera z dołączonym znakiem dolara. Początkowo hasłem konta jest po prostu nazwa komputera pisana małymi literami. Aby zasymulować konto zaufania w serwerze Samby, należy utworzyć konto użytkownika odpowiadające nazwie komputera (ze znakiem dolara) oraz dodać zaszyfrowane hasło do pliku *smbpasswd*. W przypadku dodawania nowego konta użytkownika, nie trzeba tworzyć katalogu macierzystego ani wyznaczać powłoki użytkownika. Aby dodać zaszyfrowane hasło należy użyć polecenia *smbpasswd*:

```
smbpasswd -a -m <nazwa komputera>
```

Program *smbpasswd* automatycznie ustawi początkowe hasło na nazwę komputera pisaną małymi literami. Znak dolara dodawany jest automatycznie.

Aby Samba pełniła funkcję podstawowego kontrolera domeny dla klientów Windows, po stronie serwera muszą być spełnione następujące warunki:

- w sieci dostępny jest serwer WINS (serwer Samby albo Windows NT)
- Samba jest jedynym podstawowym kontrolerem domeny w bieżącej grupie roboczej
- Samba korzysta z zabezpieczeń na poziomie użytkownika (nie przekierowuje haseł na inny komputer).

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
domain logons	Wartość logiczna	Informuje, czy będzie używane logowanie domenowe Windows. Kiedy klient pomyślnie się zaloguje w domenę, Samba przesyła mu specjalny żeton, który pozwala na dostęp do udziałów bez ponownego uwierzytelniania.	no	Globalny
domain group map	Łańcuch (ścieżka dostępu)	Określa nazwę pliku z odwzorowaniami grup uniksowych na grupy domenowe Windows NT. Podane w określonym pliku uniksowe grupy, powinny znajdować się w pliku <i>/etc/group</i> . Opcja ta działa tylko z klientami Windows NT.	Brak	Globalny
domain user map	Łańcuch (ścieżka dostępu)	Określa nazwę pliku z odwzorowaniami użytkowników uniksowych na użytkowników domeny Windows NT. Podane w określonym pliku nazwy uniksowe użytkowników, powinny znajdować się w pliku <i>/etc/passwd</i> . Opcja ta działa tylko z klientami Windows NT.	Brak	Globalny
local group map	Łańcuch (ścieżka dostępu)	Określa nazwę pliku z odwzorowaniami grup uniksowych na grupy lokalne Windows NT. Podane w określonym pliku uniksowe grupy, powinny znajdować się w pliku <i>/etc/group</i> . Opcja ta działa tylko z klientami Windows NT.	Brak	Globalny
revalidate	Wartość logiczna	Określa, czy Samba ma wymuszać uwierzytelnianie użytkowników przy każdym łączeniu się z udziałem, niezależnie od poziomu bezpieczeństwa używanego przez serwer Samba.	no	Udział

Tab. 46. Opcje logowania domenowego

Opcje skryptów logowania

Samba umożliwia wykonywanie skryptów logowania Windows, czyli plików wsadowych (*.bat* lub *.cmd*) uruchamianych w kliencie podczas logowania się w domenę Windows. Skrypty te są przechowywane w serwerze uniksowym, a następnie przesyłane przez sieć do klienta i wykonywane wtedy, gdy użytkownik się zaloguje. Skrypty logowania są bardzo przydatnym narzędziem do konfigurowania parametrów sieciowych dla logujących się użytkowników.

W Windows 95/98 oraz NT, każdy użytkownik może mieć swój własny profil. W profilu gromadzone są informacje takie, jak: wygląd pulpitu użytkownika, aplikacje widoczne w menu *Start*, tło i inne. Jeśli profil jest przechowywany na lokalnym dysku, nazywany jest on profilem lokalnym. Jeśli profil jest przechowywany w serwerze, użytkownik może pobrać go do każdego komputera, który jest połączony z serwerem. Taki

profil nazywany jest przechodnim (*roaming profile*), ponieważ użytkownik może korzystać z tego samego profilu podczas pracy na różnych komputerach.

Jeśli Samba ma włączoną obsługę profili przechodnich, kiedy użytkownik zaloguje się po raz pierwszy, klient Windows utworzy plik *user.dat* lub *ntuser.dat* - w zależności od systemu operacyjnego klienta. Klient następnie zapisze w odpowiednim katalogu udziału zawartość pulpitu, menu *Start*, *Otoczenia sieciowego* oraz folderów programów. Przy ponownym logowaniu informacje te zostaną pobrane z serwera i wykorzystane do skonfigurowania komputera, z którego zalogował się użytkownik. Kiedy użytkownik wyloguje się, informacje zostaną ponownie zapisane na serwerze aż do czasu następnego logowania.

Użytkownicy mogą mieć także profile obowiązkowe, czyli profile przechodnie, których nie mogą zmienić. Profil obowiązkowy to plik *user.dat*, którego nazwa została zmieniona na *user.man*, i który jest przeznaczony tylko do odczytu. Zwykle zawiera on ustawienia, które administrator uznał za obowiązujące dla danych użytkowników.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
logon script	Łańcuch (ścieżka dosowa)	Określa położenie pliku wykonywalnego Windows, który zostanie uruchomiony w kliencie po zalogowaniu się użytkownika w domenę. Ponieważ skrypty te są pobierane przez klienta i wykonywane po stronie Windows, muszą zawierać dosowe sekwencje znaków powrót karetki – nowa linia.	Brak	Globalny
logon path	Łańcuch (nazwa UNC serwera i udziału)	Określa położenie profili przechodnich. Kiedy użytkownik się loguje, profil przechodni jest przekazywany z serwera do klienta i uaktualniany dla tego użytkownika. Kiedy użytkownik się wyloguje, zawartość profilu jest składowana z powrotem na serwerze.	\\%N%\%U\ profile	Globalny
logon drive	Łańcuch (litera dysku)	Opcja ta określa literę dysku w kliencie NT, na którą będzie mapowany katalog macierzysty podany w opcji logon home. Opcja ta działa tylko z klientami Windows NT.	Z:	Globalny
logon home	Łańcuch (nazwa UNC serwera i udziału)	Określa położenie katalogu macierzystego użytkownika na potrzeby dosowych poleceń NET. Opcja ta dobrze działa z sekcją [homes].	\\%N%\%U	Globalny

Tab. 47. Opcje skryptów logowania

Opcje konfiguracji odwzorowywania nazw

Samba dysponuje czterema różnymi mechanizmami odwzorowywania nazw:

- **Plik */etc/hosts***

Plik ten zawiera przekształcenia adresów IP na nazwy hostów.

- **Plik *LMHOSTS***

LMHOSTS to standardowy plik hostów LAN Managera, używany do odwzorowywania nazw na adresy IP w lokalnym systemie. Plik *LMHOSTS* ma format podobny do pliku */etc/hosts*, z tą różnicą, że nazwy po prawej stronie są nazwami NetBIOSowymi.

- **Rozgłaszanie**

Rozgłaszanie polega na tym, że jeśli potrzeby jest adres komputera, jego nazwa jest rozgłaszana w całej sieci i klient oczekuje na odpowiedź od tego komputera. Pakiety rozgłoszeniowe nie przechodzą jednak przez granice podsieci. Ponadto częste rozgłoszenia obniżają efektywność sieci.

- **WINS**

WINS (*Windows Internet Name Service*) to serwer NBNS dostępny z wielu podsieci. Dzięki temu można wyznaczyć jeden z komputerów na serwer WINS, a następnie podać jego adres każdemu z klientów, które korzystają z usług odwzorowywania nazw. W rezultacie żądania rejestracji i odwzorowywania nazwy mogą być kierowane do jednego komputera z dowolnego punktu sieci.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
wins support	Wartość logiczna	Określa, czy Samba ma świadczyć usługi WINS na rzecz komputerów w sieci.	no	Globalny
wins server	Łańcuch (adres IP lub nazwa DNS)	Określa adres innego dostępnego w sieci serwera WINS. Aby opcja ta zadziałała, opcja wins support musi być ustawiona na no .	Brak	Globalny
wins proxy	Wartość logiczna	Określa, czy Samba ma działać jako pośrednik innego serwera WINS, a więc przekazywać kierowane do siebie żądania rejestracji i odwzorowywania nazw do rzeczywistego serwera WINS.	no	Globalny
dns proxy	Wartość logiczna	Określa, czy Samba ma wyszukiwać nazwy komputerów za pomocą standardowych usług DNS serwera.	no	Globalny
name resolve order	lmhosts, hosts, wins lub bcast	Określa kolejność metod używanych przez Sambę podczas odwzorowywania nazw.	lmhosts hosts wins bcast	Globalny

Tab. 48. Opcje konfiguracji odwzorowywania nazw

Opcje obsługi języków narodowych

Samba posiada ograniczoną zdolność posługiwania się narodowymi językami. Jeśli istnieje potrzeba posługiwania się znakami spoza standardowego zbioru ASCII, pomagają w tym opcje obsługi języków narodowych.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
client code page	Wartość liczbowa	Ustawia stronę kodową, którą posługują się klienci. Strony kodowe są używane przez klienty DOS-a i Windows w celu ustalenia reguł odwzorowywania liter małych na duże.	850	Globalny
character set	Łańcuch	Tłumaczy strony kodowe na alternatywne zestawy znaków uniksowych.	Brak	Globalny
valid chars	Łańcuch (zbiór znaków)	Dodaje pojedyncze znaki do strony kodowej. Należy jej używać już po ustawieniu strony kodowej klienta. Opcja ta jest jednak wypierana przez nowocześniejsze systemy kodowania.	Brak	Globalny

Tab. 49. Opcje obsługi języków narodowych

Opcje różne

Wiele opcji Samby jest związanych ze specyfikacją systemu operacyjnego – systemu uniksowego albo Windows. Zwykle nie powinno się zmieniać ich domyślnych wartości.

Opcja	Parametry	Funkcja	Wartość domyślna	Zasięg
time server	Wartość logiczna	Określa, czy serwer Samby ma ogłaszać się jako serwer czasu SMB.	no	Globalny
deadtime	Wartość liczbowa (ilość minut)	Określa w minutach czas bezczynności, po którym połączenie powinno zostać zakończone.	0	Globalny
max disk size	Wartość liczbowa (w MB)	Ustawia największy rozmiar dysku zwracany klientom. Nie ma wpływu na rzeczywiste operacje dyskowe.	0	Globalny
fstype	NTFS, FAT lub Samba	Ustawia typ systemu plików, który serwer zgłasza klientom.	NTFS	Globalny
panic action	Łańcuch (polecenie)	Określa program, jaki powinien zostać uruchomiony w razie błędu serwera Samby.	Brak	Globalny

Tab. 50. Opcje różne

Plik /etc/samba/smb.conf

```
# Przykładowy plik konfiguracyjny /etc/samba/smb.conf

#Sekcja globalna
[global]
    client code page = 852
    workgroup = TK
    netbios name = FRAXINUS
    server string = Serwer Samby
    interfaces = 192.168.1.0/255.255.255.0
    encrypt passwords = True
    bind interfaces only = Yes
    null passwords = No
    passwd program = /usr/bin/passwd %u
    log level = 4
    log file = /var/log/samba/%I.log
    max log size = 2500
    name resolve order = wins bcast hosts lmhosts
    time server = Yes
    deadtime = 15
    keepalive = 15
    max open files = 500
    character set = ISO8859-2
    mangled stack = 100
    logon script = %u.bat
    logon path = /samba/netlogon
    domain logons = Yes
    os level = 100
    preferred master = True
    local master = True
    domain master = True
    wins support = Yes
    comment = Łączenie z serwerem Samby
    admin users = admin
    create mask = 0700
    hosts allow = localhost, 192.168.1.0/255.255.255.0
    nt acl support = No
    write cache size = 65536
    dont descend = /dev,/proc,/boot,/root,/var,/bin,/etc,/sbin,/usr
    dos filetimes = Yes

#Sekcje udziałów
[homes]
    comment = Katalog indywidualny
    path = /home/%u
    write list = @school
    directory mask = 0700
    browseable = No

[netlogon]
    comment = Usługi logowania sieciowego
    path = /samba/netlogon
    browseable = No
    write list = @experts
    create mask = 0755
    directory mask = 0770
    guest ok = Yes
    locking = No
    share modes = No
```

```
[public]
comment = Katalog dla sieci lokalnej
path = /samba/pub
write list = @experts
create mask = 0770
directory mask = 0770
volume = public

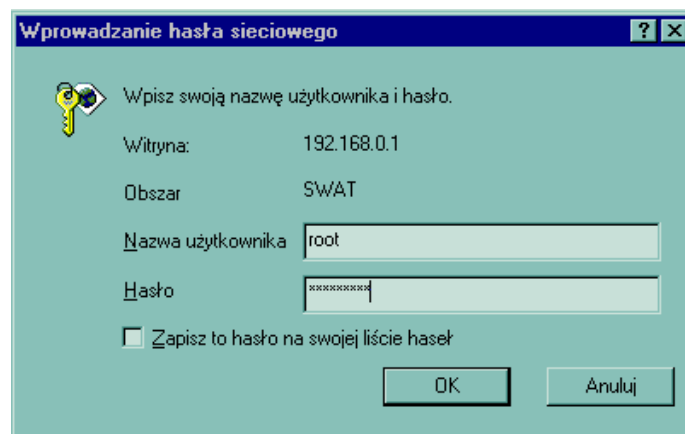
[goscie]
comment = Katalog dla wszystkich
path = /samba/goscie
write list = @experts
create mask = 0770
directory mask = 0770
guest ok = Yes
volume = goscie

[install]
comment = Katalog instalacyjny
path = /samba/install
write list = @experts
create mask = 0770
directory mask = 0770
guest ok = Yes
volume = install
browseable = No
```

Konfiguracja za pomocą programu swat

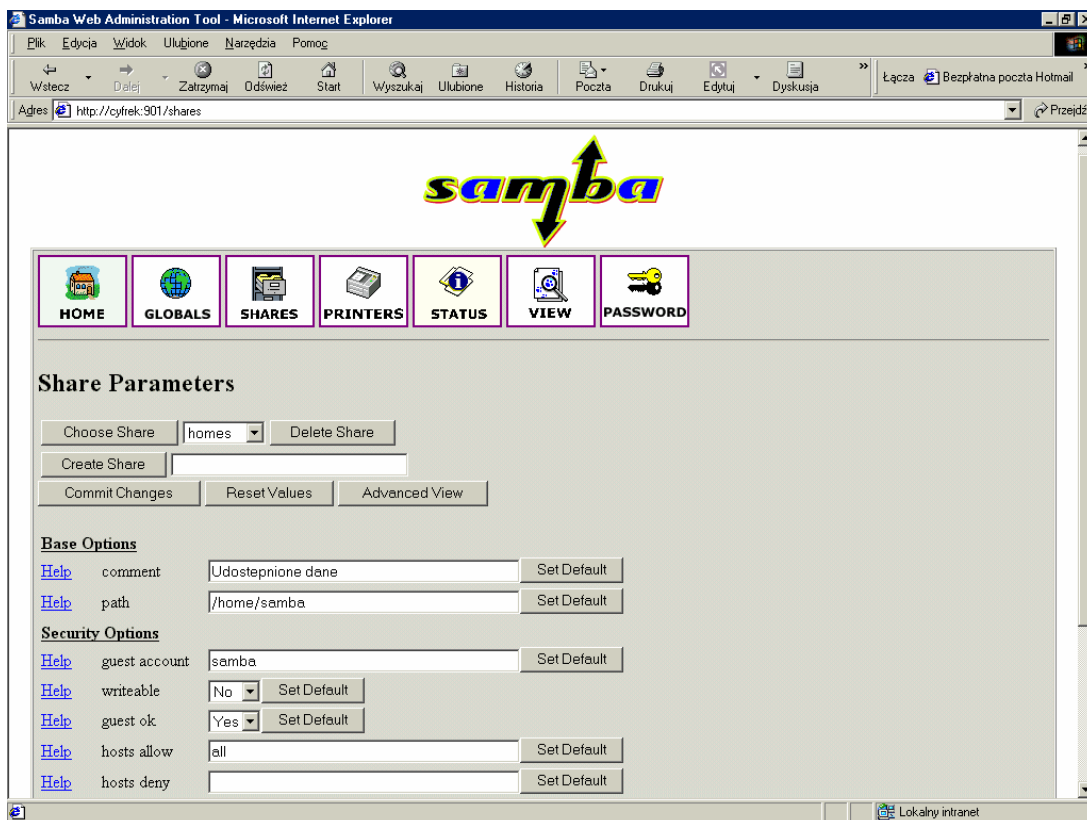
W przypadku posiadania zainstalowanego serwera www, np. *apache*, istnieje możliwość konfiguracji serwera Samby za pomocą przeglądarki internetowej przy wykorzystaniu programu *swat*.

Aby uzyskać połączenie z programem *swat*, należy w oknie przeglądarki wpisać adres IP serwera wraz z numerem portu, na jakim nasłuchuje program *swat* (domyślnie jest to port 901). Po połączeniu z udziałem, należy podać nazwę użytkownika oraz uniksowe hasło.



Rys. 15. Proces wprowadzania hasła sieciowego w celu uzyskania dostępu do programu *swat*

Po pomyślnym zweryfikowaniu nazwy użytkownika i hasła przez serwer, w oknie przeglądarki pojawi się strona umożliwiająca zmianę parametrów Samby.



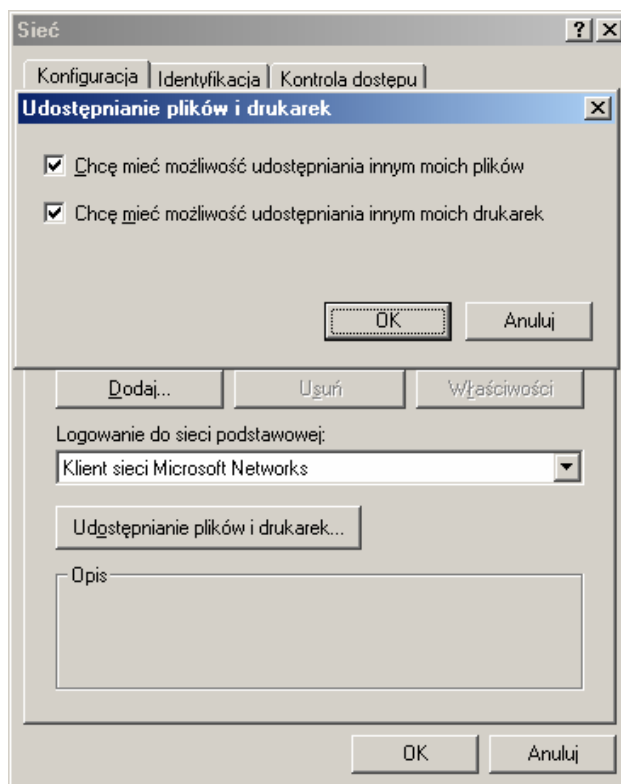
Rys. 16. Wygląd okna przeglądarki po uzyskaniu dostępu do programu swat

W menu dostępne są następujące sekcje:

- **HOME**
Umożliwia obejrzenie podstawowych informacji o Samba i programie *swat*.
- **GLOBALS**
Umożliwia zmianę opcji globalnych.
- **SHARES**
Umożliwia zmianę opcji udostępnionych zasobów.
- **PRINTERS**
Umożliwia zmianę opcji drukarek.
- **STATUS**
Umożliwia obejrzenie informacji o stanie demonów Samby.
- **VIEW**
Umożliwia wgląd w tekstowy plik konfiguracyjny Samby */etc/samba/smb.conf*.
- **PASSWORD**
Umożliwia dodawanie zmian haseł istniejącym użytkownikom.

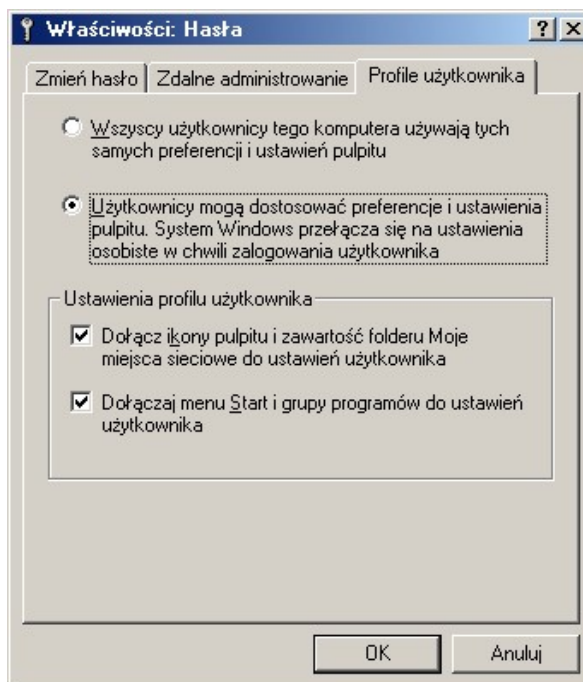
Konfiguracja klientów Windows

Pierwszą czynnością, jaką należy zrobić to umożliwienie klientom Windows korzystania z protokołu SMB. W tym celu należy otworzyć okno dialogowe Sieć w Panelu Sterowania. Następnie należy kliknąć na ramkę Udostępnianie plików i drukarek i zaznaczyć obydwa lub jedno pole wyboru.



Rys. 17. Okno dialogowe Sieć

Klienty Windows nie zostały zaprojektowane jako platformy dla wielu użytkowników, w przeciwieństwie do systemów uniksowych. Systemy te mogą jednak do pewnego stopnia obsługiwać wielu użytkowników – system operacyjny Windows może przechowywać oddzielny profil oraz plik haseł dla każdego użytkownika.

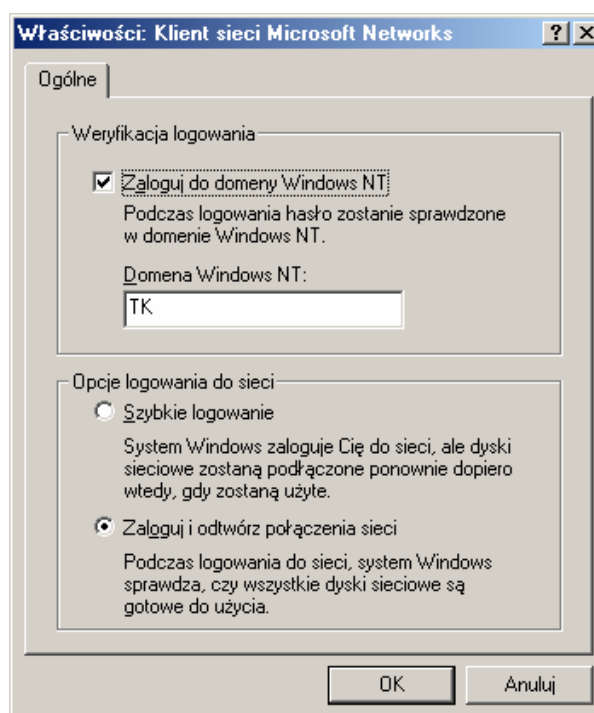


Rys. 18. Okno dialogowe Właściwości: Hasła

Należy zatem poinformować Windows, że powinien przechowywać oddzielne profile użytkowników i gromadzić nazwy użytkowników i hasła w celu uwierzytelniania osób pracujących skorzystać z zasobów Samby. Służy do tego aplet Hasła w Panelu Sterowania. W oknie Właściwości: Hasła należy kliknąć kartę Profile użytkownika. Następnie należy zaznaczyć pole zaczynające się od słów Użytkownicy mogą dostosować preferencje. Spowoduje to, że system Windows będzie przechowywał oddzielny profil dla każdego użytkownika. Należy także zaznaczyć obydwa pola wyboru w ramce Ustawienia profilu użytkownika.

Jeśli klientem jest komputer z systemem operacyjnym Windows95, należy zainstalować aplikację (*vrdrupd.exe*) w celu uaktywnienia obsługi haseł zaszyfrowanych. System Windows98 i nowsze domyślnie obsługują hasła zaszyfrowane.

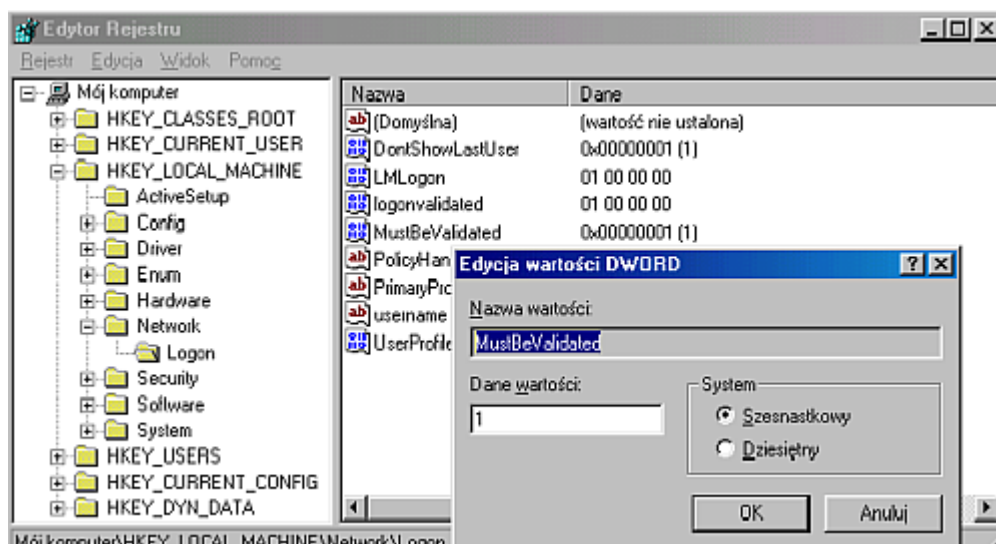
Kolejną czynnością, jaką należy wykonać, jest umożliwienie klientom Windows logowania domenowego. W tym celu należy otworzyć okno dialogowe Sieć z Panelu Sterowania, zaznaczyć pozycję Klient sieci Microsoft Networks i kliknąć przycisk Właściwości. Ukaze się okno, w którym należy zaznaczyć pole wyboru Zaloguj do domeny Windows NT na górze okna dialogowego i wpisać nazwę grupy roboczej.



Rys. 19. Okno dialogowe Właściwości: Klient sieci Microsoft Networks

Aby zabronić ominięcia okna logowania przez wciskanie przycisku Anuluj lub klawisza [Esc], należy w rejestrze systemu (*regedit.exe*), w kluczu HKEY_LOCAL_MACHINE\Network\Logon, utworzyć wartość DWORD o nazwie MustBeValidated i przypisać jej wartość 1.

Aby wyłączyć w oknie logowania pojawiania się nazwy ostatniego zalogowanego użytkownika, należy w kluczu HKEY_LOCAL_MACHINE\Network\Logon, utworzyć wartość DWORD o nazwie DontShowLastUser i przypisać jej wartość 1.



Rys. 20. Okno Edytora Rejestru

Po wykonaniu tych czynności będzie możliwe korzystanie z udziałów komputerów w sieci lokalnej oraz pełne wykorzystanie opcji Samby, jak na przykład logowanie domenowe.

Skrypty logowania

Skrypty logowania są przechowywane na serwerze Samby i pobierane do klienta przy każdym logowaniu. Skrypty logowania korzystają zatem z aplikacji konfiguracji sieci używane przez klientów Windows (program *net.exe*), a nie z programów uniksowych.

Blokada	Opis
CONFIG	Wyświetla ustawienia bieżącej grupy roboczej
DIAG	Uruchamia program diagnostyczny w celu wyświetlenia informacji o sieci
HELP	Wyświetla informacje o poleceniach i komunikatach błędów
INIT	Ładuje protokoły i sterowniki kart sieciowych
LOGOFF	Przerywa połączenie z podłączonymi do komputera zasobami
LOGON	Identyfikuje jako członka grupy roboczej
PASSWORD	Zmienia hasło logowania
PRINT	Wyświetla informacje o kolejkach wydruku i steruje zadaniami wydruku
START	Uruchamia usługi
STOP	Zatrzymuje usługi
TIME	Wyświetla czas lub synchronizuje zegar komputera z zegarem serwera
USE	Podłącza lub odłącza od udostępnionego zasobu lub wyświetla informacje o połączeniach
VER	Wyświetla typ i wersję używanego readresatora grupy roboczej
VIEW	Wyświetla listę komputerów udostępniających zasoby lub listę udostępnianych zasobów na określonym komputerze

Tab. 51. Opcje polecenia NET

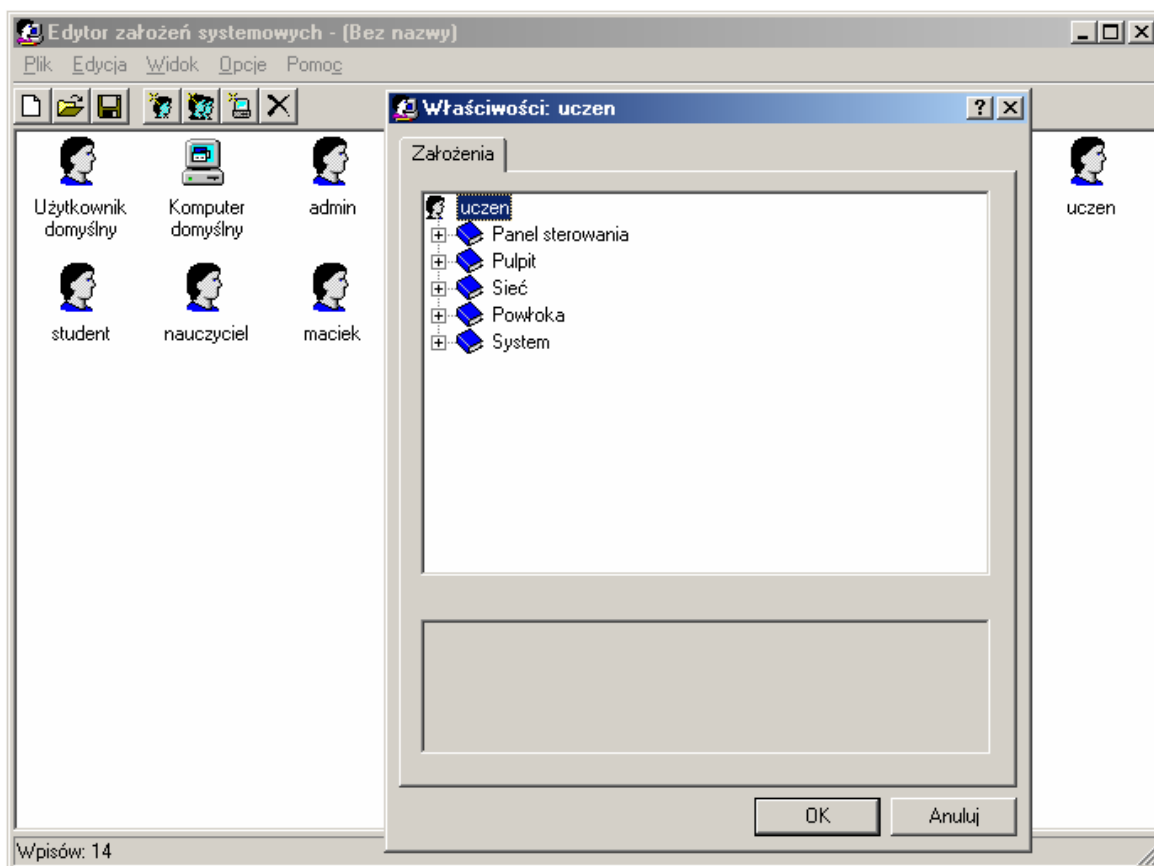

```
@rem Przykładowy skrypt logowania *.bat dla klientów

net time \\fraxinus /set /yes
@rem Synchronizacja czasu z serwerem

net use h: /home /yes
net use i: \\fraxinus\public /yes
@rem Mapowanie dysków sieciowych
```

Profile przechodnie

Narzędziem przeznaczonym do nadawania uprawnień użytkownikom jest Edytor założeń systemowych (*poledit.exe*). Zapewnia on wygodny dostęp do wszystkich wymienionych opcji przy pomocy graficznego interfejsu.



Rys. 21. Widok okna edytora założeń systemowych

Najpierw należy w programie *poledit.exe* otworzyć rejestr, kliknąć na ikonę Komputer lokalny, w oknie właściwości przejść do gałęzi Sieć, potem do podgałęzi Aktualizacja i z listy Typ aktualizacji wybrać pozycję Ręczny (używa podanej ścieżki). W polu Ścieżka do ręcznej aktualizacji należy wpisać ścieżkę dostępu, w której będzie się znajdował plik z założeniami systemowymi (w celu zwiększenia bezpieczeństwa plik może znajdować się na serwerze).

Następnie należy stworzyć nowy plik założeń systemowych, dodać wszystkich korzystających z systemu użytkowników, przypisać odpowiednie uprawnienia i restrikcje, a następnie zapisać w miejscu, które wcześniej zostało podane jako Ścieżka do ręcznej aktualizacji.

6. BEZPIECZEŃSTWO SYSTEMU

ZAGADNIENIA BEZPIECZEŃSTWA

Zawsze istnieje niebezpieczeństwo, że ktoś nieupoważniony do tego, przejmie informacje przesyłane przez sieć. Zabezpieczenie systemu musi prowadzić do zmniejszenia ryzyka nieupoważnionego zmieniania danych, czy to przesyłanych przez sieć, czy to składowanych w systemie.

Rodzaje ataków

Zasadniczo, ataki można podzielić na lokalne i zdalne. Choć ataki lokalne, mogą być bardzo niebezpieczne, to jednak nie są one tak częste, jak ataki zdalne.

Ataki zdalne można podzielić następująco:

- ataki na „czynniki ludzkie”
- ataki *Denial-of-Service*
- sniffing
- łamanie haseł
- spoofing
- ataki typu *Man-in-the-Middle*
- konie trojańskie, wirusy, robaki
- exploity.

Ataki na „czynniki ludzkie”

Ataki na „czynniki ludzkie” to prawdopodobnie najstarszy sposób uzyskiwania nieuprawnionego dostępu do obszarów zabezpieczonych. Bazują one na fakcie, że zawsze najsłabszym ogniwem łańcucha jest człowiek.

Jedynym sposobem obrony przed takimi atakami jest posiadanie jasno zdefiniowanych założeń bezpieczeństwa i wytycznych oraz szkolenie użytkowników i operatorów.

Ataki *Denial-of-Service*

Celem ataku *Denial-of-Service* (odmowy wykonania usługi, w skrócie *DOS*) jest uniemożliwienie korzystania z systemu uprawnionym użytkownikom. Przykładowo, można otworzyć jednocześnie bardzo wiele połączeń z serwerem i utrzymywać je otwarte, sprawiając, że serwer nie będzie w stanie obsłużyć innych użytkowników.

Ponieważ ataki *DOS* można przeprowadzać w bardzo różny sposób, praktycznie nie ma możliwości zabezpieczenia się przed ich wszystkimi rodzajami. Niektóre spośród tych ataków wykorzystują wady samego protokołu TCP/IP.

Sniffing

Sniffer to typ programu, który podsłuchuje wszystkie pakiety przesyłane przez sieć i wyszukuje wśród nich identyfikatory użytkowników oraz hasła. Pakiety podróżujące przez Internet przechodzą po drodze przez wiele komputerów i sieci, dlatego nie powinno się logować się na zdalnym serwerze, korzystając z nieszyfrowanego połączenia. Jedynym sposobem zabezpieczenia się przed programami typu sniffer jest korzystanie z programów szyfrujących połączenia, takich jak *ssh* (zabezpieczona powłoka).

Łamanie haseł

Wielu użytkowników wpisuje jako hasła słowa łatwe do odgadnięcia. Użytkownicy często wybierają na hasło swoje imię bądź nazwisko, datę urodzin, numer dowodu osobistego, czy też nazwę ulubionego filmu. Wszystkie tego typu hasła stanowią dobry cel ataków.

Istnieje wiele sposobów zapobiegania atakowi tego rodzaju. Pierwszym z nich jest zastosowanie programu narzędziowego, który będzie sprawdzał, czy wprowadzone przez użytkownika hasło znajduje się w słowniku. Użytkownicy powinni również zostać zmuszeni do regularnego zmieniania haseł. Inną ważną czynnością jest uaktywnienie obsługi haseł cieniowanych (*shadow password*).

Spoofing

W trakcie ataku tego typu fabrykowane są pakiety z fałszywym adresem źródłowym, co sprawia, że docelowy komputer wierzy, iż pakiety te pochodzą ze znanego mu, bezpiecznego komputera. Nigdy zatem nie można zakładać, że adresy IP połączeń przychodzących z Internetu są prawdziwe.

Ataki tego typu są zazwyczaj przeprowadzane „na ślepo”, bez możliwości sprawdzenia odpowiedzi atakowanego serwera, chyba że komputer, z którego zostaje przeprowadzony atak, znajduje się w tej samej podsieci co komputer docelowy lub też podsieci znajdującej się pomiędzy komputerem atakowanym a komputerem. Osoba atakująca może również łączyć *spoofing* ze zmianą trasy przesyłania pakietów (*source-routing*). W tym wypadku wysyłany pakiet zawiera w nagłówku listę adresów komputerów, przez które musi przejść by powrócić do komputera źródłowego. Gdy taki pakiet dotrze do niezabezpieczonego komputera, komputer wyśle odpowiedź nie trasą standardową, lecz trasą zgodną z opisem w pakiecie, dzięki czemu napastnik będzie mógł zaobserwować efekt swojego ataku.

Innym sposobem wykorzystania techniki *spoofing* jest zacieranie za sobą śladów. Przykładowo, napastnik może przeskanować daną sieć, sprawdzając, czy są w niej jakieś słabe punkty, które mógłby wykorzystać, a podczas skanowania wysyłać, oprócz właściwych pakietów skanujących, wiele pakietów o sfałszowanych adresach źródłowych.

Ataki typu Man-in-the-Middle

Atak typu *Man-in-the-Middle* (człowiek pośrodku) występuje, gdy osoba atakująca przechwytuje komunikację w obie strony i modyfikuje ją. Dzięki temu klient sądzi, że porozumiewa się z serwerem, a serwer, że nawiązuje połączenie z klientem – w rzeczywistości jednak zarówno serwer, jak i klient przesyłają informacje do napastnika.

Przed takimi atakami można zabezpieczyć się za pomocą połączenia szyfrowanego po obu stronach za pomocą pary kluczy prywatny-publiczny. Ponieważ dane szyfrowane za pomocą klucza publicznego, mogą zostać odszyfrowane wyłącznie za pomocą klucza prywatnego i odwrotnie, taka para kluczy może zostać wykorzystana do „podpisywania” informacji. Każda strona oblicza sumę kontrolną danych i szyfruje je kluczem prywatnym, dzięki czemu druga strona może odszyfrować dane odpowiednim kluczem publicznym i sprawdzić, czy suma kontrolna jest zgodna z oczekiwaniami.

Konie trojańskie, wirusy i robaki

Koń trojański to program, który rzekomo wykonuje czynności pożądane przez użytkownika, ale po uruchomieniu w rzeczywistości robi coś innego. Dodatkowo, jeśli program zostanie uruchomiony z uprawnieniami administratora, zainstaluje furtkę, która umożliwi napastnikowi wejście do zaatakowanego systemu w każdej chwili.

W odróżnieniu od koni trojańskich, wirusy to programy dołączające się do innych programów i uruchamiane przy uruchamianiu zarażonego oprogramowania. Jeśli jedyną funkcją wirusa jest rozmnażanie się, jego działanie może nie być bardzo uciążliwe – równie dobrze jednak wirus może mieć zaprogramowaną sekwencję niszczącą, uaktywnianą po spełnieniu pewnych warunków (na przykład w chwili wystąpienia określonej daty, wirus sformatuje dysk twardy).

Ostatnim typem programów, które mogą być potencjalnie niebezpieczne są tzw. robaki (*worms*). Schemat działania robaków polega na wysyłaniu kopii samych siebie do innych komputerów w sieci. Programy te mogą na przykład wykorzystywać luki w bezpieczeństwie programów pocztowych do wysyłania swoich kopii do innych serwerów lub też odczytywać książkę adresową użytkownika i wysyłać swoje kopie na wszystkie znalezione w niej adresy. Robaki, w odróżnieniu od wirusów, nie dołączają się do innych programów.

Exploity

Wiele osób atakujących systemy komputerowe rozpowszechnia swoją wiedzę w formie gotowych programów nazywanych *exploitami*. Programy te wykorzystują wykryte słabe punkty oprogramowania. Załóżmy przykładowo, że w jakiejś wersji serwera IMAP, który wykonuje część kodu z uprawnieniami administratora, wykryto błąd występujący po wydaniu serwerowi określonej sekwencji nieprawidłowych poleceń. Napastnik taki może napisać program, który wykonuje dokładnie tę sekwencję i opublikować go na łamach popularnych witryn internetowych. Inne osoby mogą pobrać gotowy *exploit* i wykorzystać go, nie wiedząc nawet, w jaki sposób działa.

Polityka bezpieczeństwa

Jednym z problemów wynikających z korzystania z potężnych, zorientowanych sieciowo systemów operacyjnych, takich jak Linux, czy też każdy inny system uniksowy, jest bezpieczeństwo. Możliwość dokonywania zdalnie, jako legalny użytkownik systemu, niemal dowolnego zadania oznacza zarazem, że, potencjalnie, może je wykonać również ktoś inny, podający się za legalnego użytkownika. Rozpowszechnienie się Internetu spowodowało, że dostanie się do komputerów stało się znacznie łatwiejsze i to bez konieczności posiadania fizycznego dostępu do atakowanych maszyn.

Komputer musi w jakiś sposób bezbłędnie odróżniać legalnych użytkowników od intruzów i pozwalać użytkownikom uprawnionym na korzystanie z usług, do których powinni mieć dostęp, równocześnie odmawiając tego prawa intruzom. Komputer musi również radzić sobie z atakami nastawionymi nie na uzyskanie dostępu do usług, ale na zablokowanie możliwości korzystania z nich legalnym użytkownikom (poprzez przeciążanie systemu).

Bez wątpienia zabezpieczanie systemu może być trudnym zadaniem. Nie ma zabezpieczeń, które chroniłyby przed wszystkimi atakami. Tym niemniej, można bardzo utrudnić życie osobom próbującym włamać się do systemu – na tyle, by zniechęcić większość z nich i zmusić osoby, którym, być może, mogłoby się powieść, do poświęcenia temu jak największej ilości czasu i wysiłku.

Zasadą ogólną tworzenia zabezpieczeń jest to, by nie utrudniały w dużym stopniu zadań, do których przeznaczony jest komputer i zmuszały osoby atakujące do spędzenia jak największej ilości czasu na próbach przedostania się przez mechanizmy zabezpieczające.

Najlepszym sposobem uniknięcia nieupoważnionego dostępu jest podjęcie działań prewencyjnych związanych z pozbawieniem możliwości uzyskania dostępu przez sieć do maszyny.

LINUX-PAM

Linux-PAM (*Pluggable Authentication Modules for Linux*) jest zestawem bibliotek, które pozwalają administratorowi systemu wybrać sposób uwierzytelniania użytkowników przez poszczególne aplikacje. By system PAM działał, poszczególne aplikacje muszą być skompilowane z obsługą PAM.

Linux-PAM daje stabilny i ogólny interfejs (API), któremu podlegają w zadaniach autentykacji programy dające przywileje.

Podstawową właściwością podejścia PAM jest to, że natura autentykacji jest dynamicznie konfigurowalna. Innymi słowy, administrator systemu ma znaczne możliwości w wyborze sposobu autentykacji poszczególnych aplikacji. Ta dynamiczna konfiguracja jest ustawiana zawartością pojedynczego pliku konfiguracyjnego Linux-PAM, czyli */etc/pam.conf*.

Alternatywnie, można wszystko konfigurować pojedynczymi plikami konfiguracyjnymi, zlokalizowanymi w katalogu */etc/pam.d/*. Obecność tego katalogu spowoduje, że Linux-PAM zignoruje */etc/pam.conf*.

Pliki konfiguracyjne Linux-PAM definiują połączenia pomiędzy usługami a wstawialnymi modułami autentykacji (PAM'ami), które dokonują rzeczywistych zadań autentykacji.

Biblioteki Linux-PAM rozdzielają zadania autentykacji na cztery niezależne grupy zarządzania:

- **account**

Daje usłudze możliwość weryfikacji konta użytkownika, m.in. ważność hasła, dostęp użytkownika do żądanych usług.

- **authentication**

Ustala, czy użytkownik jest tym, za którego się podaje. Zazwyczaj robi się to poprzez zapytanie użytkownika o pewną odpowiedź, której musi udzielić. Nie wszystkie autentykacje są tego rodzaju, istnieją też sprzętowe schematy autentykacji, które mają odpowiednie moduły, nadające się do bezproblemowego podstawienia za standardowe modele autentykacji.

- **password**

Zadaniem tej grupy jest odświeżanie mechanizmów autentykacji. Zazwyczaj usługi takie są ściśle związane z tymi z *auth*. Niektóre mechanizmy autentykacji dobrze nadają się do odświeżania tą funkcją.

- **session**

Zadania tej grupy obejmują rzeczy, które powinny być dokonane przed daniem usługi oraz po jej wycofaniu. Zadania takie to m.in. obsługa śladów rewizyjnych i montowanie katalogu domowego użytkownika. Grupa obsługi sesji jest ważna, gdyż udostępnia zarówno hak otwierający, jak i zamykający modułów.

Konfiguracja PAMów

Gdy uruchamiana jest aplikacja świadoma przyznawania uprawnień poprzez bibliotekę Linux-PAM, aktywuje ona swoje powiązanie z PAM-API. Aktywacja ta określa wiele rzeczy, wśród których najważniejszą jest przeczytanie plików konfiguracyjnych: */etc/pam.conf*, lub w wypadku istnienia odpowiedniego katalogu, pliki z */etc/pam.d/*.

Plik */etc/pam.conf*

Pliki te wymieniają PAMy, które będą się zajmowały zadaniami autentykacji danej usługi i odpowiednie zachowanie PAM-API, gdy któryś z PAMów zawiedzie.

Składnia pliku konfiguracyjnego */etc/pam.conf* jest następująca. Plik jest złożony z listy reguł, przy czym każda z nich zwykle jest umieszczana w pojedynczej linii, choć może być też złożona na końcu linii. Format każdej reguły to rozdzielona spacjami kolekcja elementów, z których pierwsze trzy nie rozróżniają wielkości liter:

<usługa> <rodzaj> <kontrola> <ścieżka modułu> <argumenty modułu>

gdzie **usługa** jest nazwą odpowiadającej aplikacji (usługa **other** jest zarezerwowana na tworzenie reguł domyślnych), **rodzaj** określa grupę zarządzania, której odpowiada reguła (poprawne wpisy to: **account**, **auth**, **password** i **session**), pole **kontrola** określa zachowanie PAM-API po niepowodzeniu modułu w procesie autentykacji, **ścieżka modułu** jest pełną nazwą pliku PAMu używanego przez aplikację, zaś **argumenty modułu** to lista rozdzielonych spacjami elementów, używanych do modyfikowania określonego zachowania danego PAMu.

Flaga kontroli	Działanie
requisite	W przypadku niepowodzenia takiego PAM, nastąpi natychmiastowe zatrzymanie procesu autentykacji
required	W przypadku niepowodzenia takiego PAM, zostanie zwrócony błąd przez PAM-API, lecz dopiero po tym, jak pozostałe moduły dla tej usługi i rodzaju PAM zostaną wykonane
sufficient	Sukces takiego modułu wystarcza do zadowolenia wymagań autentykacji w stosie modułów (jeżeli wcześniej nie powiódł się moduł required , sukces tego jest ignorowany)
optional	Sukces lub niepowodzenie tego modułu jest istotny tylko jeśli jest jedynym modułem na stosie związanym z daną usługą i rodzajem PAM

Tab. 52. Flagi kontrolne w pliku */etc/pam.conf*

Składnia plików z */etc/pam.d/* jest taka sama, z tą różnicą że nie ma tam pól usług. W tym wypadku, usługa jest nazwą pliku z */etc/pam.d/*. Nazwa pliku musi być zapisana małymi literami.

Ważną właściwością Linux-PAM jest to, że można zestawić na stosie wiele reguł i łączyć tak usługi wielu PAMów dla danego zadania autentykacji.

Plik /etc/pam.d/login

```
# Przykładowy plik /etc/pam.d/login

auth            requisite    pam_securetty.so
#Wyłącza możliwość zalogowania jako root z wyjątkiem terminali
#określonych w pliku /etc/securetty

auth            requisite    pam_nologin.so
#Wyłącza możliwość zalogowania innego niż root w sytuacji, gdy
#istnieje plik /etc/nologin.

auth            required     pam_env.so
#Zamienia plik /etc/environment i pozwala także na użycie
#rozszerzonej konfiguracji w pliku /etc/security/pam_env.conf.

auth            required     pam_unix.so          nullok
#Standardowa Uniksowa autentykacja. Opcja "nullok" pozwala na
#rozliczanie haseł.

#auth           optional     pam_group.so
#Pozwala pewnym dodatkowym grupom przyznawać użytkownikowi podstawowe
#z rzeczy jak czas, terminal, usługa czy użytkownik

#account        requisite    pam_time.so
#Pozwala na ustawienie czasu wstrzymywania logowania za pomocą pliku
#/etc/security/time.conf.

#account        required     pam_access.so
#Pozwala na ustawianie limitów dostępu za pomocą pliku
#/etc/security/access.conf.

account         required     pam_unix.so
session         required     pam_unix.so
#Standardowe Uniksowe rozliczanie i zarządzanie sesją.

#session        required     pam_limits.so
#Ustawia limity użytkowników za pomocą pliku
#/etc/security/limits.conf.

session         optional     pam_lastlog.so
#Wyświetla ostatnią informację o pomyślnym logowaniu do systemu.

session         optional     pam_motd.so
#Wyświetla wiadomość dnia (plik /etc/motd)po pomyślnym zalogowaniu do
#systemu.

session         optional     pam_mail.so          standard    noenv
#Wyświetla status skrzynki pocztowej po pomyślnym zalogowaniu. Można
#też stąd uaktywnić zmienną środowiskową MAIL, ale lepiej jest radzić
#sobie przez plik /etc/login.defs, odkąd polecenie userdel jest także
#używane do upewniania się, że usunięcie użytkownika, także usuwa
#jego skrzynkę pocztową.

password        required     pam_unix.so nullok obscure min=4 max=8 md5
#Jest używane tylko, kiedy wygasa ważność hasła i musi być zmienione.
#Opcja "nullok" umożliwia użytkownikom zmianę pustego hasła, bo
#inaczej puste hasła są traktowane jako konta zablokowane. Wpis "md5"
#włącza obsługę haseł MD5. Opcje "min" i "max" egzekwują długość
#nowego hasła.
```


Plik /etc/pam.d/passwd

```
# Przykładowy plik /etc/pam.d/passwd

password      required      pam_unix.so nullok obscure min=4 max=8 md5
#Jest używane tylko, kiedy wygasa ważność hasła i musi być zmienione.
#Opcja "nullok" umożliwia użytkownikom zmianę pustego hasła, bo
#inaczej puste hasła są traktowane jako konta zablokowane. Wpis "md5"
#włącza obsługę haseł MD5. Opcje "min" i "max" egzekwują długość
#nowego hasła.
```

Plik /etc/pam.d/su

```
# Przykładowy plik /etc/pam.d/su

auth          required      pam_wheel.so
#Umożliwia korzystanie z polecenia su tylko użytkownikom z grupy
#root (domyślnie) lub po jej utworzeniu - grupy wheel.

#auth         sufficient    pam_wheel.so      trust
#Umożliwia korzystanie z polecenia su bez podawania hasła roota
#członkom grupy wheel.

#auth         required      pam_wheel.so      deny group=<grupa>
#Uniemożliwia członkom grupy "grupa" korzystania z polecenia su

auth          sufficient    pam_rootok.so
#Umożliwia użytkownikowi root korzystania z su bez podawania hasła
#(standardowa operacja).

#account      requisite     pam_time.so
#Umożliwia ustawianie czasu wyłączenia polecenia su za pomocą pliku
#/etc/security/time.conf.

auth          required      pam_unix.so
account       required      pam_unix.so
session       required      pam_unix.so
#Standardowe Uniksowa autentykacja.

session       required      pam_limits.so
#Ustawia limity użytkowników za pomocą pliku
#/etc/security/limits.conf.
```

Konfiguracja ograniczeń

Konfigurację podstawowych ograniczeń oraz przywilejów umożliwiają pliki konfiguracyjne z katalogu */etc/security*. Zawiera on następujące pliki:

- */etc/security/access.conf*
- */etc/security/group.conf*
- */etc/security/limits.conf*
- */etc/security/pam_env.conf*
- */etc/security/time.conf*

Plik /etc/security/access.conf

Plik */etc/security/access.conf* pozwala na kontrolę tego, skąd następuje logowanie poszczególnych użytkowników.

Każdy wpis w tym pliku ma następujący format:

```
<zezwozenie>:<uzytkownicy>:<pochodzenie>
```

gdzie **zezwozenie** może przyjmować dwie wartości: **+** (przyznanie dostępu) lub **-** (zabronienie dostępu), pole **uzytkownicy** określa listę użytkowników, których dotyczy ma zezwozenie, zaś **pochodzenie** oznacza listę terminali logowania, nazwy hostów, adresy hostów lub nazwy domen.

```
# Przykładowy plik /etc/security/access.conf

-:root:ALL EXCEPT LOCAL .tk.krakow.pl
#Zabrania logowania administratora ze wszystkich hostów oprócz hostów
#lokalnych i z hostow domeny tk.krakow.pl

-:school:ALL EXCEPT LOCAL
#Zabrania logowania członkom grupy school logowania z hostów nie
#należących do sieci lokalnej
```

Plik /etc/security/limits.conf

Plik */etc/security/limits.conf* umożliwia wprowadzenie różnych przydatnych ograniczeń dla użytkowników systemu.

Każdy wpis w tym pliku ma następujący format:

```
<dziedzina> <typ> <rzecz> <wartość>
```

gdzie **dziedzina** określa nazwę użytkownika lub nazwę grupy (ze znakiem @), **typ** przyjmuje dwie wartości: **soft** dla wprowadzania w życie limitów, po przekroczeniu których system zacznie wyświetlać komunikaty o ich przekroczeniu i **hard** dla limitów, po przekroczeniu których system zablokuje możliwość dalszego ich przekraczania; **rzecz** określa, czego dotyczy limit i może przyjmować wartości: **core** (limity plików *core* w KB), **data** (maksymalna wielkość danych w KB), **fsize** (maksymalny rozmiar plików w KB), **memlock** (maksymalna przestrzeń adresowa ulokowana w pamięci w KB), **nofile** (maksymalna liczba otwartych plików), **rss** (maksymalny rozmiar używanej pamięci w KB), **stack** (maksymalny rozmiar stosu w KB), **cpu** (maksymalny czas procesora w minutach), **nproc** (maksymalna liczba uruchomionych procesów), **as** (limit przestrzeni adresowej), **maxlogins** (maksymalna liczba zalogowań) oraz **priority** (priorytet, z jakim uruchamiane są procesy), natomiast **wartość** określa wartość przypisanego limitu.

```
# Przykładowy plik /etc/security/limits.conf

*                soft          core          0
#Ustawia miękki limit dla plików core dla wszystkich użytkowników

@user            -              maxlogins     4
#Ustawia limit liczby zalogowań na 4 dla członków grupy user

@school          hard          nproc         20
#Ustawia twardy limit uruchomionych procesów na 20 dla członków grupy
#school
```

FIREWALL

Zapora sieciowa (*firewall*) to konstrukcja zapewniająca kontrolowane połączenie pomiędzy siecią lokalną i Internetem (siecią publiczną). Dostarcza ona mechanizmu kontroli ilości i rodzaju ruchu sieciowego między obydwoma sieciami.

Podstawowe funkcje, które powinna spełniać zapora sieciowa to:

- zapewnienie „bezpiecznego” dostępu do Internetu użytkownikom sieci prywatnej
- zapewnienie ochrony sieci prywatnej przed atakami z zewnątrz
- blokowanie dostępu do określonych miejsc w Internecie
- monitorowanie komunikacji pomiędzy siecią prywatną a Internetem
- rejestrowanie całości lub określonej części ruchu międzysieciowego.

Klasyczny filtr pakietowy to zbiór reguł określających, co system powinien zrobić z pakietem przychodzącym z sieci, lub do niej wychodzącym. Podstawą działania filtrów jest bieżące śledzenie i analiza przechodzących przez dany węzeł połączeń. Filtr cały czas przechowuje w pamięci informacje na temat aktualnego stanu każdego połączenia, wiedząc, jakie stany są dozwolone z punktu widzenia protokołu i polityki bezpieczeństwa.

Filtr jest sterowany zbiorem reguł, których podstawowymi elementami są wzorce oraz akcje mówiące, co zrobić z pakietem pasującym do danej reguły. Przetwarzanie tych reguł odbywa się dla każdego pakietu przychodzącego lub wychodzącego z danego węzła. Pakiet pasujący do określonego w danej regule wzorca jest traktowany zgodnie z przypisaną do niego akcją. Z reguły ogranicza się ona do przepuszczenia lub zablokowania pakietu, z ewentualnym odesłaniem odpowiedniego komunikatu ICMP.

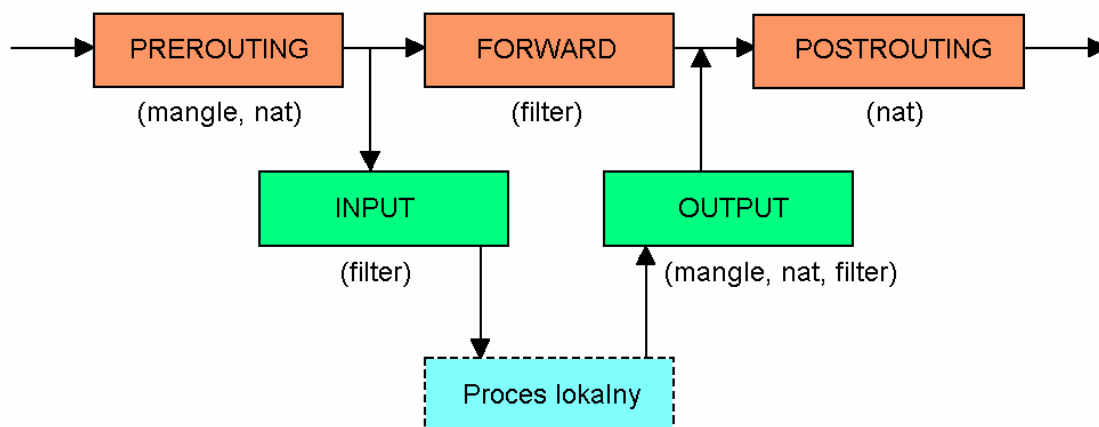
Polecenie *iptables* ma w większości przypadków postać:

```
iptables [-t <tablica>] <operacja> [<dopasowanie>] [<cel/skok>]
```

Tablice

W jądrach Linuksa w wersji 2.4 jest dostępny mechanizm filtrowanie pakietów i translacji adresów znany pod nazwą *IPTables*. Całość tego mechanizmu została podzielona na logiczne klasy-tablice (*tables*). Każda z nich zawiera predefiniowane zbiory reguł. Zaimplementowane są następujące tablice:

- **filter**
Przeznaczona jest głównie do budowy zapór filtrujących, kontroli i zliczania ruchu w sieci, diagnostyki sieci
- **nat**
Przeznaczona jest głównie do maskowania adresów IP (*IP masquerading*), przekierowania portów (*port forwarding*), budowy przezroczystych proxy (*transparent proxying*)
- **mangle**
Przeznaczona jest głównie do kontroli przepływu danych (ograniczanie pasma).



Rys. 22. Przepływ pakietów przez tablice IPTables

Tablica	Zbiory reguł	Funkcja
filter	INPUT	Zbiór reguł dla pakietów przeznaczonych dla lokalnego hosta
	OUTPUT	Zbiór reguł dla pakietów pochodzących z lokalnego hosta
	FORWARD	Zbiór reguł dla pakietów przechodzących przez lokalny router
nat	PREROUTING	Zbiór reguł dla pakietów przychodzących z zewnątrz hosta
	OUTPUT	Zbiór reguł dla pakietów pochodzących z lokalnego hosta
	POSTROUTING	Zbiór reguł dla pakietów wychodzących na zewnątrz hosta
mangle ¹	PREROUTING	Zbiór reguł dla pakietów przychodzących z zewnątrz hosta
	OUTPUT	Zbiór reguł dla pakietów pochodzących z lokalnego hosta

Tab. 53. Zbiory reguł i ich funkcje dla tablic

Stany pakietów

Pakiety mogą przyjmować kilka różnych stanów wewnątrz swojego jądra, w zależności od protokołu, jakiego używają. Jakkolwiek, na zewnątrz jądra, istnieją tylko cztery stany pakietów. Stany te mogą głównie być używane wraz ze stanami dopasowania, które będą później zdolne do dopasowania pakietów bazujących na ich bieżącym połączeniu.

Możliwe stany pakietu to: NEW (nowy), ESTABLISHED (ustalony), RELATED (spokrewniony) i INVALID (błędny).

Moduł ścieżki połączenia przechowuje wszystkie stany na bieżąco w pamięci, według różnych zasad. Moduł ten określa jeden z czterech możliwych stanów pakietu TCP i UDP oraz kilka dodatkowych wartości. Podstawowe wartości używane do określania stanu dla potoków TCP i UDP są: źródłowy adres IP, docelowy adres IP, źródłowy port i port docelowy.

¹ Od wersji jądra 2.4.18 dla tablicy **mangle** dostępne są także reguły: **INPUT** (reguły dla pakietów przeznaczonych dla lokalnego hosta), **FORWARD** (reguły dla pakietów przechodzących przez lokalny router) i **POSTROUTING** (reguły dla pakietów wychodzących na zewnątrz hosta).

Stan	Opis
NEW	Oznacza, że pakiet jest nowy w połączeniu, czyli pierwszy pakiet, który wędruje do hosta.
ESTABLISHED	Oznacza, że ruch pakietów odbywa się w obu kierunkach. Połączenie NEW zyskuje status ESTABLISHED, kiedy po wysłaniu pierwszego pakietu nastąpi odpowiedź na niego.
RELATED	Oznacza, że pakiet jest spokrewniony z pakietami typu ESTABLISHED. Kiedy połączenie ESTABLISHED daje początek nowemu połączeniu, to nowe połączenie zyskuje miano RELATED.
INVALID	Oznacza, że pakiet nie może być zidentyfikowany lub nie ma żadnego stanu. Mogą być różne przyczyny takiego stanu pakietu, np. przepełnienie pamięci systemu.

Tab. 54. Opis stanów pakietów

Łańcuchy

Łańcuch (*chain*) to lista reguł. Każda reguła określa działanie, jakie zostanie podjęte na podstawie nagłówka pakietu. Jeśli reguła nie pasuje do pakietu, sprawdzana jest następna. Na koniec, jeśli nie ma więcej reguł, stosowana jest zasada domyślna (*policy*) danego łańcucha. W systemie, w którym dba się o bezpieczeństwo, zasada powinna mówić jądro, by odrzucić pakiet.

Operacje

Operacje decydują, co zrobić z dalszą częścią linii komend, która jest przekazywana do *iptables*.

Operacja	Działanie
-N (--new-chain)	Stworzenie nowego łańcucha
-X (--delete-chain)	Skasowanie pustego łańcucha
-P (--policy)	Zmiana zasady dla wbudowanego łańcucha
-L (--list)	Wylistowanie reguł w łańcuchu
-F (--flush)	Wyczyszczenie reguł z łańcucha
-Z (--zero)	Wyzerowanie liczników pakietów i bajtów we wszystkich regułach w łańcuchu

Tab. 55. Lista operacji na całych łańcuchach

Operacja	Działanie
-A (--append)	Dodanie nowej reguły do łańcucha
-I (--insert)	Wstawienie nowej reguły na wskazanej pozycji w łańcuchu
-R (--replace)	Wymiana reguły na wskazanej pozycji w łańcuchu
-D (--delete)	Skasowanie pierwszej pasującej reguły z łańcucha

Tab. 56. Manipulacja regułami w środku łańcuchów

Dopasowania ogólne

Dopasowania ogólne są rodzajem dopasowań, które są zawsze dostępne niezależnie od rodzaju pracującego protokołu czy załadowanych dopasowań rozszerzonych. Do załadowania dopasowania ogólnego, nie są wymagane żadne dodatkowe parametry.

Dopasowanie	Przykład	Działanie
-p (--protocol)	<i>iptables -A INPUT -p tcp</i>	Używane do sprawdzania protokołów. Protokół może być podany też jako wartość liczbowa (z pliku <i>/etc/protocols</i>). Wpis może zawierać znak negacji (!) przed protokołem, co oznacza wszystkie inne protokoły oprócz podanego.
-s (--source)	<i>iptables -A INPUT -s 192.168.1.1</i>	Oznacza źródło (adres) pochodzenia pakietów. Wpis może zawierać znak negacji (!) przed adresem, co oznacza wszystkie inne adresy oprócz podanego. Domyślnie oznaczane są wszystkie adresy IP.
-d (--destination)	<i>iptables -A INPUT -d 192.168.1.1</i>	Oznacza miejsce przeznaczenia (docelowe) pakietów. Ma taką samą składnię jak dopasowanie -s.
-i (--in-interface)	<i>iptables -A INPUT -i eth0</i>	Oznacza interfejs, z którego pochodzą pakiety. To dopasowanie jest zgodne tylko z łańcuchami PREROUTING, FORWARD i INPUT. Wpis może zawierać znak negacji (!) przed interfejsem, co oznacza wszystkie inne interfejsy oprócz podanego.
-o (--out-interface)	<i>iptables -A FORWARD -o eth0</i>	Oznacza interfejs, do którego mają trafić pakiety. To dopasowanie jest zgodne tylko z łańcuchami, FORWARD, OUTPUT i POSTROUTING. Ma taką samą składnię jak oznaczenie -i.
-f (--fragment)	<i>iptables -A INPUT -f</i>	Oznacza drugą i kolejne części podzielonych pakietów. Oznacz się je ponieważ nie ma innej możliwości przekazania do miejsca przeznaczenia lub pochodzenia informacji o fragmentach. Wpis może zawierać znak negacji (!) przed dopasowaniem, co oznacza tylko pierwszy z podzielonych pakietów.

Tab. 57. Opis dopasowań ogólnych wraz z przykładami

Dopasowania pośrednie

Istnieją trzy typy dopasowań pośrednich, które są ładowane automatycznie dla trzech różnych protokołów. Są to dopasowania TCP, UDP i ICMP.

Dopasowania TCP

Dopasowania TCP są określone dla protokołu TCP i dostępne są tylko wtedy, kiedy pracują wraz z pakietami i strumieniami TCP. Do użycia tych dopasowań, należy określić w linii komend dopasowanie ogólne `--protocol tcp`.

Dopasowanie	Przykład	Działanie
<code>--sport</code> (<code>--source-port</code>)	<code>iptables -A INPUT -p tcp --sport 22</code>	Określa port źródłowy pakietów. Wpis może zawierać port w postaci numerycznej lub jako nazwa usługi z pliku <code>/etc/services</code> . Jednak jeśli określa się bardzo dużo zasad, lepiej jest stosować porty w postaci numerycznej. Można określić zakres portów oddzielając wartości graniczne dwukropkiem.
<code>--dport</code> (<code>--destination-port</code>)	<code>iptables -A INPUT -p tcp --dport 22</code>	Określa port docelowy pakietów. Ma taką samą składnię jak dopasowanie <code>--sport</code> .
<code>--tcp-flags</code>	<code>iptables -p tcp --tcp-flags SYN,ACK,FIN SYN</code>	Określa zależne flagi TCP w pakiecie. Pierwsze z dopasowań wczytuje listę flag do porównania, a drugie – wczytuje listę flag, które powinny być ustawione na wartość 1 lub włączone. Obie listy powinny być wewnątrz oddzielone przecinkami. Możliwe do użycia flagi to SYN, ACK, FIN, RST, URG, PSH. Ale możliwe do użycia wpisy to także ALL (wszystkie flagi) i NONE (żadna flaga).
<code>--syn</code>	<code>iptables -p tcp --syn</code>	Określa dopasowania pakietów, jeśli mają bit SYN i nie ustawione bity ACK i FIN. Takie pakiety są używane na potrzebę nowych połączeń TCP, głównie z serwera. Jeśli te pakiety zostaną zablokowane, powinno to skutecznie zablokować wszystkie próby połączeń przychodzących.
<code>--tcp-option</code>	<code>iptables -p tcp --tcp-option 16</code>	Określa zależności pakietów w ich opcjach TCP. Opcje TCP są określoną częścią nagłówka. Ta część składa się z 3 różnych pól. Pierwsze zawiera 8 bitów i określa, które opcje są użyte w tym strumieniu. Drugie pole też zawiera 8 bitów i określa, jak długie jest pole opcji. To dopasowanie jest używane do określenia różnych opcji TCP zależnych od ich wartości dziesiętnych.

Tab. 58. Opis dopasowań TCP

Dopasowania UDP

Dopasowania UDP są dostępne tylko wtedy, kiedy pracują wraz z pakietami UDP. Do użycia tych dopasowań, należy określić w linii komend dopasowanie ogólne `--protocol udp`.

Dopasowanie	Przykład	Działanie
<code>--sport</code> (<code>--source-port</code>)	<code>iptables -A INPUT -p udp --sport 53</code>	Określa port źródłowy pakietów. Wpis może zawierać port w postaci numerycznej lub jako nazwa usługi z pliku <code>/etc/services</code> . Można określić zakres portów oddzielając wartości graniczne dwukropkiem.
<code>--dport</code> (<code>--destination-port</code>)	<code>iptables -A INPUT -p udp --dport 53</code>	Określa port docelowy pakietów. Ma taką samą składnię jak dopasowanie <code>--sport</code> .

Tab. 59. Opis dopasowań UDP

Dopasowania ICMP

Dopasowania ICMP są dostępne tylko wtedy, kiedy pracują wraz z protokołem ICMP. Do użycia tych dopasowań, należy określić w linii komend dopasowanie ogólne `--protocol icmp`.

Dopasowanie	Przykład	Działanie
<code>--icmp-type</code>	<code>iptables -A INPUT -p icmp --icmp-type 8</code>	Określa typ ICMP. Typy ICMP mogą być określone przez wartości numeryczne lub przez ich nazwy. By uzyskać listę możliwych typów ICMP należy wydać polecenie: <code>iptables -p icmp -help</code> . Wpis może zawierać znak negacji (!) przed typem ICMP, co oznacza wszystkie inne typy oprócz podanego.

Tab. 60. Opis dopasowań ICMP

Dopasowania wyraźne

Dopasowania wyraźne to dopasowania, które muszą być specjalnie ładowane z opcją `-match`. Niektóre z tych dopasowań mogą być specyficzne dla niektórych protokołów lub zostały utworzone do celów testowych. Wszystkie te dopasowania nie zawsze mogą być użyteczne. Różnica pomiędzy dopasowaniami pośrednimi a dopasowaniami wyraźnymi polega na tym, że dopasowania pośrednie są automatycznie ładowane, zaś dopasowania wyraźne w żadnym przypadku nie są ładowane automatycznie i należy je uruchomić przed ich użyciem.

Dopasowania MAC

Dopasowania MAC są używane to dopasowania pakietów, których podstawą jest adres źródłowy MAC interfejsu sieciowego. Do użycia tych dopasowań należy określić w linii komend dopasowanie **-m mac**.

Dopasowanie	Przykład	Działanie
<code>--mac-source</code>	<code>iptables -A INPUT -m mac --mac-source 00:00:00:00:00:00</code>	Określa adres źródłowy MAC interfejsu sieciowego. Adres MAC musi być podany w formie <code>XX:XX:XX:XX:XX:XX</code> . To dopasowanie jest zgodne tylko z łańcuchami <code>INPUT</code> , <code>FORWARD</code> i <code>PREROUTING</code> .

Tab. 61. Opis dopasowań MAC

Dopasowania limitujące

Dopasowania limitowe są stosowane do ograniczenia zapisu dzienników na określonych zasadach. Oznacza to, że za pomocą tego dopasowania można ograniczać to, jak wiele razy pewna zasada może być dopasowana do pewnego okresu czasu. Do użycia tych dopasowań należy określić w linii komend dopasowanie **-m limit**.

Dopasowanie	Przykład	Działanie
<code>--limit</code>	<code>iptables -A INPUT -m limit 3/hour</code>	Ustawia maksymalne średnie tempo ograniczenia dopasowań. To dopasowanie jest określone wraz z liczbą i opcjonalnie określeniem czasu. Dostępne są następujące znaczniki czasu: <code>/second</code> , <code>/minute</code> , <code>/hour</code> , <code>/day</code> . Domyślną wartością jest 3 na godzinę. Oznacza to, ile razy pozwolić uruchomić dopasowanie w ciągu określonego czasu.
<code>--limit-burst</code>	<code>iptables -A INPUT -m limit -limit-burst 5</code>	Określa maksymalną początkową ilość pakietów w dopasowaniu. Ilość jest ponownie ładowana za każdym razem, kiedy średnie określone ograniczenie (przez dopasowanie <code>--limit</code>) nie jest osiągalne, aż do ilości określonej wartością <code>--limit-burst</code> . Kiedy limit przebicia jest osiągalny, obniża się najniższy możliwy poziom ograniczenia. Po tym, przydzielany jest jeden „żeton” dla każdej określonej jednostki czasowej, której nie obciąża się nowym limitem, zanim wartość limitu nie dotrze ponownie do wartości limitu przebicia. Domyślna wartość <code>--limit-burst</code> wynosi 5.

Tab. 62. Opis dopasowań limitujących

Dopasowania wieloportowe

Dopasowania wieloportowe używane są do określenia większej ilości portów przeznaczenia i zasięgu portów niż jeden. Stosuje się je najczęściej do zbudowania kilku zasad wyglądających dokładnie tak samo, tylko że dla różnych portów. Do użycia tych dopasowań należy określić w linii komend dopasowanie **-m multiport**.

Dopasowanie	Przykład	Działanie
<code>--source-port</code>	<code>iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110</code>	Określa zbiorowe porty źródłowe. Może być określonych jednocześnie 15 oddzielonych od siebie portów. Porty muszą być oddzielone przecinkami. To dopasowanie może być tylko używane z dopasowaniami <code>-p tcp</code> lub <code>-p udp</code> . To dopasowanie jest rozszerzoną wersją dopasowania <code>--source-port</code> .
<code>--destination-port</code>	<code>iptables -A INPUT -p tcp -m multiport --destination-port 22,53,80,110</code>	Określa zbiorowe porty docelowe. Ma taką samą składnię, jak dopasowanie <code>-m multiport --source-port</code> . To dopasowanie może być tylko używane z dopasowaniami <code>-p tcp</code> lub <code>-p udp</code> . To dopasowanie jest rozszerzoną wersją dopasowania <code>--destination-port</code> .
<code>--port</code>	<code>iptables -A INPUT -p tcp -m multiport --port 22,53,80,110</code>	Określa pakiety, których podstawą są zarówno porty źródłowe i docelowe. Pracuje dokładnie, jak dopasowania <code>-m multiport --source-port</code> oraz <code>-m multiport --destination-port</code> połączone razem i ma taką składnię jak te dwa dopasowania. To dopasowanie może być tylko używane z dopasowaniami <code>-p tcp</code> lub <code>-p udp</code> .

Tab. 63. Opis dopasowań wieloportowych

Dopasowania zaznaczające

Dopasowania zaznaczające są stosowane do dopasowania pakietów na podstawie zaznaczeń, które mają ustawione. Zaznaczenie jest specjalnym polem utrzymywanym wewnątrz jądra, które jest kojarzone z pakietami. Mogą być one użyte przez różne układy jądra dla takich zadań jak ruch formujący i filtrowanie. Pole zaznaczenia jest obecnie ustawione na liczbę całkowitą typu `unsigned`, czyli na jedną z 2^{16} możliwych wartości lub też na jedną z 2^{32} możliwych wartości (w systemie 32-bitowym). Do użycia tych dopasowań należy określić w linii komend dopasowanie **-m mark**.

Dopasowanie	Przykład	Działanie
--mark	<i>iptables -t mangle -A INPUT -m mark --mark 1</i>	Określa pakiety, które uprzednio zostały zaznaczone. Zaznaczenia mogą być ustawiane obiektem MARK. Wszystkie pakiety przechodzące przez filtr sieciowy ma specjalne zaznaczenie skojarzone z nimi. Zaznaczenia są liczbami całkowitymi typu unsigned. Z zaznaczeniem może być też użyta maska.

Tab. 64. Opis dopasowań zaznaczających

Dopasowania własnościowe

Dopasowania własnościowe są używane do dopasowania pakietów na podstawie ich właścicieli. To dopasowanie pracuje tylko z łańcuchem OUTPUT. Do użycia tych dopasowań należy określić w linii komend dopasowanie -m owner.

Dopasowanie	Przykład	Działanie
--uid-owner	<i>iptables -A OUTPUT -m owner --uid-owner 500</i>	Określa pakiet, jeśli został utworzony przez danego użytkownika. Może to być użyte do dopasowania wychodzących zaznaczanych pakietów na podstawie tego, kto je utworzył.
--gid-owner	<i>iptables -A OUTPUT -m owner --gid-owner 0</i>	Określa wszystkie pakiety, które zostały utworzone przez daną grupę. Może to służyć do zablokowania jakiejś grupie użytkowników wyjścia poza sieć lokalną.
--pid-owner	<i>iptables -A OUTPUT -m owner --pid-owner 78</i>	Określa pakiety na podstawie procesu, przez który zostały utworzone.
--sid-owner	<i>iptables -A OUTPUT -m owner --sid-owner 100</i>	Określa pakiety na podstawie sesji, przez którą zostały utworzone. Wartość identyfikatora sesji (sid) jest ustawiona na różne procesy zależne od procesów, z których powstały (jeśli są procesami potomnymi) lub na procesy, które go utworzyły.

Tab. 65. Opis dopasowań własnościowych

Dopasowania stanu

Dopasowania stanu są stosowane wraz z kodem śledzonego połączenia w jądrze i pozwalają na dostęp do stanu pakietów śledzonego połączenia. Do użycia tych dopasowań należy określić w linii komend dopasowanie -m state.

Dopasowanie	Przykład	Działanie
--state	<i>iptables -A INPUT -m state RELATED, ESTABLISHED</i>	Określa pakiety na podstawie ich stanów. Stany pakietów (NEW, ESTABLISHED, RELATED i INVALID) zostały omówione wcześniej.

Tab. 66. Opis dopasowań stanu

Dopasowania TOS

Dopasowania TOS są używane do dopasowania pakietów na podstawie pola TOS (*Type Of Service*). Rodzaj usługi (TOS) składa się z 8 bitów i znajduje się w nagłówku IP. TOS jest używane do poinformowania pośrednich poprzedzających hostów o strumieniu i rodzaju zawartości. Do użycia tych dopasowań należy określić w linii komend dopasowanie `-m tos`.

Dopasowanie	Przykład	Działanie
<code>--tos</code>	<code>iptables -A INPUT -p tcp -m tos -tos 0x16</code>	Określa pakiety na podstawie pola TOS i jego wartości. To dopasowanie może być użyte między innymi do zaznaczania pakietów do zaawansowanych opcji wyboru marszruty. Dopasowania zawierają heksadecymalne lub liczbowe wartości jako opcje, lub też jedną z nazw uzyskanych po wydaniu polecenia: <code>iptables -m tos -h</code> . Możliwe są opcje: <ul style="list-style-type: none"> ▪ Minimize-Delay (0x10) Minimalizuje opóźnienie dla pakietów ▪ Maximize-Throughput (0x08) Znajduje ścieżkę, która umożliwi tak dużą przepustowość jak to możliwe ▪ Maximize-Reliability (0x04) Maksymalizuje niezawodność połączenia i używa tras, które są najbardziej pewne ▪ Minimize-Cost (0x02) Minimalizuje koszty zanim pakiety uzyskają połączenie z danym hostem ▪ Normal-Service (0x00) Określa normalne protokoły, które nie potrzebują specjalnych opcji transferowych.

Tab. 67. Opis dopasowań TOS

Dopasowania TTL

Dopasowania TTL są używane do dopasowywania pakietów na podstawie ich pola TTL (*Time-To-Live*) znajdującego się w nagłówku IP. Pole TTL składa się z 8 bitów i jego wartość jest obniżana za każdym razem, kiedy przechodzi przez pośrednie hosty. Do użycia tych dopasowań należy określić w linii komend dopasowanie `-m ttl`.

Dopasowanie	Przykład	Działanie
<code>--ttl</code>	<code>iptables -A OUTPUT -m ttl --ttl 60</code>	Określa pakiety na podstawie ich wartości TTL. Dopasowanie to może być używane w celu rozwiązywania problemów w sieci lokalnej.

Tab. 68. Opis dopasowań TTL

Cele i skoki

Cele i skoki są używane do przekazania zasad dotyczących pakietów, które dokładnie pasują do danego dopasowania. Skok jest dokładnie tym samym, co cel z wyjątkiem tego, że wymaga łańcucha wewnątrz tej samej tablicy, do której ma być wykonany skok. Aby zastosować w danej regule cel lub skok należy użyć opcji -j.

Cel ACCEPT

Cel ten nie posiada specjalnych opcji. Kiedy pakiet dokładnie pasuje do danego dopasowania i ustawiony jest cel **ACCEPT**, pakiet ten jest akceptowany i nie kontynuuje wędrówki poprzez łańcuch od miejsca, kiedy został zaakceptowany. Pakiet, który został zaakceptowany w jednym łańcuchu, nadal wędruje przez inne łańcuchy, gdzie może być odrzucony.

Cel DROP

Cel **DROP** odrzuca pakiety i odmawia dalszego ich przetwarzania. Pakiet, który dokładnie pasuje do danego dopasowania i ustawiony jest cel **DROP** zostanie zablokowany i nie będzie podjęte żadne dalsze działanie na nim. Pakiet taki nie będzie wędrował przez inne łańcuchy ani tablice.

Cel QUEUE

Cel **QUEUE** jest używany do kolejkowania pakietów w drodze do programów i aplikacji użytkowników. Cel ten przeznaczony jest do zastosowań programistycznych i może być używany na przykład do zliczania ruchu sieciowego albo do tworzenie specyficznych i zaawansowanych aplikacji do filtrowania pakietów.

Cel RETURN

Cel ten powoduje przerwanie wędrówki pakietu przez dany łańcuch w miejscu, gdzie znajduje się dana zasada. Jeśli jest to podłańcuch do innego łańcucha, pakiet będzie kontynuował wędrówkę przez łańcuchy w strukturze, które nie są głównym łańcuchem. Jeśli łańcuch, w którym występuje ten cel, jest łańcuchem głównym, na pakiecie zostanie dokonana domyślna polityka danego łańcucha (najczęściej **ACCEPT** lub **DROP**).

Cel LOG

Cel **LOG** używany jest w celu rejestrowania informacji o pakietach, które są niedozwolone, lub też w celu znajdowania błędów w procesie filtrowania. Cel ten zawiera informacje takie jak nagłówek IP i inne informacje poprzez funkcje rejestracyjne jądra. Informacje te mogą być czytane przez *dmesg* lub *syslogd* oraz przez podobne aplikacje. Cel **LOG** doskonale sprawdza się podczas rozwiązywania problemów z ustalonymi regułami.

Cel **LOG** zawiera obecnie pięć opcji, które pozwalają na podanie większej ilości informacji oraz na ustawienie różnych wartości.

Opcja	Przykład	Działanie
--log-level	<i>iptables -A FORWARD -p tcp -j LOG --log-level debug</i>	Określa, jakiego poziomu rejestracji ma użyć <i>iptables</i> i <i>syslog</i> . Można użyć następujących poziomów rejestrowania: <ul style="list-style-type: none"> ▪ debug ▪ info ▪ notice ▪ warning (warn) ▪ error (err) ▪ crit ▪ alert ▪ emerg (panic).
--log-prefix	<i>iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"</i>	Informuje o prefiksie rejestrowanych przez <i>iptables</i> komunikatów. Prefiks może zawierać do 29 znaków. Opcja ta ma najczęściej zastosowanie wraz z poleceniami uniksowymi, jak <i>grep</i> .
--log-tcp-sequence	<i>iptables -A INPUT -p tcp -j LOG --log-tcp-sequence</i>	Rejestruje sekwencję TCP dla komunikatów. Sekwencja TCP to specjalne numery, które identyfikują każdy pakiet oraz informacje, gdzie powinien pasować do TCP i jak pakiet powinien być odtworzony.
--log-tcp-options	<i>iptables -A FORWARD -p tcp -j LOG --log-tcp-options</i>	Rejestruje różne opcje z nagłówka pakietów TCP. Opcja ta może być pomocna przy usuwaniu błędów.
--log-ip-options	<i>iptables -A FORWARD -p tcp -j LOG --log-ip-options</i>	Rejestruje większość opcji nagłówka pakietów IP. Działa tak samo jak opcja --log-tcp-options, z tym że dotyczy pakietów IP.

Tab. 69. Opcje dla celu LOG

Cel MARK

Cel MARK jest używany do ustawiania wpisów wartości, które są skojarzone z określonymi pakietami. Cel ten może być zdefiniowany tylko dla tablicy *mangle*.

Opcja	Przykład	Działanie
--set-mark	<i>iptables -t mangle -A PREROUTING -p tcp --dport 22 -j MARK --set-mark 2</i>	Opcja ta jest niezbędna do ustawienia poziomu i wyraża się liczbą całkowitą.

Tab. 70. Opcje dla celu MARK

Cel REJECT

Cel ten zasadniczo działa tak samo jak cel *DROP*, z tą różnicą, że wysyłają z powrotem wiadomość o błędzie do hostu wysyłającego pakiet, który został zablokowany. Cel *REJECT* może współpracować tylko z łańcuchami *INPUT*, *FORWARD* i *OUTPUT*.

Opcja	Przykład	Działanie
<code>--reject-with</code>	<code>iptables -A FORWARD -p tcp --dport 22 -j REJECT --reject-with tcp-reset</code>	Określa, jaką odpowiedź należy wysłać do hostu, od którego został otrzymany odrzucony pakiet. Istnieją następujące typy odrzucenia pakietów: <ul style="list-style-type: none"> ▪ <code>icmp-net-unreachable</code> ▪ <code>icmp-host-unreachable</code> ▪ <code>icmp-port-unreachable</code> ▪ <code>icmp-proto-unreachable</code> ▪ <code>icmp-net-prohibited</code> ▪ <code>icmp-host-prohibited</code> ▪ <code>echo-reply</code> (tylko z pakietami ICMP ping) ▪ <code>tcp-reset</code> (tylko z pakietami TCP). Domyślną odpowiedzią o błędzie do hosta źródłowego jest <code>port-unreachable</code> .

Tab. 71. Opcje dla celu REJECT

Cel MIRROR

Cel ten jest używany do odwrócenia w nagłówku IP pola odbiorcy na pole nadawcy, co powoduje że pakiet jest wysyłany z powrotem do hosta, który go przysłał.

Cel SNAT

Cel SNAT (*Source Network Address Translation*) może być zdefiniowany tylko dla łańcucha POSTROUTING tablicy `nat` i powoduje, że dla danego pakietu zamieniany jest adres źródłowy.

Opcja	Przykład	Działanie
<code>--to-source</code>	<code>iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to source 81.21.195.159:1024-32000</code>	Określa źródło pakietów. Opcja ta, bierze jeden adres IP, do którego powinny być kierowane wszystkie źródłowe adresy IP.

Tab. 72. Opcje dla celu SNAT

Cel DNAT

Cel DNAT (*Destination Network Address Translation*) może być zdefiniowany jedynie dla tablicy `nat` i powoduje, że dla danego pakietu zamieniany jest adres docelowy.

Opcja	Przykład	Działanie
<code>--to-destination</code>	<code>iptables -t nat -A PREROUTING -p tcp -d 81.21.195.159 --dport 80 -j DNAT --to-destination 192.168.1.10</code>	Określa adres IP przeznaczenia pakietów, co powoduje, że wszystkie pakiety pasujące do wzorca, będą przesyłane na podany opcją adres IP.

Tab. 73. Opcje dla celu DNAT

Cel MASQUERADE

Cel ten definiowany jest tylko dla łańcucha POSTROUTING tablicy nat i powoduje, że dla danego pakietu zamieniany jest adres źródłowy. Różnica między tym celem a celem SNAT polega na tym, że w celu MASQUERADE adres źródłowy zamieniany jest na adres interfejsu, do którego skierowany zostanie pakiet. Dlatego cel ten może być stosowany przy dynamicznym przydziale adresów IP.

Opcja	Przykład	Działanie
--to-ports	<code>iptables -t nat -A POSTROUTING -p tcp -j MASQUERADE --to-ports 1024-31000</code>	Ustawia źródłowy port lub porty, które będą używane w wychodzących pakietach.

Tab. 74. Opcje dla celu MASQUERADE

Cel REDIRECT

Cel REDIRECT jest używany do przeadresowania pakietów i strumieni danych.

Opcja	Przykład	Działanie
--to-ports	<code>iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080</code>	Określa port lub zbiór portów przeznaczenia, na które będą przeadresowane pakiety.

Tab. 75. Opcje dla celu REDIRECT

Konfiguracja iptables

Aby w pełni wykorzystać możliwości programu *iptables*, należy utworzyć wykonywalny skrypt, a jego ścieżkę dodać do domyślnego katalogu uruchamiania usług przez proces *init*. Spowoduje to, że ustawienia lokalnej zapory, będą każdorazowo ustawiane, jeśli tylko nastąpi ponowne uruchomienie serwera.

```
# Prosty skrypt firewalla

iptables -F
iptables -P OUTPUT ACCEPT
iptables -t nat -F
iptables -t mangle -F
#Wyczyszczenie łańcuchów i ustawienie domyślnej polityki dla łańcucha
#OUTPUT

iptables -t nat -I POSTROUTING -s 192.168.1.0/255.255.255.0 -j SNAT
--to-source 81.21.195.159
echo 1 > /proc/sys/net/ipv4/ip_forward
#Włączenie translacji adresów sieciowych z LAN

iptables -A INPUT -i eth1 -s 192.168.1.0/255.255.255.0 -j ACCEPT
iptables -A FORWARD -i eth1 -s 192.168.1.0/255.255.255.0 -j ACCEPT
iptables -A FORWARD -o eth1 -d 192.168.1.0/255.255.255.0 -j ACCEPT
#Włączenie przekazywania adresów dla LAN

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#Wpuszczenie przez zaporę tylko pakietów, których status jest znany
#lub określony
```



```
iptables -A INPUT -p tcp --dport 21 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -j ACCEPT
iptables -A INPUT -p tcp --dport 113 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
#Wpuszczenie przez zapore pakietow dla uslug serwera:
#FTP, SSH, DNS, HTTP, SMTP, POP3, IDENT

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
#Umozliwienie pingowania hosta

iptables -A INPUT -j LOG -m limit --limit 10/hour
iptables -A FORWARD -j LOG -m limit --limit 10/hour
#Ustawienie limitow logowania pakietow, ktore nie przeszly przez
#firewall

iptables -P INPUT DROP
iptables -P FORWARD DROP
#Ustawienie domyslnej polityki dla lancuchow INPUT i FORWARD
```

7. ADMINISTROWANIE SERWERA

Rola administratora jest fachowa opieka nad systemem dla utrzymania jego maksymalnej efektywności, ochrony jego zasobów i zapewnienia każdemu użytkownikowi indywidualnie wymaganych przez niego usług i dostępu do oprogramowania.

Zadania, jakie stoją przed administratorem systemu to w ogólności:

- utrzymanie spójności systemu
- dokonywanie okresowych składowań systemu plików
- obsługa problemów związanych z ograniczonymi zasobami systemu
- dokonywanie aktualizacji systemu operacyjnego
- zapewnienie ogólnej pomocy użytkownikom
- usuwanie uszkodzeń w systemie
- zapis wszystkich modyfikacji systemu i zdarzeń w dzienniku systemu
- dbanie o bezpieczeństwo systemu

ZARZĄDZANIE UŻYTKOWNIKAMI

Użytkownicy są głównym powodem istnienia systemów komputerowych. Istnieje wiele czynności związanych z administracją użytkownikami w systemie komputerowym, jak choćby tworzenie lub usuwanie użytkowników.

Ponieważ UNIX jest systemem wielodostępnym, bardzo ważne jest rozróżnianie między jego użytkownikami. Jest to główny powód, dla którego tworzy się oddzielne konta dla każdego użytkownika systemu. Konto jest zbiorem informacji, które określają, kim jest użytkownik oraz jakie działania są dla niego dozwolone.

Element	Opis
nazwa konta	Unikalna nazwa konta użytkownika.
hasło	Hasło dostępu do konta jest kluczem pozwalającym na wykorzystanie konta przez użytkownika. Każdy użytkownik powinien posiadać hasło dostępu do systemu. Hasło w systemie jest przechowywane w formie zaszyfrowanej.
identyfikator użytkownika (UID)	Unikalna liczba przypisana do każdego użytkownika w systemie. Jest ona wykorzystywana przez system do śledzenia informacji związanych z użytkownikiem.
grupa	Zbiór użytkowników posiadających pewne cechy wspólne, do której należy dany użytkownik.
identyfikator grupy (GID)	Identyfikator domyślnej grupy, do której należy użytkownik – liczba podobna do UID, z tą różnicą, że GID identyfikuje grupę.
komentarz	Prawdziwa nazwa użytkownika oraz dodatkowe informacje o użytkowniku.
katalog macierzysty	Początkowy katalog, do którego użytkownik uzyskuje dostęp natychmiast po wykonaniu procedury logowania do systemu.
powłoka logowania	Powłoka, która jest uruchamiana w czasie logowania do systemu.
limity dyskowe (<i>quotas</i>)	Przydział miejsca na dysku twardym, po przekroczeniu którego nie będzie możliwy zapis żadnych plików przez użytkownika.

Tab. 76. Cechy charakterystyczne konta użytkownika

Cechy kont użytkowników

Użytkownicy, grupy, numery UID i GID

Każdy użytkownik systemu UNIX posiada nazwę użytkownika (*username*), która go w sposób unikalny identyfikuje w systemie. Nazwy użytkowników są kojarzone z numerami UID (*User IDentification*) i w rzeczywistości to numer UID jest podstawą identyfikacji użytkowników. Jakkolwiek nazwa użytkownika jest łatwiejsza do zapamiętania i to ona stosowana jest powszechnie.

Nazwy użytkowników są zazwyczaj złożone ze znaków alfanumerycznych, to jest z liter i cyfr. Większość nazw użytkowników to formy imion użytkowników lub nazwa odnosząca się do pełnionej funkcji, lub też pseudo-użytkownik utworzony przez administracyjną część systemu (np. *root* – administrator systemu).

Grupy są stosowane do oznaczania logicznych grup użytkowników systemu. Grupy cechuje numer GID (*Group IDentification*), tak jak nazwy użytkowników cechują numery UID. Grupy mogą być wskazywane zarówno przez nazwę grupy, jak i numer GID. Typowym przykładem grupy może być grupa *webadmin*, grupująca użytkowników zarządzających stronami web. Pozwala to m.in. użytkownikom na tworzenie plików i katalogów, które będą mogły być czytane oraz zapisywane przez członków tej grupy.

Użytkownicy mogą zazwyczaj określać zarówno poprzez nazwę użytkownika, jak i przez numer UID, natomiast grupy mogą być określane przez nazwę grupy, a także przez numer GID.

Niektóre identyfikatory UID i GID są zarezerwowane do używania przez różne pakiety. Dzieje się tak, ponieważ niektóre pakiety wymagają dołączenia plików, które posiadają swoich użytkowników i grupy lub potrzebują identyfikatorów wkompiowanych w pliki binarne. Identyfikatory te używane są przez system tylko w celu, do jakiego są przeznaczone.

Numery UID i GID są podzielone na klasy:

- **0-99**

Globalnie przydzielone, te same dla każdej dystrybucji Debiana. Przeznaczone są dla pakietów, które potrzebują statycznego przydziału pojedynczego identyfikatora.

- **100-999**

Dynamicznie przydzielane dla użytkowników i grup systemowych. Przeznaczone są dla pakietów, które potrzebują użytkownika lub grupy, ale mogą mieć je przydzielane dynamicznie i inaczej w każdym systemie.

- **1000-29999**

Dynamicznie przydzielane dla kont użytkowników.

- **30000-59999**

Zarezerwowane.

- **60000-64999**

Globalnie przydzielone, ale tworzone tylko na żądanie. Identyfikatory są przydzielane centralnie i statycznie, ale rzeczywiste konta są tworzone tylko przez użytkowników systemowych na żądanie. Przeznaczone są dla mało znanych pakietów lub takich, które wymagają wielu statycznie przydzielonych numerów.

- **65000-65533**

Zarezerwowane.

- **65534**

Dla UID – użytkownik *nobody*, dla GID – grupa *nogroup*, czyli użytkownik lub grupa bez praw.

Hasła

Każdy użytkownik systemu UNIX posiada hasło, którego musi użyć w celu zalogowania do systemu. Hasło to jest początkowo określone w czasie tworzenia konta i może być zmienione w dowolnym czasie przez użytkownika lub przez administratora systemu.

Hasła w systemie są przechowywane w zaszyfrowanej formie. Rodzaj użytego kodowania oznacza się jako jedno-drogowe lub „zapadniowe” kodowanie, co oznacza że nie jest możliwe znalezienia oryginalnego hasła przez odwrócenie sposobu kodowania. Zamiast tego, hasło wpisywane przez użytkownika podczas logowania jest podobnie kodowane i dwie wersje zakodowanego hasła są porównywane w celu sprawdzenia, czy są takie same.

Hasła są zazwyczaj złożone ze znaków alfanumerycznych oraz znaków interpunkcyjnych. Uważa się za dobrą praktykę wybierania haseł, które zawierają mieszaninę wielkich i małych liter, cyfr i znaków interpunkcyjnych, jako że czyni te hasła trudniejszymi do odgadnięcia i mniej podatnymi na ataki „słownikowe”, za pomocą których każdy wyraz w słowniku jest kodowany i ta wersja jest porównywana z zakodowanym hasłem w systemie.

Przesłaniane hasła (*shadow passwords*) są używane w celu ukrycia każdej zakodowanej formy haseł użytkowników. Standardowy plik z hasłami (*/etc/passwd*) może odczytać każdy użytkownik, ale przesłaniany plik z hasłami (*/etc/shadow*) może odczytać tylko administrator.

Limity dyskowe

Limity dyskowe po raz pierwszy były dostępne w czasach systemu UNIX w komputerach o dzielonym dostępie. Limity zmuszają użytkowników do ograniczenia swoich zasobów i co za tym idzie, do odebrania im zdolności do zabierania nieskończonej ilości pamięci dyskowej. Limity wprowadzają ważne zabezpieczenie dla aplikacji. Jeżeli są ustawione, proces użytkownika, który wymknął się spod kontroli, nie zajmie całej dostępnej przestrzeni dyskowej.

Powłoki użytkowników

Powłoka użytkownika (*shell*) jest prostym makroprocesorem, który wykonuje polecenia. Shell jest pośrednikiem między użytkownikiem i jądrem systemu. Zadaniem powłoki jest umożliwienie wprowadzania poleceń z klawiatury, wykonywanie pewnych dodatkowych funkcji (jak np. podstawianie wartości pod zmienne, obsługa strumieni i potoków czy generowanie nazw plików na podstawie metaznaków), a wreszcie przekazywanie zinterpretowanych poleceń jądra systemu do wykonania.

Shell jest uruchamiany po udanym zalogowaniu się użytkownika do systemu i jest wykorzystywany jako główna metoda relacji pomiędzy użytkownikiem i jądrem systemu aż do wylogowania się użytkownika. Każdy użytkownik w systemie posiada domyślną powłokę.

W Linuksie jest dostępnych kilka różnych rodzajów powłok. Najbardziej powszechnymi są:

- **Bourne shell (*sh*)**

Jest oryginalną UNIXową powłoką i jest dostępny na każdym systemie UNIXowym.

- **Bourne Again shell (*bash*)**

Jest rozszerzeniem *sh* i w pełni z nią kompatybilna wstecz. Zawiera jednak wiele ulepszeń i dodatkowych cech, które nie są obecne w *sh*.

- **C shell (*csh*)**

Zawiera cechy takie jak kompletną linię komend. Powłoka ta uważana jest przez wielu nie tak dobra jak *bash*, ale jest używana przez programujących w C, ponieważ składnia tej powłoki jest podobna do tej, jaka jest używana w języku C.

- **Tenex C shell (*tcs*)**

Jest rozszerzoną i poprawioną wersją powłoki *csh*. Wiele spośród rozszerzeń wniesionych przez *bash* do *sh* zostało również dodanych do *tcs*, choć mają inną składnię.

- **Korn shell (*ksh*)**

Zawiera najlepsze cechy *sh* i *csh* i jest ich kombinacją w jednej pełnej powłoce kompatybilnej z *sh*. Powłoka ta jest wydajna i zawiera zarówno dobry interfejs użytkownika i dobry interfejs programistyczny.

Pliki konfiguracyjne

Plik */etc/passwd*

Plik */etc/passwd* jest wykorzystywany do konfiguracji kont użytkowników. Jest to plik tekstowy zawierający poprawne nazwy użytkowników oraz związane z nimi informacje. Każdy wiersz tego pliku podzielony jest na 7 pól oddzielonych od siebie dwukropkami. Format tego zapisu jest następujący:

```
<nazwa>:<zaszyfrowane hasło>:<UID>:<GID>:<komentarz>:<katalog  
macierzysty>:<powłoka logowania>
```

Znak *x* obok nazwy użytkownika wskazuje, że system wykorzystuje mechanizm haseł typu *shadow*.

Plik */etc/group*

Plik */etc/group* zawiera aktualną listę grup zdefiniowanych w systemie oraz użytkowników, którzy należą do każdej z tych grup. Każdy użytkownik jest przypisany do co najmniej jednej grupy. Wszystkie dodatkowe grupy, do których należy użytkownik, są wymienione w pliku */etc/group*. Format pliku */etc/group* jest następujący:

```
<nazwa grupy>:<zaszyfrowane hasło>:<GID>:<lista członków grupy>
```

Domyślna grupa nie musi posiadać zapisu w pliku */etc/group*. Wszystkie inne grupy, do których należą użytkownicy, muszą być dodane do pliku */etc/group*.

Plik `/etc/shadow`

Plik `/etc/shadow` zawiera zakodowane hasła, do którego dostęp ma tylko administrator systemu. Plik `/etc/shadow` zawiera pojedynczy wiersz dla każdego użytkownika. Ten wiersz obejmuje następujące informacje:

```
<nazwa>:<zaszyfrowane hasło>:<data ostatniej zmiany hasła>:<liczba dni, po upływie których można zmienić hasło>:<liczba dni, po upływie których trzeba zmienić hasło>:<liczba dni przed utratą ważności hasła, kiedy użytkownik otrzymuje ostrzeżenie>:<liczba dni po utracie ważności hasła, kiedy konto traci ważność>:<data zablokowania konta>
```

przy czym w polach `data ostatniej zmiany hasła` oraz `data zablokowania konta` wartość w pliku jest podawana w formacie liczby dni, które upłynęły od 1.01.1970 r.

Znak `*` w miejscu zaszyfrowanego hasła (po nazwie użytkownika) wskazuje, że użytkownik nie posiada hasła, a zatem nie może zalogować się do systemu.

Katalog `/etc/skel`

Dobrym zwyczajem jest stworzenie standardowego zestawu plików startowych, które są kopiowane do katalogu macierzystego nowego użytkownika. Można to zrobić w prosty sposób – poprzez utworzenie katalogu wzorcowego. W czasie tworzenia nowego użytkownika, pliki zawarte w katalogu `/etc/skel` są kopiowane do jego katalogu macierzystego. Pliki te mogą być później dowolnie modyfikowane przez użytkownika – w zależności od potrzeb.

Administrator systemu, tworząc nowe środowisko dla użytkowników, ma prawo dodawania, usuwania i modyfikowania plików zawartych w katalogu `/etc/skel`. Aby w maksymalnym stopniu ochronić domyślne środowisko dla nowych użytkowników, zaleca się, żeby zawartość katalogu `/etc/skel` była modyfikowana wyłącznie przez administratora systemu.

Programy zarządzania użytkownikami

Istnieją dwa podstawowe zadania zarządzania użytkownikami: tworzenie użytkowników i ich usuwanie. W systemie Debian istnieją narzędzia do ich łatwego dodawania oraz usuwania.

Program `useradd`

Aby utworzyć konto dla nowego użytkownika, należy wykonać następujące czynności:

- dodanie użytkownika do pliku `/etc/passwd`
- ustawienie domyślnego hasła dla nowego użytkownika
- dodanie użytkownika do odpowiednich grup
- utworzenie katalogu macierzystego dla nowego użytkownika
- utworzenie plików pocztowych dla nowego użytkownika
- utworzenie plików startowych potrzebnych użytkownikowi.

Ręczne tworzenie użytkownika, chociaż nie jest skomplikowane, jest dosyć pracochłonne. System Debian zawiera jednak program *useradd*, który automatycznie tworzy konto dla nowego użytkownika.

Opcja	Znaczenie
-c	Dodaje podany komentarz (zazwyczaj pełna nazwa użytkownika) w pliku <i>/etc/passwd</i>
-d	Określa katalog domowy użytkownika
-e	Określa datę w formacie MM/DD/YY, od której konto użytkownika zostanie wyłączone
-f	Określa liczbę dni po wygaśnięciu hasła do stałego wyłączenia konta
-g	Określa nazwę lub numer początkowej grupy logowania użytkownika
-G	Określa listę dodatkowych grup, do których należy będzie użytkownik
-m	Powoduje utworzenie domowego katalogu użytkownika
-s	Określa nazwę powłoki użytkownika
-u	Określa numeryczną wartość identyfikatora użytkownika
-D	Powoduje wyświetlenie bieżących wartości polecenia <i>useradd</i> oraz możliwość ich zmiany

Tab. 77. Opcje polecenia *useradd*

Po dodaniu użytkownika, należy jeszcze ustalić dla niego hasło. Służy do tego program *passwd*, w którym jako parametr należy podać nazwę istniejącego użytkownika.

Program *userdel*

Aby usunąć użytkownika z systemu, należy wykonać następujące czynności:

- usunięcie użytkownika z pliku */etc/passwd*
- usunięcie użytkownika z pliku */etc/group*
- usunięcie katalogu macierzysty użytkownika
- usunięcie plików poczty użytkownika.

System Debian jest wyposażony w program, za pomocą którego można w łatwy sposób usunąć użytkownika z systemu. Tym programem jest *userdel*. Ma on prostą składnię:

```
userdel [-r] <użytkownik>
```

gdzie opcjonalna opcja *-r* zastosowana w poleceniu informuje program, że należy usunąć katalog macierzysty użytkownika.

Program *usermod*

Program *usermod* zmienia systemowe pliki kont, odzwierciedlając zmiany podane w wierszu poleceń. Program *usermod* posiada w większości te same opcje, co program *useradd* oraz 2 dodatkowe:

- *-L* – blokuje hasło użytkownika poprzez wstawienie znaku przed zakodowanym hasłem w pliku */etc/shadow*, na skutek czego zostaje ono wyłączone
- *-U* – odblokowuje hasło użytkownika poprzez usunięcie znaku przed zakodowanego hasła w pliku */etc/shadow*.

Limity dyskowe

Obsługa limitów występuje w każdej partycji typu *ext2*. Może być ona dodana lub usunięta w każdej chwili, ponieważ nie wymaga zaimplementowania specjalnych zmian w systemie plików. Limity mogą być stosowane dla pojedynczych użytkowników oraz dla grup. Jeśli wykorzystywane są obydwa sposoby jednocześnie, zawsze obowiązuje najbardziej restrykcyjne ograniczenie. Zatem użytkownik nie może przekroczyć limitu jako członek grupy, mimo że ma jeszcze miejsce w swoim osobistym limicie.

Konfiguracja limitów

Aby dokonać ustawień dotyczących limitów, najpierw należy dokonać edycji pliku */etc/fstab* i w polu *options* każdego systemu plików, dla którego ma być włączona obsługa limitów, dodać flagi *usrquota* lub *grpquota*. Te opcje działają jako flagi dla narzędzi obsługujących limity i są ignorowane przez sterowniki systemów plików, dlatego ponowne montowanie systemów plików nie jest wymagane.

Po włączeniu obsługi limitów dla określonych systemów plików, utworzenie wstępnych plików limitów i włączenie obsługi limitów można wykonać za pomocą polecenia:

```
/etc/init.d/quota start
```

To polecenie tworzy pliki limitów dla każdego zamontowanego urządzenia, włącza limity i, jeżeli jest to konieczne, dla każdego systemu plików eksportowanego przez NFS włącza serwer limitów. Domyślnie nie są ustawiane żadne limity dla użytkowników oraz grup. Po zakończeniu tego procesu każdy system plików z włączoną obsługą limitów posiada do dwóch plików w jego głównym katalogu: *quota.user* zawierający informacje o limitach dla użytkowników oraz *quota.group*, który zawiera informacje o limitach dla grup. To polecenie zwykle jest uruchamiane w czasie startu systemu.

Po utworzeniu wstępnych plików, można dokonywać edycji domyślnych limitów za pomocą polecenia *edquota*:

```
edquota -u|-g <użytkownik lub grupa> <nazwa>
```

Polecenie *edquota* odczytuje bieżące ustawienia limitów (oraz bieżącą zajętość dysku) dla każdego użytkownika lub grupy przekazanego jako argument i tworzy niewielki raport w formacie ASCII na temat ustawień limitów oraz ich wykorzystania. Następnie raport ten jest otwierany za pomocą domyślnego edytora (określonego zmiennymi środowiskowymi *EDITOR* lub *VISUAL*). Należy dokonać edycji ustawień w pliku, zapisać je i zakończyć pracę edytora. Następnie polecenie *edquota* czyta plik w poszukiwaniu zmian w ustawieniach limitów i odpowiednio uaktualnia rekord dotyczący limitów.

Istnieje możliwość ograniczenia liczby bloków wykorzystywanych przez użytkownika (każdy blok to 1024 bajty) lub liczby i-węzłów. Limit *soft* wskazuje poziom, od którego system rozpocznie wyświetlanie komunikatów ostrzegawczych, że użytkownik

przekracza limity, natomiast limity **hard** określają poziom, od którego jądro uniemożliwi zajmowanie kolejnych bloków przez użytkownika.

Kiedy użytkownik przekroczy limit **soft**, system przydziela użytkownikowi czas, na zmniejszenie zajętości dysku poniżej limitu. Kiedy ten czas minie, limit **soft** jest traktowany jako limit **hard** i użytkownik nie może zajmować dodatkowej przestrzeni do czasu, gdy ograniczy rozmiar swoich plików poniżej limitu. Ten okres może być różny dla każdego systemu plików i może być zmieniony za pomocą polecenia *edquota -t*.

Zestaw narzędzi dotyczący limitów zawiera także narzędzia do wykonywania raportów. Polecenie *quota* wyświetli raport na temat limitów użytkownika lub grupy. Wywołuje się je w następujący sposób:

```
quota [-g] [-qv] [nazwa[nazwa...]]
```

Domyślnie, polecenie *quota* wyświetla raport dotyczący bieżącego użytkownika. Flaga **-g** informuje polecenie *quota* o wyświetleniu raportu na temat limitów grup. Domyślnie będzie wyświetlony raport o wszystkich grupach, których członkiem jest bieżący użytkownik. Flaga **-q** spowoduje wyświetlenie raportu tylko o tych systemach plików, w których użytkownik przekroczył limit, natomiast flaga **-v** wyświetla raport na temat użytkowników, którzy nie korzystają z danego systemu plików. Jeżeli użytkownik przekroczył limit, polecenie zwraca niezerowy status.

KONTROLOWANIE SYSTEMU

Pliki dzienników systemowych są bardzo ważnym elementem każdego systemu. Pozwalają na monitorowanie jego pracy i wykrywanie anomalii. Pozwalają także na zbieranie statystyk takich jak np. czas największej zajętości systemu lub liczba dostępów do systemu przypadająca na jednego użytkownika.

Rozliczanie (*accounting*) jest procesem, dzięki któremu system obserwuje wykorzystanie danego zasobu. Na przykład rozliczanie procesów pozwala na przechowywanie informacji o tym, przez jaki czas określony proces wykorzystywał procesor i pamięć. Z kolei rozliczanie pracy w sieci pozwala na śledzenie wykorzystania pasma. Dzięki temu można obserwować procesy, nad którymi straciło się kontrolę, bez wpływu na wydajność systemu.

W systemie Linux można zarejestrować praktycznie wszystkie zdarzenia. Pliki dzienników są domyślnie umieszczane w katalogu */var/log* i może je odczytywać tylko administrator.

Funkcje rejestrowania w systemie Linux opiera się na dwóch demonach: *syslogd* oraz *klogd*. Główne zadania systemu dzienników spoczywają na demonie *syslogd*. Otrzymuje on komunikaty z różnych procesów i rozprowadza je zgodnie z plikiem konfiguracyjnym. Z kolei *klogd* jest demonem dzienników jądra. Jego funkcja to zbieranie komunikatów generowanych przez jądro i przekazywanie ich do demona *syslogd*.

Demon syslogd

Większość funkcji rejestrowania w systemie Linux jest wykonywana przez demona *syslogd*. Jest on zazwyczaj uruchamiany w czasie ładowania systemu. Jego działanie polega na monitorowaniu żądań rejestrowania od programów oraz innych komputerów i wysyłanie ich do różnych plików docelowych, zgodnie ze źródłem oraz stopniem ważności. W niektórych programach można dokonać wyboru pomiędzy stosowaniem plików dzienników programu lub korzystaniem z demona *syslogd*. W takich przypadkach zwykle lepiej jest stosować *syslogd*, ponieważ zapewnia on jednolity interfejs konfiguracji rejestrowania.

Demon *syslogd* jest wszechstronny. Może on być skonfigurowany do wykonywania centralnego rejestrowania, tak by komunikaty z całej sieci trafiały do pojedynczego hosta, a także do przetwarzania komunikatów na różne sposoby. Może na przykład wysyłać niektóre komunikaty na konsolę, inne do pliku, jeszcze inne na terminale konkretnych użytkowników w czasie, kiedy są zalogowani, lub też do wszystkich tych miejsc.

Demon *syslogd* jest uruchamiany automatycznie podczas startu systemu przez skrypt */etc/init.d/sysklogd*. Demon *syslogd* może być uruchomiony z kilkoma parametrami. Aby zastosować te parametry, należy dokonać edycji pliku */etc/init.d/sysklogd* i dodać je do wiersza wywołującego demona *syslogd*.

Opcja	Działanie
-d	Włączenie trybu śledzenia. Opcję stosuje się tylko w czasie śledzenia programu <i>syslogd</i> .
-f <plik>	Wykorzystanie podanego pliku jako pliku konfiguracyjnego zamiast <i>/etc/syslog.conf</i> .
-h	Włączenie przekazywania przez <i>syslogd</i> wszystkich otrzymywanych komunikatów zależnie od przekazujących hostów określonych w pliku konfiguracyjnym. Domyślnie <i>syslogd</i> nie przekazuje żadnych komunikatów, które otrzymuje od innych hostów.
-l <lista hostów>	Dziennikowanie wymienionych hostów (oddzielonych dwukropkami) poprzez samą nazwę hosta z pominięciem domeny.
-m <liczba sekund>	Określenie przedziału czasowego (w sekundach) pomiędzy komunikatami znaczników. Komunikaty znaczników są wysyłane okresowo przez <i>syslogd</i> do funkcji <i>mark</i> . Domyślną wartością jest 20 sekund.
-n	Pozostawienie <i>syslogd</i> na pierwszym planie. Zazwyczaj w czasie uruchamiania <i>syslogd</i> przemieszcza się na drugi plan (tak jak wiele innych demonów). Jest to ważne na przykład, gdy <i>syslogd</i> jest uruchamiany bezpośrednio z pliku <i>etc/inittab</i> .
-p <gniazdo>	Ustawienie gniazda do wydobywania komunikatów od lokalnych procesów. Domyślnie <i>syslogd</i> monitoruje gniazdo <i>/dev/log</i> .
-r	Włączenie monitorowania sieci w gniazdach wymienionych jako <i>syslog</i> w pliku <i>/etc/services</i> (zwykle UDP port 514). Domyślnie <i>syslogd</i> nie monitoruje sieci.
-s <lista domen>	Odrzucanie podanych domen (oddzielonych dwukropkami) z nazwy hosta przed zapisaniem komunikatu. Zastosowana będzie pierwsza zgodna domena.
-v	Wyświetla wersję demona <i>syslogd</i> i kończy działanie.

Tab. 78. Opcje demona *syslogd*

Konfiguracja demona *syslogd*

Konfigurację demona wykonuje się przez edycję pliku */etc/syslog.conf*. Każdy wiersz pliku */etc/syslog.conf* składa się z dwóch części: selektora oraz akcji oddzielonych od siebie spacją.

Selektor wskazuje, do których komunikatów ma być zastosowane działanie. Akcja określa, co należy zrobić z tymi komunikatami.

Komunikaty są wybierane według dwóch parametrów. Pierwszy z nich to funkcja. Wskazuje on program lub usługę, skąd pochodzi komunikat. Ponieważ liczba nazw funkcji jest ograniczona, niektóre programy są zmuszone do korzystania z nazw funkcji, które nie odpowiadają ich rzeczywistym nazwom.

Nazwa funkcji	Znaczenie
auth	Komunikaty związane z autoryzacją
auth-priv	Inne komunikaty związane z autoryzacją
cron	Komunikaty <i>cron</i> d
daemon	Komunikaty pozostałych demonów
kern	Komunikaty jądra systemu
local[0-7]	Komunikaty lokalne
lpr	Komunikaty systemu obsługi drukarki
mail	Komunikaty systemu obsługującego pocztę
mark	Znaczniki czasowe wysyłane w regularnych odstępach
news	Komunikaty systemu wiadomości
security	Starszy odpowiednik funkcji auth
syslog	Komunikaty demona <i>syslogd</i>
user	Komunikaty od procesów użytkowników
uucp	Komunikaty protokołu <i>uucp</i>

Tab. 79. Nazwy funkcji selektora i ich znaczenie

Drugim parametrem wybierania komunikatów jest priorytet. Jest to wskazanie ważności komunikatu.

Priorytet	Znaczenie
debug	Komunikaty śledzenia
info	Komunikaty informacyjne
notice	Komunikaty w warunkach normalnych, ale system wymaga uwagi
warning (warn)	Komunikaty ostrzeżeń
err (error)	Komunikaty o błędach
crit	Komunikaty o błędach krytycznych
alert	Komunikaty, przy których należy natychmiast podjąć działania
emerg (panic)	Komunikaty o niemożności pracy systemu

Tab. 80. Priorytety selektora i ich znaczenie

Selektor ma formę <funkcja>.<priorytet>. Ta forma jest zgodna z wszystkimi komunikatami wybranego priorytetu i wyższymi. Jeżeli forma ma się odnosić wyłącznie do określonego priorytetu, można zastosować znak równości <funkcja>.<priorytet>. Gwiazdka oznacza wszystko, zatem <funkcja>.* oznacza wszystkie komunikaty <funkcja> (co jest równoważne z <funkcja>.debug), natomiast *.<priorytet> jest zgodne z wszystkimi komunikatami o podanym priorytecie i wyższymi. Wykrzyknik oznacza negację warunku. Zatem [funkcja].![priorytet] oznacza wszystkie komunikaty niższe od podanego.

Można połączyć kilka funkcji w jednym selektorze, stosując przecinki między funkcjami. Można też połączyć kilka selektorów za pomocą średnika. W tym przypadku można wykorzystać specjalny priorytet `none` do określenia braku komunikatów z danej usługi. Na przykład `*.<priorytet>;<funkcja>.none` odpowiada wszystkim komunikatom na poziomie `<priorytet>` lub wyższym z wyjątkiem tych pochodzących z funkcji `<funkcja>`.

Pole akcji określa działanie, jakie ma być wykonane z komunikatami zgodnymi z danym selektorem. Pierwszy znak akcji określa jej typ.

Akcja, która zaczyna się od znaku ukośnika określa nazwę pliku, do którego będą dodawane komunikaty (np. `/var/log/messages`). Jeżeli plik jest urządzeniem `tty`, będzie zastosowana specjalna obsługa `tty`. W celu skierowania komunikatów na konsolę, należy zastosować `/dev/console`.

Znak minus poprzedzający znak ukośnika również określa plik, ale wskazuje, że zapis do pliku nie będzie wykonywany po każdej aktualizacji. Dzięki temu demon `syslogd` jest bardziej wydajny. Jednakże istnieje ryzyko utraty niektórych komunikatów w przypadku awarii systemu. Tej akcji nie powinno się stosować z ważnymi dziennikami.

Znak `@` określa nazwę hosta, do którego mają być wysyłane komunikaty. Znak `|` określa nazwę potoku (`fifo`), gdzie mają być kierowane komunikaty. Potok powinien zostać utworzony za pomocą polecenia `mkfifo` przed uruchomieniem demona `syslogd`. Dowolne znaki alfanumeryczne określają początek listy użytkowników oddzielanych przecinkami. Jeżeli są oni zalogowani, komunikaty będą wysyłane na ich terminale. Gwiazdka wskazuje, że komunikat będzie wyświetlony na terminalach wszystkich zalogowanych użytkowników.

Aby stworzyć pojedynczy host rejestrowania w sieci, należy do wszystkich plików konfiguracyjnych demona `syslogd` we wszystkich innych hostach dopisać linię:

```
*.* @<logger>
```

Natomiast w komputerze `<logger>` zbierającym dzienniki od innych hostów, należy dokonać edycji pliku `/etc/init.d/syslogd`:

```
SYSLOGD="-r"
```

Po dokonaniu edycji należy ponownie uruchomić demona `syslogd`.

```
# Przykładowy plik /etc/syslog.conf

auth,authpriv.*          /var/log/auth.log
daemon.*                 -/var/log/daemon.log
kern.info                 -/var/log/kern.log
mail.*                   -/var/log/mail/mail.log
user.*                   -/var/log/user.log
*.*;auth,authpriv,kern.none;\
    daemon,mail,user.none  /var/log/otherlog
#Standardowe pliki dzienników.

*.emerg                  *
#Powoduje wysłanie komunikatów o najwyższym priorytecie na ekrany
#terminali wszystkich zalogowanych użytkowników.
```

Demon klogd

W systemie Linux komunikaty jądra są tradycyjnie kierowane na konsolę. Jednakże w wielu przypadkach nie jest to pożądane. Demon *klogd* jest tak zaprojektowany, by przechwytywał wszystkie komunikaty jądra i przekazywał je do demona *syslogd*. Demon *syslogd* może wykonywać z nimi bardziej skomplikowane operacje.

Jeżeli jest zamontowany system plików */proc*, jądro udostępnia swoje komunikaty poprzez pseudo-plik */proc/kmsg*. Demon *klogd* czyta te komunikaty za pomocą odwołań systemowych.

Demon *klogd* rejestruje komunikaty za pomocą funkcji *kern*. Priorytet jest przydzielany przez samo jądro. Demon *klogd* dokonuje konwersji tych priorytetów na określony priorytet demona *syslogd*. Priorytety jądro są wymienione w pliku */usr/include/linux/kernel.h*.

Demony *syslogd* i *klogd* pozwalają na zarządzanie sposobem zapisu komunikatów generowanych przez procesy oraz jądro systemu Linux do dzienników systemowych. Pliki dzienników są przydatne, gdy zachodzi potrzeba sprawdzenia, jakie zdarzenia miały miejsca w systemie.

Administracja dziennikami

Jeżeli dzienniki systemowe będą pozostawione bez nadzoru, będą one rosły, aż w końcu zapelniaą przypisaną partycję dysku. Dlatego dobrym pomysłem jest przypisanie oddzielnej partycji do katalogu */var*. W przeciwnym wypadku dzienniki mogą rozrosnąć się do zbyt dużych rozmiarów i wypełnią partycję *root*, co sprawi, że praca w systemie będzie praktycznie niemożliwa.

Aby zapobiec niekontrolowanemu rozrastaniu się plików dzienników, system Debian zawiera narzędzie o nazwie *savelog*. Zadaniem tego narzędzia jest zarządzanie rotacją dzienników systemowych. Oznacza to okresową obsługę przesuwania plików dzienników i śledzenie konfigurowanej liczby plików archiwalnych.

Za każdym razem, kiedy działa program *savelog*, zmienia on nazwę bieżącego pliku dziennika na tę samą nazwę z przyrostkiem *.0*. Jeżeli taki już istnieje, zmienia mu przyrostek na *.1* itd. Kiedy liczba archiwalnych dzienników systemowych osiągnie liczbę określoną w konfiguracji, program *savelog* może też skompresować te pliki, żeby zajmowały mniej miejsca na dysku. Nigdy nie jest kompresowany plik z przyrostkiem *.0*, ponieważ mogą być do niego zapisywane informacje w trakcie trwania rotacji.

Konfiguracja programu savelog

Program *savelog* jest zwykle uruchamiany okresowo przez narzędzie *anacron*. Jest on uruchamiany raz dla każdego rotowanego pliku dzienników. Tak więc każda aplikacja wysyłająca komunikaty zwykle ma swój własny skrypt w pliku */etc/cron.daily* lub */etc/cron.weekly*. Można dokonać edycji tych plików w celu wykonania jakiejś czynności przed lub po rotacji dzienników.

Opcja	Działanie
-m <atrybut>	Atrybut pliku archiwalnego jest zmieniany na podany (ósemkowo)
-u <użytkownik>	Właściciel pliku archiwalnego jest zmieniany na podanego
-g <grupa>	Grupa pliku archiwalnego jest zmieniana na podaną
-c <cykl>	Największa liczba przyrostka, do którego przechowywane są pliki archiwalne
-t	Tworzy pusty plik dziennika po rotacji
-l	Nie kompresuje plików archiwalnych
-p	Ochronia atrybut, właściciela oraz grupę pliku dzienników

Tab. 81. Opcje programu *save*log

Analiza wydajności systemu

W rozliczeniach dotyczących procesów, należy wziąć pod uwagę przede wszystkim ilość zasobów systemu (pamięci oraz procesora) wykorzystywanych przez proces. Zestawienia dotyczące procesów są przydatne w przypadku analizowania wydajności serwera oraz w czasie wykonywania planowania możliwości systemu. System Linux zawiera kilka narzędzi do wykonania rozliczeń dotyczących procesów.

Polecenie *ps*

Polecenie *ps* powoduje wyświetlenie listy uruchomionych procesów w systemie w danym momencie.

Opcja	Działanie
l	Wyświetla proces w postaci długiego formatu
j	Wyświetla proces w postaci formatu prac
s	Wyświetla proces w postaci formatu sygnału
u	Wyświetla format użytkownika (nazwa użytkownika i czas startu)
m	Wyświetla informacje o pamięci
f	Wyświetla informacje w formacie drzewiastym
a	Wyświetla procesy należące do innych użytkowników
x	Wyświetla procesy, które nie są sterowane przez terminal (demony)
w	Wyświetla procesy w następnej linii, jeśli nie mieszczą się w jednej
r	Wyświetla tylko pracujące procesy
--sort=[-]<klucz>	Wyświetla procesy według wieloliterowego klucza, natomiast opcjonalny znak [-] oznacza zmniejszanie numerycznego lub leksykograficznego porządku
--version	Wyświetla wersję i źródło programu
--help	Wyświetla wszystkie opcje i ich krótki opis

Tab. 82. Ważniejsze opcje polecenia *ps*

Klucz	Opis
c (cmd)	Sama nazwa programu wykonywalnego
C (cmdline)	Pełna linia komend
g (pgrp)	ID grupy procesu
G (tpgid)	ID grupy procesu kontrolującego <i>tty</i>
o (session)	ID sesji
p (pid)	ID procesu
P (ppid)	ID procesu rodzicielskiego
t (tty)	Numer urządzenia <i>tty</i> (terminala)
T (start_time)	Czas uruchomienia procesu
U (uid)	ID użytkownika
u (user)	Nazwa użytkownika
v (vsize)	Całkowity rozmiar pamięci wirtualnej w bajtach
y (priority)	Priorytet w kolejce procesów jądra

Tab. 83. Ważniejsze klucze sortowania polecenia *ps*

Pole	Opis
USER	Nazwa użytkownika, który jest właścicielem procesu
PID	Numer ID procesu
COMMAND	Polecenie wykonywane przez proces
SIZE	Całkowity, wirtualny rozmiar procesu w KB
RSS	Rozmiar pamięci RAM w KB aktualnie wykorzystywanej przez proces
SHARE	Rozmiar pamięci RAM w KB dzielonej z innymi procesami
TIME	Sumaryczny czas procesora wykorzystany przez proces w postaci [mm]:[ss]
STAT	<p>Informacje o statusie procesu:</p> <ul style="list-style-type: none"> R – działający (<i>runnable</i>) S – nieaktywny przez więcej niż 20 sekund (<i>sleeping</i>) I – nieaktywny przez mniej niż 20 sekund D – oczekujący na dostęp do dysku (<i>uninterruptible sleep</i>) T – śledzony lub zatrzymany (<i>traced or stopped</i>) Z – porzucony, który powinien zostać zamknięty (<i>zombie</i>) <p>Dla formatu BSD, mogą być wyświetlane dodatkowe informacje w postaci:</p> <ul style="list-style-type: none"> W – proces przeniesiony do pliku wymiany (<i>resident pages</i>) < – wysoki priorytet zadania N – niski priorytet zadania (dodatnia wartość <i>nice</i>) L – proces ma strony zamknięte w pamięci
TTY	Kontrolujący proces <i>tty</i> (terminal)
%CPU	Procentowy udział czasu procesora wykorzystywany przez proces
%MEM	Procentowy pamięci systemowej RAM wykorzystywany przez proces

Tab. 84. Ważniejsze pola polecenia *ps*

Polecenie top

Polecenie *top* jest interaktywną wersją polecenia *ps* pracującą w czasie rzeczywistym. Pokazuje ono te same podstawowe statystyki jak polecenie *ps*, uaktualniając wyświetlanie co 5 sekund. Pozwala ono na sortowanie wyników według %MEM, %CPU oraz TIME, a także pozwala na wyświetlanie tylko procesów określonego użytkownika albo procesów, które nie znajdują się w stanie oczekiwania. Pozwala także na niszczenie zadań.

```
00:03:42 up 24 min, 3 users, load average: 0.11, 0.03, 0.02
52 processes: 51 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 1.2% user, 1.6% system, 0.0% nice, 97.3% idle
Mem: 29192K total, 23188K used, 6004K free, 1032K buffers
Swap: 120956K total, 11684K used, 109272K free, 11064K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
569	maciek	11	0	2380	2380	1648	S	0.5	8.1	0:06	hitchx
564	root	9	0	2000	1964	1660	S	0.0	6.7	0:00	sshd
543	root	11	0	2000	1960	1668	S	0.3	6.7	0:03	sshd
576	root	9	0	1996	1960	1656	S	0.0	6.7	0:00	sshd
570	maciek	9	0	1292	1292	988	S	0.0	4.4	0:00	mc
637	user	9	0	1288	1288	988	S	0.0	4.4	0:00	mc
572	maciek	8	0	1244	1244	1020	S	0.0	4.2	0:00	bash
639	user	8	0	1240	1240	1020	S	0.0	4.2	0:00	bash
648	jakis	9	0	1228	1228	1012	S	0.0	4.2	0:00	bash
590	user	9	0	1216	1216	884	S	0.0	4.1	0:00	screen
651	root	9	0	1216	1216	984	S	0.0	4.1	0:00	bash
565	maciek	9	0	1208	1208	992	S	0.0	4.1	0:00	bash
558	dyplo	9	0	1204	1204	992	S	0.0	4.1	0:00	bash
577	user	9	0	1204	1204	992	S	0.0	4.1	0:00	bash
561	dyplo	9	0	1212	1180	880	S	0.0	4.0	0:00	screen
591	user	9	0	1124	1124	908	S	0.0	3.8	0:00	ncftp
562	dyplo	9	0	1120	1112	908	S	0.0	3.8	0:00	irc

Rys. 23. Wynik działania polecenia *top*

Polecenie vmstat

Polecenie *vmstat* okresowo próbuje stan systemu i wyświetla w wyniku jeden wiersz. Zawiera on średnią wartość statystyki obciążenia systemu podczas tego przedziału. Program *vmstat* jest zazwyczaj wywoływany z jednym argumentem określającym liczbę sekund pomiędzy próbkami. Można także podać drugi argument liczbowy, który będzie określał liczbę próbek, jakie należy wykonać.

Ważną rzeczą, jaką należy zauważyć w przypadku korzystania z *vmstat*, jest fakt, że powinno się zawsze ignorować pierwszy wiersz wyniku. Ten wiersz określa średnią od momentu załadowania systemu i zazwyczaj nie ma znaczenia. Użyteczne wartości zaczynają się od drugiego wiersza wyniku. Jako próbkę program *vmstat* generuje wiersz o długości 72 znaków.

Mimo, że *vmstat* zajmuje się statystykami dotyczącymi pamięci wirtualnej, jest on przydatny do monitorowania aktywności dysku i procesora komputera. Okresowe uruchamianie programu *vmstat*, pozwala na zapoznanie się z liczbami, jakie wyświetla w różnych warunkach obciążenia systemu. Polecenie *vmstat* można także wykorzystywać do wykrywania, czy problem wydajności wynika ze względu na problemy dysku czy obciążenie procesora.

Pole	Opis
procs: r	Liczba procesów czekająca na uruchomienie. Zwykle r nie powinno być większe niż liczba procesorów w systemie.
procs: w	Liczba procesów w stanie „uśpienia”, czekających na jakieś zdarzenie. Zwykle czekają one na zakończenie żądania wejścia-wyjścia.
procs: b	Liczba innych procesów czekających na uruchomienie znajdujących się w pamięci wirtualnej.
mem: swpd, free	Ilość pamięci wykorzystywanego obszaru wymiany oraz ilość wolnej pamięci RAM (w KB).
mem: buf, cache	Ilość wykorzystywanej pamięci, odpowiednio przez bufor dyskowy oraz jako dyskowa pamięć podręczna w KB. Ta pamięć jest zarządzana dynamicznie przez jądro systemu Linux, tak więc większość z tych wartości może być włączona do pamięci wolnej w przypadku braku pamięci RAM.
-swap: si, so	Liczba KB pamięci wymieniana do pamięci wirtualnej i z powrotem na sekundę. Jeżeli te wartości są stosunkowo wysokie, host potrzebuje więcej pamięci RAM.
-io: bi, bo	Liczba bloków na sekundę, odpowiednio otrzymanych z i wysłanych do urządzenia blokowego.
-system: in	Liczba otrzymanych przerw na sekundę, włącznie z przerwaniem zegara.
-system: cs	Liczba przełączeń zawartości na sekundę. Przełączenie zawartości ma miejsce, kiedy procesor przełącza się od jednego procesu do innego.
cpu: sy, us, id	Procentowy udział czasu procesora odpowiednio w systemie, w procesie użytkownika oraz w stanie oczekiwania. Jeżeli procentowy udział stanu oczekiwania jest ciągle wyższy niż 70%, procesor hosta jest przeciążony.

Tab. 85. Pola polecenia *vmstat*

Polecenie *lastcomm*

Polecenie *lastcomm* powoduje wyświetlenie informacji na temat wszystkich programów, które były uruchomione (i zakończyły swoje działanie) od założenia pliku */var/account/pacct*. Domyślnie ten plik jest rotowany codziennie poprzez plik */etc/cron.daily/acct*. Jeżeli mają być zebrane dane za kilka dni, można zastosować flagę *-f* w celu czytania starszych plików (po ich uprzedniej dekompresji) lub zmodyfikować skrypt */etc/cron.daily/acct*.

Polecenie *lastcomm* wyświetla dane w następujących polach:

- uruchomione polecenie
- flagi
- użytkownik, który uruchamiał polecenie
- terminal *tty*, z którego było uruchomione polecenie
- ilość czasu procesora wykorzystana przez proces
- czas zakończenia procesu.

Flaga	Opis
-S	Polecenie było wykonane przez superużytkownika
F	Program wykonał rozwidlenie (<i>fork</i>), czyli rozdzielenie na dwa procesy
D	Program zakończył działanie w nadzwyczajnych warunkach, generując plik <i>core</i> .
X	Program został zatrzymany przez sygnał SIGTERM.

Tab. 86. Flagi polecenia *lastcomm*

Do polecenia *lastcomm* można dodać parametry, które będą działały jak klucze wyszukiwania, wyświetlając tylko te rekordy, które są zgodne z dowolnymi parametrami w dowolnych polach.

Polecenie *sa*

Polecenie *sa* jest wykorzystywane do wykonywania podsumowań informacji zawartych w pliku */var/account/pacct*. Dodatkowo może ono zapisywać sumaryczne informacje w pliku */var/account/savacct* oraz */var/account/usracct*.

Pole	Opis
cp	Całkowity czas CPU, suma czasu systemowego i użytkownika
re	Całkowity rzeczywisty czas uruchomienia w sekundach
re/cp	Współczynnik pomiędzy czasem rzeczywistym a czasem CPU
u	Czas CPU użytkownika w sekundach
s	Systemowy czas CPU w sekundach

Tab. 87. Pola polecenia *sa*

Nazwy pól nie są wyświetlane w nagłówkach kolumn. Zamiast tego, są one dodane do każdej wartości. Polecenie *sa* może być wywołane z kilkoma opcjami.

Opcja	Działanie
-a	Wyświetla wszystkie polecenia.
-c	Wyświetla procentowy współczynnik całkowitego czasu dla każdej z wartości: użytkownik, system oraz czas rzeczywisty.
-l	Oddziela czas użytkownika od czasu systemowego. Normalnie, są one dodane i wyświetlone jako czas CPU.
-m	Wyświetla liczbę procesów oraz liczbę minut procesora dla każdego użytkownika.
-s	Podsumowuje plik <i>/var/account/pacct</i> do plików <i>savecat</i> oraz <i>usracct</i> . Wymaga uruchomienia przez użytkownika <i>root</i> .
-t	Wyświetla współczynnik czasu rzeczywistego w stosunku do całkowitego czasu procesora dla każdego procesu.
-u	Wyświetla tylko identyfikator ID użytkownika oraz nazwę polecenia dla każdego zapisu w pliku <i>pacct</i> . Jeśli określono opcję <i>-u</i> , wszystkie inne opcje są ignorowane.

Tab. 88. Opcje polecenia *sa*

Rozliczanie użytkowników

Podobnie jak rozliczanie procesów, rozliczanie użytkowników zajmuje się ilością zasobów systemowych, które są wykorzystywane przez proces. Celem rozliczania użytkowników jest śledzenie ilości zasobów systemowych zużytych przez każdego z użytkowników. W związku z tym to zagadnienie jest blisko związane z rozliczaniem procesów i niektóre programy wykorzystywane do rozliczania procesów mogą dostarczać danych do rozliczania użytkowników. Jednakże istnieje kilka narzędzi, których jedynym celem jest rozliczanie użytkowników, szczególnie tych, którzy logują się do serwera. Te polecenia to *last* oraz *ac*.

Polecenia *last* oraz *ac* wyświetlają dane zapisane w pliku */var/log/wtmp*. Dane do tego pliku są dostarczane przez procesy *login* oraz *init*. Skrypt */etc/cron.monthly/acct* powoduje jego rotację po podsumowaniu jego zawartości w pliku */var/log/wtmp.report*. Jeżeli plik */var/log/wtmp* nie istnieje, rejestrowanie nie jest możliwe.

Polecenie last

Polecenie *last* wyświetla rekordy z pliku */var/log/wtmp*. Każdy rekord składa się z następujących danych:

- nazwa użytkownika
- terminal *tty*, w którym miało miejsce logowanie
- host, z którego nastąpiło logowanie lub konsola, jeżeli było to logowanie lokalne
- etykieta czasowa logowania, a następnie czas wylogowania lub jeśli użytkownik nadal jest zalogowany – tekst **still logged**
- całkowity czas sesji podany w nawiasach w formie [gg]:[mm]

Specjalny użytkownik **reboot** oznacza czasy ładowania systemu.

Polecenie ac

Polecenie *ac* wyświetla listę całkowitego połączenia w godzinach użytkowników. Bez podanych parametrów *ac* wyświetla listę całkowitego czasu połączenia dla wszystkich użytkowników.

Opcja	Działanie
-d	Wyświetla podsumowania dzienne zamiast pojedynczych podsumowań
-p	Wyświetla podsumowania dla każdego użytkownika

Tab. 89. Najczęściej stosowane opcje polecenia *ac*

ODTWARZANIE PO AWARII

Jeżeli system Debian jest zainstalowany, skonfigurowany i przetestowany, trudno sobie wyobrazić jakiegokolwiek problemy. Jednak wszystkie systemy operacyjne muszą pracować na sprzęcie, a w sprzęcie mogą wystąpić błędy w wewnętrznym oprogramowaniu, może być on źle wykonany, przestarzały, źle wykorzystywany oraz mogą wystąpić problemy niezgodności.

Istnieją trzy czynności, które można wykonać w celu zabezpieczenia się przed awariami. Są to:

- wykonywanie kopii zapasowych
- przygotowanie dyskietki ratunkowej
- dokumentowanie konfiguracji systemu.

Kopie zapasowe

Najlepszym sposobem na utrzymanie bezpiecznego systemu jest regularne wykonywanie kopii zapasowych. Wewnętrzne mechanizmy wielozadaniowości oraz planowania zadań w systemie Debian, pozwalają na wykonywanie kopii zapasowych w regularnych odstępach czasu automatycznie lub z minimalną potrzebą ręcznych interwencji.

Decydowanie o tym, co należy zarchiwizować

Najważniejszą decyzją do podjęcia w czasie planowania strategii wykonywania kopii zapasowych jest decyzja o tym, co należy archiwizować.

Wykonywanie kopii wszystkiego (oprócz niebezpiecznych katalogów jak */proc*) ma swoje zalety. Jeżeli wykonana zostanie kopia zapasowa wszystkiego, można odtworzyć system z dyskietki ratunkowej bez konieczności ponownej instalacji. Jednak nośnik kopii zapasowej może nie być wystarczająco duży, aby zapisać wszystkie dane w systemie.

Z pewnością ważne z punktu widzenia późniejszej możliwości odtworzenia systemu są katalogi */etc*, */home* oraz */var*.

Wybór programu

Istnieje wiele programów do tworzenia kopii zapasowych. Tradycyjnymi programami do tego przeznaczonymi są: *tar*, *cpio*, *dump*. Wybór narzędzia może być podyktowany wybranym medium.

Programy *tar* i *cpio* są podobne. Oba programy potrafią zapisywać i odczytywać dane z taśm, do tego potrafią obsłużyć każde medium, z którym potrafi sobie poradzić jądro.

Program *dump* pracuje w sposób nieco odmienny niż powyższe programy – odczytuje pliki bezpośrednio z urządzenia, bez pośrednictwa systemu plików. Program został specjalnie napisany do tworzenia kopii zapasowych, *tar* i *cpio* powstały w celu archiwizowania danych.

Bezpośrednie odczytywanie systemu plików ma pewne zalety – umożliwia odczytywanie plików bez zmiany ich znaczników czasowych. Bezpośrednie odczytywanie systemu plików jest wydajniejsze przy kopiowaniu wszystkiego. Główną wadą takiego podejścia jest ograniczenie programu do jednego systemu plików.

Program *dump* ma wbudowaną obsługę poziomów kopiowania, natomiast *tar* i *cpio* wymagają do tego dodatkowych narzędzi.

Program tar

Program *tar* jest standardowym narzędziem wykonywania kopii zapasowych w systemie uniksowym. Narzędzie *tar* zostało wybrane jako standard tworzenia plików archiwów systemów uniksowych. W istocie polecenie *tar* czyta pliki z lokalnego systemu plików i zapisuje je do pliku archiwum, który może być urządzeniem.

Typowe polecenie *tar* ma następującą postać:

```
tar [<opcje>] [-f <plik archiwum>] [katalog]
```

Opcja	Znaczenie
f	Podaje położenie pliku archiwum
c	Utworzenie archiwum
d	Sprawdza różnice pomiędzy zawartością katalogu, a archiwum i wyświetla je na ekranie
t	Testuje archiwum i wyświetla jego zawartość
x	Odtwarza pliki z archiwum do określonego katalogu
z	Wykonuje kompresję za pomocą programu <i>gzip</i> bezpośrednio przed zapisem do archiwum
l	Nie przechodzi do systemów plików zamontowanych w podanym katalogu
M	Umożliwia wykonanie archiwum złożonego z wielu wolumenów
p	W czasie wydobywania plików zachowuje informacje o prawach dostępu oraz własności
v	Wyświetla dodatkowe informacje na temat bieżącego procesu, łącznie z nazwami plików
W	Weryfikuje archiwum po jego zapisaniu

Tab. 90. Ważniejsze opcje polecenia *tar*

Program cpio

Polecenie *cpio* jest podobne do polecenia *tar*. Działa tylko z plikami archiwum, kopiując pliki do tych plików oraz z tych plików. Jednakże, inaczej niż polecenie *tar*, polecenie *cpio* czyta listę plików, które mają być zapisane do archiwum ze standardowego urządzenia wejściowego. Z tego względu, jest ono nieco mniej wygodne w użyciu, ale pozwala na bardzo szczegółową kontrolę nad tym, jakie dokładnie pliki mają być uwzględnione w archiwum. Polecenie *cpio* może być także wykorzystane do wykonania szybkiej kopii, czytając listę plików do skopiowania ze standardowego urządzenia wejściowego i kopiując je do podanego katalogu.

Do wygenerowania listy plików dla polecenia *cpio* może być wykorzystane dowolne ze standardowych narzędzi. Nie trzeba korzystać z ręcznie tworzonego,

statycznego pliku. Dobrym narzędziem dla wykonania tego zadania jest polecenie *find*, które przegląda katalogi i generuje listę w zależności od określonej specyfikacji.

Polecenie *afio* jest ulepszoną wersją polecenia *cpio*, zaprojektowaną specjalnie dla wykonywania kopii zapasowych. W tej metodzie kompresja dotyczy każdego pliku przed jego zapisaniem do archiwum, a nie archiwum jako całości. Dzięki temu, jeżeli wystąpi błąd w odtwarzaniu skompresowanego archiwum, utracony zostanie tylko jeden plik. Stosując polecenie *tar* lub *cpio*, pojedynczy błąd odczytu może spowodować, że pozostała część kopii zapasowej, począwszy od tego punktu nie będzie mogła być odczytana. Polecenie *afio* ma także większe możliwości obsługi błędów odczytu w czasie odtwarzania.

Program dump

Polecenia *tar* oraz *cpio* w czasie wykonywania kopii zapasowej czytają każdy system plików w zwykły sposób. Natomiast polecenie *dump* działa bezpośrednio z systemem plików, pomijając jądro. Z tego względu kopie zapasowe oraz odtwarzanie mogą być wykonywane dla zamontowanych systemów plików oraz dla tych, które nie są zamontowane. Polecenie *dump* pobiera jako argument albo „surowy” system plików, albo katalog i wykonuje kopię zapasową tego systemu plików lub wszystkich plików w podanym katalogu. Ponieważ działa ono bezpośrednio z systemem plików, może wykonywać kopię zapasową tylko tych systemów plików, które są dla niego znane.

Za pomocą polecenia *dump* można wykonywać przyrostowe oraz różnicowe kopie zapasowe, wykorzystując system poziomów zrzutów. Zrzuty są uruchamiane na określonych poziomach. Pliki są archiwizowane, jeżeli były modyfikowane od ostatniego zrzutu na poziomie tym samym lub niższym. Zgodnie z tym schematem, zrzut na poziomie 0 gwarantuje wykonanie całości kopii zapasowej, natomiast poziom 1 powoduje zrzut wszystkich plików od wykonania ostatniego zrzutu na poziomie 0 lub 1. Dowolne pliki zarchiwizowane na poziomie 2 będą ponownie zarchiwizowane w czasie wykonywania zrzutu na poziomie 1.

Ponieważ polecenie *dump* działa bezpośrednio z systemem plików, a nie poprzez jądro, można tak skonfigurować system, żeby użytkownik mógł wykonywać kopię zapasową plików, do których nie ma dostępu. W systemie Debian można to uzyskać poprzez dodanie użytkownika do grupy *disk*. W takim przypadku użytkownik będzie miał możliwość bezpośredniego wykonywania zrzutu systemu plików. Tę właściwość należy wykorzystywać ostrożnie. Użytkownik ma pełny dostęp do zapisu i odczytu na całym dysku. Pozwala to mu zastępowanie części systemu plików, a nawet wykonanie formatowania.

Polecenie *dump* może dodatkowo analizować plik */etc/fstab* w celu określenia, który system plików ma być zarchiwizowany. Piąte pole pliku */etc/fstab* (plik, w którym określa się, które systemy plików powinny być zamontowane w czasie ładowania systemu) zawiera liczbę. Liczba różna od zera informuje polecenie *dump*, że system plików jest przewidziany do wykonywania zrzutów poleceniem *dump*. Pozwala to poleceniu *dump* na

wyświetlanie informacji na temat tego, który system plików powinien być zarchiwizowany oraz kiedy ostatni raz był wykonywany rzut.

Jeżeli polecenie *dump* zostanie wywołane interaktywnie, na terminalu będzie wyświetlona informacja dotycząca bieżącego stanu. Jeżeli wystąpi problem, będzie wyświetlony komunikat diagnostyczny i oczekiwanie na odpowiedź operatora na pytanie. Można także tak skonfigurować polecenie, że w przypadku wystąpienia problemu, będą alarmowani wszyscy użytkownicy z danej grupy.

Proces odtwarzania kopii zapasowej wykonanej za pomocą polecenia *dump* jest dosyć skomplikowany. W celu wykonania pełnego odtwarzania z kopii zapasowej utworzonej za pomocą polecenia *dump*, najpierw należy utworzyć pusty system plików. Następnie muszą być wykonane odtworzenia ostatnich rzutów na odpowiednich poziomach. Pierwszą operacją odtwarzania jest zawsze najbardziej aktualny rzut na poziomie 0. Następny rzut do odtworzenia to rzut o najniższym poziomie. Jeżeli wykonano wiele rzutów na danym poziomie, korzysta się z najbardziej aktualnego. Następnie wszystkie poprzednie rzuty o poziomach wyższych wykonane przed określonym rzutem są pomijane i jest wykonywane odtwarzanie rzutu o kolejnym, najniższym poziomie. Proces trwa do momentu, kiedy będą uwzględnione wszystkie rzuty.

Dyskietki ratunkowe

W przypadku pojawienia się problemów z systemem Debian, najważniejszym narzędziem, które trzeba mieć pod ręką, jest dyskietka ratunkowa. Wiele problemów, które mogą dotknąć system Debian, powoduje, że nie może się on prawidłowo załadować. Inne powodują, że po załadowaniu systemu, korzystanie z niego jest niemożliwe. W takich sytuacjach dyskietki ratunkowe przywrócą system do postaci, która umożliwi odnalezienie błędu i jego naprawę.

Tworząc „skrzynkę z narzędziami” niezbędnymi do odtwarzania systemu w czasie awarii, powinno się umieścić w niej trzy zestawy dyskietek ratunkowych: dostosowaną dyskietkę startową, która została utworzona w czasie instalacji, dyskietkę ratunkową systemu Debian oraz system Linux skonfigurowany do pracy z dyskietek.

Dyskietka dostosowana do potrzeb

W czasie instalacji oraz za każdym razem, gdy instalowane jest nowe jądro, system daje możliwość utworzenia własnej dyskietki startowej. Jeżeli wówczas nie zostanie ona utworzona, można to zrobić w dowolnym momencie poprzez umieszczenie sformatowanej, pustej dyskietki i uruchomienie jako administrator następujących poleceń:

```
dd if=/vmlinuz of=/dev/fd0  
rdev /dev/fd0 /dev/hda1  
rdev -R /dev/fd0 1
```

Tak przygotowana dyskietka startowa umożliwi załadowanie zainstalowanego systemu Linux w przypadku problemów z systemem rozruchowym. W przypadku niepoprawnego

działania systemu LILO, uszkodzenia sektora rozruchowego, utraty lub uszkodzenia obrazu jądra, ta dyskietka przywróci system do stanu pełnej używalności. Nie pomoże w sytuacjach, gdy problem nie pochodzi z systemu rozruchowego lub jądra, ponieważ opiera się na systemie już zainstalowanym na twardym dysku.

Dyskietka ratunkowa systemu Debian

Dyskietka ratunkowa systemu Debian jest stosowana do instalacji systemu. Jest także przydatna jako ogólna dyskietka startowa wykorzystywana w celach odtwarzania systemu, szczególnie w sytuacjach, gdy nie została stworzona dyskietka startowa dostosowana do potrzeb danego systemu.

Dyskietka ratunkowa jest również pomocna w przypadku napotkania na problemy jądra lub problemy sprzętowe. Ponieważ nie obsługuje ona sprzętu oraz ponieważ zawiera znane, poprawne jądro, może pomóc w uruchomieniu systemu w sytuacji, kiedy dziwne zachowanie jądra lub problem sprzętowy będzie przeszkadzał dyskietce startowej dostosowanej do potrzeb danego systemu na załadowanie systemu.

Dokumentacja systemu

Jeżeli kiedykolwiek zaistnieje potrzeba odtworzenia systemu, bardzo przydatna będzie wszelka dostępna dokumentacja. Należy zatem zrobić tę dokumentację wtedy, kiedy wszystko działa poprawnie.

Poniżej przedstawiono niektóre ważne sprawy do umieszczenia w dokumentacji systemu:

- typ systemu (architektura sprzętowa)
- ogólna konfiguracja sprzętu (liczba procesorów, ilość pamięci RAM, urządzenia peryferyjne takie jak modemy, karty SCSI ich ustawienia itp.)
- konfiguracja dysku (IDE czy SCSI, numer modelu, rozmiar, ustawienia napędu jak np. ilość cylindrów oraz ścieżek, numer ID SCSI lub ustawienia IDE podstawowy pomocniczy)
- ustawienia systemu kopii zapasowej (rodzaj nośnika, informacje o sprzęcie itp.), harmonogram wykonywania kopii zapasowych oraz miejsce przechowywania nośników kopii zapasowych
- ustawienia systemu (nazwa, opcje uwierzytelniania itp.)
- ustawienia dotyczące sieci (informacje dotyczące karty sieciowej, ustawienia TCP/IP, uruchomione usługi, ustawienia zapory firewall / NAT itp.)
- najważniejsze zainstalowane oprogramowanie.

SŁOWNIK POJĘĆ

adres	address	Liczba lub nazwa określająca lokalizację urządzenia. W sieci adresem jest liczba nazywana numerem IP, używana do identyfikacji komputera lub urządzenia.
adres e-mail	mail address	Łańcuch znaków określający nadawcę i adresata poczty. Składa się z dwóch części połączonych symbolem @. Z prawej strony tego symbolu znajduje się identyfikator systemu, z lewej – nazwa użytkownika.
adres IP	IP address	Numer identyfikujący komputer w sieci. Podawany w notacji kropkowo-dziesiętnej, składa się z czterech oddzielonych kropkami segmentów, z których każdy może być liczbą od 0 do 255.
adres rozgłoszeniowy	broadcast address	Adres, pod który urządzenia wysyłają pakiety skierowane do wszystkich maszyn w danej sieci.
adres sieci	network address	Adres IP opisujący całą sieć lokalną.
alias		Dodatkowa nazwa polecenia w Linuksie.
archiwizujący program	archiver	Program umieszczający różne pliki w jednym pliku nazywanym archiwum. W systemie Linux najczęściej funkcję taką pełni program <i>tar</i> .
bajt	byte	Jednostka informacji (np. danych lub pamięci) używana przez komputer. Jeden bajt opisuje na przykład kod pojedynczego znaku ASCII. Bajt składa się z ośmiu bitów (cyfr binarnych).
biblioteka	library	Plik zawierający elementy kodu wykorzystywane przez różne programy.
bieżący katalog	current directory	Aktualne miejsce w hierarchii katalogów. Katalog bieżący jest oznaczony przez kropkę i ukośnik (./).
bit		Najmniejsza jednostka informacji używana przez komputer, przedstawiana w postaci liczby binarnej, to jest 0 lub 1.
bug		Błąd w programie.
dowiązanie	link	Plik wskazujący na plik znajdujący się w innym miejscu.
dysk startowy	boot disk	Dyskietka lub dysk umożliwiający uruchomienie komputera i załadowanie systemu operacyjnego.
e-mail		Poczta elektroniczna. Metoda przesyłania wiadomości poprzez sieć.

Ethernet		Standard protokołów i sprzętu komunikacyjnego umożliwiający współużytkowanie danych przez kilka systemów. Sieć w standardzie Ethernet można zrealizować przy pomocy różnego okablowania i różnych systemów połączeniowych.
GNU		Projekt rozwoju oprogramowania, które może być swobodnie rozprowadzane i modyfikowane przez każdego użytkownika. Oprogramowanie GNU (<i>GNU's Not Unix</i>) jest rozpowszechniane przez <i>Free Software Foundation</i> (FSF). Definicja oprogramowania GNU jest zapisana w Powszechnej Licencji Publicznej (GPL – <i>General Public License</i>), której zadaniem jest ochrona swobody rozprowadzania i wprowadzania zmian w kodzie oprogramowania.
Internet		Ogólnoświatowa sieć łącząca sieci komputerowe. Sieć łączy ze sobą komputery i inne urządzenia, a Internet łączy sieci. Protokołem komunikacyjnym używanym w Internecie jest TCP/IP.
jawny tekst	clear text	Czytelny dla człowieka, zwykły plik tekstowy, który może być kodowany za pomocą jednego z powszechnie stosowanych systemów, takich jak ASCII czy UTF-8, nie zawierający żadnego opatentowanego systemu kodowania ani znaków formatujących.
jądro	kernel	Część kodu tworząca rdzeń systemu operacyjnego. Nazwa Linux odnosi się w rzeczywistości do jądra systemu. Istnieje wiele systemu wykorzystujących jądro Linux. Są to tak zwane dystrybucje Linuksa.
karta sieciowa	network card	Karta rozszerzeń umożliwiająca podłączenie komputera do sieci, zwana czasem kartą Ethernet, kartą LAN lub kartą interfejsu sieciowego (NIC, <i>Network Interface Card</i>).
katalog	directory	Miejsce, w którym przechowywane są pliki. W Linuksie istnieje jedna duża struktura katalogów, której podstawa nazywa się katalogiem głównym (root directory) i jest reprezentowana ukośnikiem (/). W katalogach mogą znajdować się pliki i inne katalogi. Katalogi leżące w innych katalogach są podkatalogami tych katalogów. Katalogi zawierające inne katalogi są dla nich katalogami nadrzędnymi. Katalog nadrzędny jest przedstawiany jako dwie kropki i ukośnik (../).
kolejka	spool	Inaczej bufor. Obszar, w którym tymczasowo przechowywane są dane, zanim zostaną wysłane do urządzenia lub programu. W Linuksie dane takie są często umieszczane w katalogach <i>/var/spool</i> i <i>/usr/spool</i> . W kolejce przechowywane są wiadomości pocztowe oczekujące na odebranie przez użytkowników oraz dane wysyłane do drukarki.
kompilacja	compile	Kompilacja to operacja przekształcania kodu źródłowego w plik wykonywalny dokonywana za pomocą kompilatora.
kompresja	compress	Zmniejszanie wielkości plików. Przywrócenie plików do stanu pierwotnego nazywa się dekompresją. W Linuksie najczęściej stosowanym programem do kompresji jest <i>gzip</i> .

koncentrator	hub	Urządzenie do połączenia komputerów PC w sieć Ethernet.
kopia zapasowa	backup	Zapisywanie i przechowywanie danych w osobnej lokalizacji w celu zabezpieczenia przed utratą danych w przypadku awarii sprzętu lub oprogramowania.
LAN		Dwa lub więcej komputerów położonych względnie niedaleko od siebie (w tym samym budynku, w sąsiadujących budynkach) połączonych ze sobą tak, że mogą przysyłać informacje między sobą.
Linux		Wywodzący się z Uniksa system operacyjny. Nazwa Linux jest powszechnie stosowana w odniesieniu do systemów operacyjnych używających jądra Linux.
logowanie	log in	Rejestrowanie użytkownika w systemie, odbywa się przed rozpoczęciem pracy w tym systemie.
maska sieci	netmask	Numer określający zakres adresów IP w danej sieci lokalnej.
maskowanie IP	IP masquerade	Oprogramowanie realizujące w Linuksie translację adresów sieciowych (NAT).
menedżer startu	boot manager	Program umożliwiający wybór systemu operacyjnego podczas uruchamiania komputera.
montowanie	mount	Podłączanie systemu plików do określonego katalogu (punktu montowania).
NAT		Translacja adresów sieciowych. Translacja prywatnych adresów w sieci LAN na publiczny adres IP, który może być używany w Internecie.
nazwa domeny	domain name	Końcowa część pełnej nazwy domenowej komputera. Jest to łańcuch znaków określający zwykle sieć, do której należy dany adres. W nazwach domen używa się znaków alfanumerycznych i myślników, a poszczególne człony oddziela się kropkami.
nazwa hosta	hostname	Łańcuch znaków określający nazwę komputera. Nazwa hosta może być przedstawiona w postaci pełnej nazwy domenowej lub bez części określającej domenę.
obszar wymiany	swap	Plik lub partycja używana przez system operacyjny do tymczasowego zapisywania danych z pamięci operacyjnej. W Linuksie jako obszar wymiany używana jest dedykowana partycja na dysku twardym.
pakiet	package	W Linuksie pakiet to archiwum zawierające pliki z programami, umożliwiające ich szybką instalację.
partycja	partition	Obszar dysku twardego. Na dysku mogą istnieć najwyżej cztery partycje podstawowe, a kolejne partycje (logiczne) mogą istnieć tylko na partycji rozszerzonej, zastępującej jedną z partycji podstawowych.
Pełna nazwa domenowa	Fully Qualified Domain Name	Pełna nazwa komputera, wraz z nazwą domeny.

plik dziennika	log	Plik dziennika zawiera zapis operacji wykonywanych przez jądro i demony
plik indeksu	index file	Domyślny plik wyświetlany w sytuacji, gdy podany zostanie tylko adres serwisu WWW. Zwykle plik taki nazywa się <i>index.html</i>
pliki ukryte	dot files	Pliki, których nazwa rozpoczyna się od kropki. Są to zwykle pliki konfiguracyjne lub pliki zasobów poszczególnych programów dla systemu Linux.
polecenie	command	Kierowane do systemu polecenia wykonania określonej czynności. Do wydawania poleceń służy wiersz poleceń (shell), podstawowy interfejs użytkownika, pozwalający na wprowadzanie łańcuchów znaków (nazw poleceń) i wyświetlanie wyników działania polecenia. Wśród argumentów wyróżnia się opcje, które określają sposób działania polecenia. W przypadku wielu opcji można je wpisywać osobno, poprzedzając każdą myślnikiem, lub razem, poprzedzając cały łańcuch tylko jednym myślnikiem. Istnieją jednak wyjątki od tej reguły.
port		Słowo port ma dwa znaczenia: pierwsze to gniazdo w koncentratorze, do którego podłącza się kable sieciowe. Drugie znaczenie to lokalizacja danej usługi sieciowej w protokole TCP/IP, która jest podawana właśnie przez numer portu.
proces	process	Uruchomiony program działający w systemie operacyjnym. Każdy proces oznaczony jest liczbą całkowitą nazywaną identyfikatorem procesu. Proces jest tworzony w momencie uruchomienia programu i wówczas przypisywany jest mu identyfikator. Dwa działające jednocześnie procesy nie mogą mieć tego samego identyfikatora. Po zakończeniu programu proces jest usuwany.
program		Kod komputerowy, który może być wykonany (uruchomiony) na komputerze.
protokół	protocol	Zestaw reguł i formatów wykorzystywanych przez komputery do komunikacji między sobą.
proxy		Serwer pośredniczący w dostępie do Internetu. Adresy IP korzystających z niego komputerów nie są upubliczniane.
przekazywanie	relay	Wykorzystywanie postronnego serwera poczty do wysyłania poczty do innych serwerów.
root		Użytkownik pełniący funkcję administratora systemu Linux. Nazywany także super-użytkownikiem.
ruter	router	Urządzenie w węźle internetowym używane do przekazywania pakietów TCP/IP.
rutowanie	routing	Proces przekazywania pakietów z rutera do rutera, zanim dotrą one do miejsca docelowego.
serwer	server	Serwer jest programem, który świadczy usługi różnego rodzaju na rzecz innych komputerów.

serwer wirtualny	virtual host	Na zewnątrz (w Internecie) wygląda jak każdy inny komputer, ale w rzeczywistości działa na jednej maszynie z innymi serwerami.
sieć	network	Zespół połączonych komputerów, które mogą komunikować się między sobą.
stałe połączenie	permanent connection	W odróżnieniu od połączenia komutowanego, połączenie stałe łączy sieć lokalną z Internetem bez przerwy. Zwykle połączenia stałe są realizowane poprzez łącza dzierżawione i linie cyfrowe.
sterownik	driver	Oprogramowanie niezbędne do obsługi określonego sprzętu (sterownik urządzenia).
strefa	zone	Część domeny obsługiwana przez określony serwer nazw. Jest podobna do domeny, ale zawiera tylko nazwy domeny i informacje o domenie nie przesyłane dalej.
system operacyjny	operating system (OS)	Podstawowe oprogramowanie umożliwiające korzystanie z komputera. System operacyjny przydziela zasoby komputera użytkownikom i procesom.
system plików	file system	Struktura określająca, w jaki sposób pliki zapisywane są na dysku
ścieżka	path	Droga prowadząca przez kolejne katalogi do pliku. Katalogi w ścieżce rozdziela się ukośnikami (/). Ścieżka, która rozpoczyna się od katalogu głównego nazywa się ścieżką bezwzględną. Ścieżka rozpoczynająca się od bieżącego katalogu (innego niż główny) to ścieżka względna.
środowisko programistyczne	development environment	Zestaw plików i programów potrzebnych do tworzenia programów. Główne środowisko programistyczne w Linuksie to GNU.
tekstowy plik	text file	Najprostszy format pliku dokumentu, w którym zapisana jest tylko informacja znakowa. Pliki tekstowe nie zależą od konkretnych aplikacji i mają wiele zastosowań.
terminal wirtualny	pseudo tty (virtual terminal)	Terminal wirtualny umożliwia zdalne wykonywanie poleceń linuksowych na komputerze, na którym Linux nie jest zainstalowany. Ekran Linuksa emulowany na takim komputerze nazywa się terminalem wirtualnym (pseudo tty)
UNIX	Unix family OS	Unix powstał w laboratoriach Bella firmy AT&T w roku 1971. Od tego czasu był rozwijany przez różne zespoły i obecnie istnieje mnóstwo systemów operacyjnych opartych na Uniksie.
uprawnienia	permissions	Prawa dostępu do plików i katalogów. W Linuksie można przyznać lub odebrać użytkownikom prawo do odczytu, zapisu i wykonywania plików. Użytkownicy dzielą się na trzy grupy: właściciel, grupa użytkowników i pozostali użytkownicy.
urządzenie	device	Wyposażenie podłączone do komputera. W Linuksie wszystkie urządzenia reprezentowane są przez pliki urządzeń, pozwalające na dostęp do nich. Nazwa pliku odpowiadającego danemu urządzeniu jest jednocześnie systemową nazwą urządzenia. Pliki urządzeń są zwykle przechowywane w katalogu <i>/dev</i> .

użytkownik	user	Osoba zarejestrowana w systemie Linux. Użytkownik to podstawowa jednostka przy zarządzaniu uprawnieniami.
właściciel	owner	Zwykle właścicielem pliku lub katalogu jest użytkownik, który taki plik lub katalog utworzył.
znak zachęty	prompt	Łańcuch znaków wyświetlany na ekranie, oznaczający gotowość systemu do przyjęcia polecenia od użytkownika.

BIBLIOGRAFIA

- [1] O. Andreasson. *Iptables Tutorial*. 2001.
- [2] O. Aoki. *Debian Reference*. 2002.
- [3] B. Ball. *Linux*. HELION, 1998.
- [4] M. Camou, J. Goerzen, A. Van Couwenberghe. *Debian Linux. Księga eksperta*. HELION, 2001.
- [5] M. Czajko, M. Zasada. *Elementarz un*x'owy*. 2000.
- [6] F. Defler. *Sieci komputerowe dla każdego*. HELION, 2001.
- [7] R. Eckstein, D. Collier-Brown, P. Kelly. *Samba*. RM, 2000.
- [8] C. Hunt. *Serwery sieciowe Linuksa*. MIKOM, 2000.
- [9] K. Husain, T. Parker. *Slackware Linux Unleashed*. 1994.
- [10] O. Kirch, T. Dawson. *Linux. Podręcznik administratora sieci*. RM, 2000.
- [11] B. Kiziukiewicz. *Sieci lokalne*. 2001.
- [12] K. Krawczyk, M. Topka, T. Zatorski. *Administrowanie systemami komputerowymi. Bezpieczeństwo systemu*. 2001.
- [13] P. Krawczyk. *Filtrowanie stateful-inspection w Linuksie i BSD*. 2001.
- [14] K. Krysiak. *Analiza i optymalizacja zasobów sieci LAN*. Politechnika Łódzka, 2000.
- [15] T. Parker, M. Sportack, M. Kadrach. *TCP/IP. Księga eksperta*. HELION, 2000.
- [16] J. Peña, A. Reelsen. *Securing Debian Manual*. 2002.
- [17] B. Perens, S. Rudolph, I. Grobman. *Installing Debian GNU/Linux 3.0 For Intel x86*. 2002.
- [18] A. Rangelrooij, O. Elphick, T. Lehtonen. *Debian GNU/Linux System Administrator's Manual*. 2002.
- [19] D. Scheetz. *Dwarf's Guide to Debian GNU/Linux*. 2001.
- [20] K. Stępień. *Świat sieci komputerowych*. 2002.
- [21] S. Strobel, T. Uhl. *Linux*. Wydawnictwa Naukowo-Techniczne, 1997.
- [22] Z. Suski, P. Kołodziejczyk. *Ochrona sieci lokalnej za pomocą zapory sieciowej*. 2000.
- [23] P. Tęcza. *Narzędzia do zarządzania pakietami Debiana*. 2002.
- [24] H. Tsuji, T. Watanabe. *Linux Internet Server. Czarna księga*. HELION, 2001.
- [25] L. Wirzenius, J. Oja. *The Linux System Administrators' Guide*. 1998.
- [26] J. Zieliński, T. Rak. *Domowe sieci komputerowe*. HELION, 2002.