Systemy i sieci komputerowe. Podręcznik do nauki zawodu technik informatyk

Autor: Paweł Bensel
ISBN: 978-83-246-2388-4
Format: 168×237, stron: 336

Podręcznik jest zgodny z podstawą programową kształcenia w zawodzie technik informatyk 312[01].

Swobodne poruszanie się wśród najpopularniejszych systemów operacyjnych, świetna znajomość ich architektury, obsługa urządzeń wejścia-wyjścia, a także tworzenie sieci komputerowych to podstawowe umiejętności technika informatyka. W tym podręczniku zebrano wszystkie najważniejsze informacje, pozwalające poznać i zrozumieć zależności między komputerem a jego systemem operacyjnym oraz opanować kwestie dotyczące administrowania czy zarządzania systemami Windows i Linux. Przedstawiono tu sposoby łączenia komputerów w sieci oraz omówiono zasady działania Internetu i nadawania adresów IP. Dzięki tej książce uczeń pozna także tajniki obsługi serwera sieci lokalnej. Podręcznik zawiera zagadnienia poruszane na egzaminie zawodowym potwierdzającym kwalifikacje.

„Technik Informatyk" to doskonały, charakteryzujący się wysoką jakością i kompletny zestaw edukacyjny, przygotowany przez dysponującego ogromnym doświadczeniem lidera na rynku książek informatycznych – wydawnictwo Helion.

W skład zestawu „Technik Informatyk" wchodzą także:

- „Programowanie strukturalne i obiektowe. Podręcznik do nauki zawodu technik informatyk"
- „Multimedia i grafika komputerowa. Podręcznik do nauki zawodu technik informatyk"
- „Urządzenia techniki komputerowej. Podręcznik do nauki zawodu technik informatyk"
- „Oprogramowanie biurowe. Podręcznik do nauki zawodu technik informatyk"

Podręczniki oraz inne pomoce naukowe należące do tej serii zostały opracowane z myślą o wykształceniu kompetentnych techników, którzy bez trudu poradzą sobie z wyzwaniami w świecie współczesnej informatyki.

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Wstęp

Systemy i sieci komputerowe to podręcznik do nauki w zawodzie technik informatyk. Podręcznik jest zgodny z podstawą programową kształcenia w zawodzie technik informatyk 312[01]. Treści zawarte w podręczniku mogą być również z powodzeniem wykorzystywane w przypadku innych kierunków kształcenia, a także przez osoby, które samodzielnie poszerzają swą wiedzę z zakresu systemów operacyjnych z grupy Windows, Linux oraz sieci komputerowych.

W podręczniku duży nacisk został położony na praktyczne stosowanie zdobywanej wiedzy, co przekłada się na dużą liczbę opisanych instrukcji postępowania, dzięki którym czytelnik w łatwy sposób może wdrożyć je w systemie komputerowym.

Podręcznik składa się z 19 rozdziałów omawiających kolejne zagadnienia programu nauczania oraz rozdziału zawierającego zadania występujące na egzaminie potwierdzającym kwalifikacje zawodowe z zakresu systemów operacyjnych i sieci.

Rozdział 1. „Zasady pracy z komputerem" zawiera informacje dotyczące bezpieczeństwa i higieny pracy, omawia zagadnienia związane z licencjonowaniem oprogramowania oraz prezentuje oprogramowanie do wirtualizacji zasobów komputerowych, które może być przydatne podczas przeprowadzania ćwiczeń opisanych w podręczniku.

Rozdział 2. „Budowa komputera" omawia podstawowe elementy komputera, różnice między pamięciami komputerowymi oraz przedstawia rodzaje oprogramowania komputerowego.

Rozdział 3. „Systemy plików" prezentuje systemy plików, ich porównanie, a także oprogramowanie narzędziowe do ochrony danych na dyskach.

Rozdział 4. „Obsługa urządzeń wejścia-wyjścia" omawia konfiguracje urządzeń służących do komunikacji użytkownika z komputerem.

Rozdział 5. „Zarządzanie systemem Windows" prezentuje zagadnienia związane z konfiguracją systemu operacyjnego, edycją rejestru systemowego, zarządzaniem użytkownikami oraz narzędziami administracyjnymi.

Rozdział 6. „Architektura systemu Windows" przedstawia budowę systemów operacyjnych z grupy Windows. Opisuje jądro systemu, sposób zarządzania pamięcią, tryby pracy procesora oraz etapy uruchamiania systemu Windows.

Rozdział 7. „Instalacja systemu Windows" zawiera opis typowej instalacji systemu operacyjnego, instalacji sterowników i oprogramowania, a także instalacji zaawansowanych przy użyciu narzędzi mechanizmu klonowania systemu czy usługi zdalnej instalacji.

Rozdział 8. „Praca w trybie MS-DOS" przedstawia zasady pracy w trybie tekstowym — opisuje polecenia systemu, konfiguracje pamięci, nakładki na tryb tekstowy oraz mechanizm drukowania.

Rozdział 9. „Konfiguracja sieciowa systemu Windows" prezentuje sposoby podłączania komputera z systemem Windows do sieci lokalnej oraz internetu, a także opisuje podstawowe usługi sieciowe.

Rozdział 10. „Praca w sieci równoprawnej" przedstawia mechanizmy funkcjonowania i pracy w sieci typu peer-to-peer.

Rozdział 11. „Praca w sieci z serwerem" omawia zasady działania oraz pracy w sieci zbudowanej w architekturze klient-serwer.

Rozdział 12. „Instalacja systemu Linux" opisuje kolejne etapy instalacji systemu Linux na przykładzie dystrybucji Fedora.

Rozdział 13. „Architektura systemu Linux" prezentuje budowę systemu operacyjnego Linux — omawia jądro systemu, zarządzanie pamięcią, system plików, a także uprawnienia do zasobów przypisywane użytkownikom.

Rozdział 14. „Praca w systemie Linux" opisuje podstawy pracy w Linuksie. Zawiera spis najważniejszych poleceń w trybie tekstowym (w tym obsługę edytora tekstu vi/vim), zasady pracy w trybie graficznym oraz sposoby instalacji oprogramowania.

Rozdział 15. „Administracja systemem Linux" przedstawia zagadnienia związane z administrowaniem systemem — konfiguracje interfejsów sieciowych, instalację i konfigurację usług sieciowych oraz zarządzanie użytkownikami.

Rozdział 16. „Wprowadzenie do sieci komputerowych" przedstawia zagadnienia teoretyczne dotyczące sieci komputerowych — definiuje rodzaje sieci, podstawowe urządzenia i topologie sieciowe, model OSI.

Rozdział 17. „Adresacja IP" omawia zasady adresowania za pomocą protokołu IP, zawiera opis mechanizmu podziału sieci na podsieci wraz z przykładami.

Rozdział 18. „Internet" opisuje zasady działania globalnej sieci, prezentuje mechanizm translacji adresów, sposoby podłączenia do internetu oraz wprowadza w zagadnienia bezpieczeństwa sieciowego.

Rozdział 19. „Administracja serwerem sieci lokalnej" przedstawia podstawy zarządzania serwerem lokalnym — od jego instalacji poprzez wybór roli, jaką ma odgrywać w sieci, do tworzenia zasad grup dla użytkowników.

Rozdział 20. „Zadania występujące na egzaminie potwierdzającym kwalifikacje zawodowe" przedstawia zadania oraz pytania występujące na egzaminie.

Rozdziały dotyczące systemu Linux opracowane zostały na podstawie dystrybucji Fedora. Pliki zawierające obrazy płyt z tym systemem dostępne są pod adresem *http://fedoraproject.org/pl/get-fedora*.

Aby nagrać zawartość pobranego pliku na płytę, należy skorzystać z programu nagrywającego. Przy wyborze typu danych do nagrania należy wybrać *obraz płyty*, a następnie wskazać miejsce, gdzie zapisany został pobrany plik.

13

Architektura systemu Linux

Linux powstał na bazie systemu Unix, który rozwijany był od końca lat 60. ubiegłego wieku. Wiele cech Uniksa zostało zaimplementowanych w nowym systemie.

System Linux wyposażony został w mechanizm **wielozadaniowości z wywłaszczeniem**, czyli możliwość wykonywania wielu zadań jednocześnie. Podobnie jak w przypadku systemów Windows, jeśli system wyposażony jest w jeden procesor, czas procesora dzielony jest przez specjalny algorytm szeregujący zadania, który zaimplementowany został w jądrze systemu.

Również **wielodostępność** jest cechą systemu Unix przeniesioną do Linuksa. Pozwala ona na jednoczesną pracę wielu użytkowników. Użytkownicy nie muszą pracować na klawiaturze i monitorze podłączonych do komputera, wystarczy stacja robocza, która poprzez sieć pozwala uruchamiać zadania na serwerze. Implementacja wielodostępności wymusiła konieczność stworzenia mechanizmu ochrony zasobów. Każdy plik i katalog ma właściciela i grupę, do której jest przypisany, a także prawa dostępu określane dla właściciela pliku, użytkowników należących do grupy oraz pozostałych. Każdorazowy dostęp do komputera wymaga autoryzacji, co pozwala na kontrolę zabezpieczeń.

Systemy Windows nie są systemami wielodostępnymi. Od wersji systemu Windows XP możliwe jest przełączanie między użytkownikami, przy czym dostępne jest ono tylko dla komputerów niepracujących w domenie Active Directory. Przełączanie między użytkownikami w systemach Windows nie pozwala im na jednoczesną pracę, tak jak ma to miejsce w systemach Linux. Funkcjonalność tego typu zapewniają usługi terminalowe na serwerach Windows.

13.1. Zarządzanie pamięcią

Jednym z podstawowych zadań systemu operacyjnego jest **zarządzanie pamięcią**. Przed wykonaniem kod programu i jego dane muszą zostać wczytane do pamięci operacyjnej. Uruchamianie wielu programów jednocześnie powoduje, że zasoby pamięci

mogą się wyczerpać. Aby tego uniknąć i aby poszczególne procesy mogły działać bez zakłóceń, system operacyjny dynamicznie przydziela im pamięć.

Celem zarządzania pamięcią operacyjną jest:

- przydział pamięci fizycznej poszczególnym procesom,
- odwzorowanie wirtualnej przestrzeni adresowej procesu na fizyczną przestrzeń adresową pamięci,
- ochrona zawartości pamięci,
- współdzielenie obszarów pamięci przez różne procesy.

Linux obsługuje mechanizm **pamięci wirtualnej** jako rozszerzenie pamięci fizycznej. Jądro zapisuje zawartość nieużywanych bloków pamięci fizycznej na dysku, umożliwiając tym samym późniejsze ich wykorzystanie. Operacje przekazywania danych z pamięci fizycznej do pamięci wirtualnej prowadzone są przez system operacyjny. Nie są one widoczne ani dla użytkownika, ani dla uruchomionych programów.

Operacje dyskowe są znacznie wolniejsze (czas dostępu około 10 ms) niż analogiczne działania na fizycznej pamięci (czas dostępu około 5 ns), dlatego przy częstym wykorzystaniu pamięci wirtualnej szybkość działania programów spada. Część dysku twardego wykorzystywana jako pamięć wirtualna nosi nazwę **obszaru wymiany**.

13.2. System plików

System plików tworzy mechanizm bezpośredniego przechowywania i dostępu do danych zapisanych na dyskach. Na potrzeby systemu Linux stworzony został **system plików EXT** (ang. *Extended File System*). Wraz z rozwojem systemu operacyjnego tworzone były kolejne wersje systemu plików. W roku 2008 wydany został system EXT4, który umożliwia obsługę woluminów o wielkości do 1024 petabajtów (1 petabajt = 1024 terabajty).

System plików systemu Linux, w przeciwieństwie do systemu Windows, nie dzieli przestrzeni dyskowej na dyski logiczne. W systemie operacyjnym dostępny jest tylko jeden katalog główny z hierarchiczną strukturą katalogów.

Katalog główny oznaczany jest ukośnikiem — znakiem /. Katalogi w systemie Linux przedstawione zostały poniżej.

- */* — katalog główny.
- *bin* — zawiera wykonywalne pliki najbardziej podstawowych narzędzi systemowych, dostępne dla wszystkich użytkowników.
- *boot* — zawiera pliki niezbędne do uruchomienia systemu, a w przypadku większości dystrybucji — także obraz jądra systemu.
- *dev* — zawiera pliki specjalne wskazujące na urządzenia w systemie; za ich pomocą system komunikuje się z tymi urządzeniami.
- *etc* — zawiera pliki konfiguracyjne systemu.

DEFINICJA

W systemie Linux wyróżnia się następujące rodzaje plików:

Plik zwykły (w wynikach działania komendy `ls -l` oznaczany znakiem –) — zbiór danych zapisanych na dysku.

Katalog (oznaczany literą *d*) — katalog na dysku.

Dowiązanie symboliczne (oznaczane literą *l*) — plik wskazujący na inny plik.

Urządzenie znakowe (oznaczane literą *c*) — plik specjalny reprezentujący urządzenie, do którego dostęp realizowany jest znak po znaku (bajt po bajcie).

Urządzenie blokowe (oznaczane literą *b*) — plik specjalny reprezentujący urządzenie, do którego dostęp realizowany jest poprzez większe porcje danych zwane blokami.

Nazwany potok (ang. *named pipe*) (oznaczany literą *p*) — plik wymiany informacji między procesami, działający jako kolejka FIFO (ang. *first in first out*).

Gniazdo (ang. *socket*) (oznaczany literą *s*) — plik wymiany między procesami.

- *home* — w tym katalogu znajdują się katalogi domowe użytkowników systemu.
- *lib* — zawiera dzielone biblioteki systemowe i moduły jądra (w katalogu */lib/modules*).
- *mnt* — tutaj są montowane (podłączane do systemu) dodatkowe dyski (np. partycje systemu Windows).
- *proc* — wirtualny katalog zawierający informacje o uruchomionych procesach.
- *root* — katalog domowy użytkownika *root*.
- *sbin* — zawiera pliki wykonywalne, które mogą być wykonywane tylko przez administratora systemu.
- *sys* — zawiera pliki systemu operacyjnego.
- *tmp* — katalog służący do zapisu plików tymczasowych.
- *usr* — katalog zawierający dodatkowe oprogramowanie (odpowiednik katalogu *Program Files* w systemie Windows).
- *var* — katalog przeznaczony na pliki, które często ulegają zmianie, np. logi systemowe, pliki udostępniane przez serwer WWW itp.

Linux urządzenia podłączone do komputera postrzega jako pliki, co powoduje, że każde urządzenie ma swój odpowiednik w katalogu *dev*. Aby odwołać się do jakiegoś urządzenia, system wykorzystuje odpowiedni plik w tym katalogu. Najważniejsze urządzenia w systemie operacyjnym przedstawiane są za pomocą następujących katalogów:

- */dev/console* — konsola systemu operacyjnego,
- */dev/mouse* — mysz szeregowa,
- */dev/hda* — pierwszy dysk IDE,
- */dev/hda1* — pierwsza partycja pierwszego dysku,

WSKAZÓWKA

W systemie plików Linux dane są uporządkowane. Podłączenie kolejnego dysku do systemu wymaga jego *zamontowania* (ang. *mount*). Dotyczy to zarówno płyt CD, jak i dysków twardych. Dostęp do danych zapisanych na tych dyskach możliwy jest poprzez katalog, w którym zostały one zamontowane. Jeśli na przykład dysk z systemem plików FAT32 zamontujemy w katalogu */mnt/drive_c*, to dostęp do danych zapisanych na tym dysku umożliwia katalog, w którym został on zamontowany. Jeśli na dysku FAT32 w systemie Windows utworzony został katalog *c:\zdjecia*, w systemie Linux dostępny on będzie pod adresem */mnt/drive_c/zdjęcia*.

- */dev/hda2* — druga partycja pierwszego dysku,
- */dev/hdb* — drugi dysk IDE,
- */dev/hdb1* — pierwsza partycja drugiego dysku,
- */dev/fd0* — pierwsza dyskietka,
- */dev/lp0* — pierwszy port drukarki,
- */dev/null* — urządzenie puste (do testów).

13.3. Jądro systemu Linux

Podstawę systemu Linux stanowi **jądro** (ang. *kernel*), którego funkcją jest podstawowa obsługa sprzętu, dysków, systemu plików, wielozadaniowości i zabezpieczeń. Jądro Linuksa zostało napisane w języku C, dostępne jest zarówno w postaci kodu źródłowego do samodzielnej kompilacji, jak i gotowych do użycia pakietów. Aby skorzystać z jądra udostępnionego w postaci kodu źródłowego, należy je skompilować przy użyciu kompilatora — oprogramowania tłumaczącego język programistyczny na język maszynowy zrozumiały dla mikroprocesora. Samodzielna kompilacja jądra pozwala zwiększyć szybkość działania systemu w porównaniu z systemem działającym na podstawie gotowych pakietów.

W przypadku systemu Linux wyróżnia się dwie wersje jądra: **wersję stabilną** (ang. *stable*) i **wersję rozwojową** (ang. *development*). Pierwsza z nich jest wersją sprawdzoną, przetestowaną i oficjalnie dopuszczoną do rozpowszechniania. Wersje rozwojowe przeznaczone są dla ludzi pracujących nad rozwojem systemu. Zawierają nowe elementy, które są testowane przed wydaniem w wersji stabilnej.

Jądro systemu Linux potrafi ładować i usuwać dowolną część swojego kodu stosownie do potrzeb. Osobno ładowane części noszą nazwę **modułów** — zazwyczaj obsługują one wybrane systemy plików i protokoły sieciowe lub pełnią funkcję sterowników urządzeń. Moduły jądra działają w tzw. trybie jądra. Jest to tryb pracy programów pozwalający na pełen dostęp do zasobów komputera, na którym zostały uruchomione. Wszystkie moduły dostępne w systemie znajdują się w katalogu */lib/modules/numer_wersji_naszego_jądra*.

Modularna budowa jądra niesie z sobą wiele korzyści. Niepotrzebne elementy nie są wczytywane do pamięci, dzięki czemu system może działać szybciej, stabilniej i zużywać mniej zasobów.

13.4. Interpreter poleceń

Jądro systemu operacyjnego zapewnia zarządzanie pamięcią i procesami, dostęp do zgromadzonych danych itp. Za komunikację z użytkownikiem odpowiedzialna jest powłoka systemu operacyjnego (ang. *shell*), zwana także **interpreterem poleceń**. Powłoka systemu Linux pełni taką samą funkcję jak plik *command.com* w systemie DOS, przy czym użytkownik może wybrać jedną z kilku dostępnych powłok systemowych.

DEFINICJA

Powłoka systemu operacyjnego to program, który udostępnia interfejs między użytkownikiem a jądrem systemu. W przypadku systemu Linux ma ona postać wiersza poleceń. Jądro systemu zawiera wszelkie procedury potrzebne do przeprowadzania operacji wejścia i wyjścia, zarządzania plikami itp., powłoka pozwala z nich korzystać. Powłoki obsługują również skryptowy język programowania umożliwiający tworzenie tzw. skryptów powłoki (odpowiedniki plików wsadowych w systemie DOS).

W systemie Linux najczęściej używane są następujące powłoki systemowe:

- *sh* (od ang. *shell*) — to powłoka stworzona dla systemów Unix przez Stephena Bourne'a, zwana także powłoką Bourne'a.
- *rsh* — jest jedną z odmian powłoki Bourne'a, udostępniającą okrojone funkcje powłoki *sh*. Litera *r* w jej nazwie pochodzi od słowa *reduced*, czyli ograniczona.
- *csh* (od ang. *C shell*) — jest jedną z powłok systemowych, która nawiązuje do składni języka C. Powłoka *csh* wniosła wiele ulepszeń w stosunku do *sh*, m.in. takich jak aliasy i historia komend.
- *ksh* (od ang. *korn shell*) — jest całkowicie kompatybilna wstecz z powłoką *sh*, zawiera także elementy powłoki *csh*, takie jak historia wpisanych komend. Powłoka *ksh* zawiera wbudowany system obliczania wyrażeń arytmetycznych i funkcje skryptów podobne do tych używanych w bardziej zaawansowanych językach programowania, takich jak awk, sed i perl.
- *bash* (od ang. *Bourne again shell*) — rozszerzona powłoka, zawierająca historię poleceń i konstrukcje umożliwiające sterowanie przepływem danych (`if`, `while`, `for`). Jest ona domyślną powłoką systemu Linux.

W systemie Linux powłoka ładowana jest po zalogowaniu użytkownika.

Polecenia rozpoznawane przez interpreter dotyczą:

- tworzenia procesów i zarządzania nimi,
- obsługi urządzeń wejścia-wyjścia,

- administrowania pamięcią pomocniczą i operacyjną,
- dostępu do plików i katalogów.

13.5. Konsola, terminal

Linux powstał jako system tekstowy. Graficzny interfejs użytkownika jest nakładką. W niektórych dystrybucjach po instalacji domyślnie ładowany jest graficzny system okienek dla systemu Linux, zwany X Window.

DEFINICJA

System Linux powstał na bazie systemu Unix, który eksploatowany był na komputerach typu mainframe — wydajnych maszynach, gdzie praca odbywała się poprzez *terminal*, czyli zestaw złożony z monitora i klawiatury, pozwalający na komunikację z komputerem. Aby lepiej wykorzystywać moc obliczeniową, do komputera podłączano wiele terminali, które umożliwiały pracę wielu użytkownikom jednocześnie.

Mianem *konsoli* określa się mechanizm tekstowej komunikacji z systemem operacyjnym. Przykładem konsoli jest wiersz poleceń w systemie Windows lub tryb tekstowy systemu Linux. Obecnie terminy „konsola" i „terminal" używane są zamiennie, przy czym ich geneza jest różna.

Po uruchomieniu systemu Linux użytkownik ma do dyspozycji 7 **konsoli wirtualnych** — 7 środowisk pozwalających na jednoczesną pracę użytkowników. Przełączanie między nimi umożliwia kombinacja klawiszy *Alt+F1* – *Alt+F7*. Przejście do poszczególnych konsoli z uruchomionego środowiska X Window (ładowanego na konsoli 7) umożliwia kombinacja *Ctrl+Alt+F1* – *Ctrl+Alt+F6*.

Na każdej konsoli można zalogować się jako inny użytkownik, dzięki czemu wykonywanych jest wiele zadań jednocześnie. Pracując na konsoli tekstowej, użytkownik może uruchamiać zadania w tle, co pozwala na jednoczesne przetwarzanie wielu zadań, jak również na pozostawienie działających programów po wylogowaniu użytkownika.

Linux umożliwia pracę na konsoli tekstowej, gdy uruchomione jest środowisko X Window. W tym celu konieczne jest aktywowanie **emulatora terminala**, który wyświetli okno tekstowe z prawami aktualnie zalogowanego użytkownika.

W systemie Linux istnieje możliwość zdalnej pracy. Poprzez program emulujący terminal można połączyć się z serwerem i uruchamiać wybrane zadania. Zdalne logowanie do systemu pozwala uruchomić wszystkie zadania, w tym także środowisko graficzne. Programy pozwalające na zdalną pracę to telnet (polecenia przesyłane są w formie niezaszyfrowanej) oraz ssh (polecenia przesyłane są w formie szyfrowanej).

13.6. Użytkownicy, grupy, autoryzacja

W systemie Linux bardzo ważną rolę odgrywają **uprawnienia** do zasobów. Uprawnienia nadawane są użytkownikom i grupom użytkowników. Powoduje to, że każdy użytkownik przed rozpoczęciem pracy musi zalogować się do systemu, aby ten mógł mu przydzielić dostęp do zasobów, chroniąc zarazem zasoby systemowe oraz te należące do innych użytkowników. **Autoryzacja** polega na sprawdzeniu, czy dany użytkownik został wcześniej dodany do systemu przez administratora oraz czy podane przez niego hasło jest zgodne z hasłem przypisanym mu w systemie.

Informacje o kontach użytkowników przechowywane są w pliku */etc/passwd*, informacje o hasłach — w pliku */etc/shadow*.

W pliku */etc/passwd* przechowywane są następujące informacje:

- **nazwa użytkownika** — jednoznacznie identyfikująca konto użytkownika;
- **identyfikator użytkownika UID** — numer jednoznacznie identyfikujący użytkownika w systemie;
- **identyfikator grupy GID** — numer grupy, do której należy użytkownik;
- **katalog domowy** — prywatny katalog użytkownika, w którym może bezpiecznie przechowywać pliki, zabezpieczone przed dostępem innych użytkowników;
- **powłoka logowania** — nazwa interpretera poleceń, który jest uruchamiany po zalogowaniu użytkownika.

Użytkownicy systemu przypisywani są do grup użytkowników, co pozwala na łatwiejsze zarządzanie uprawnieniami. Jeśli podczas dodawania nowego użytkownika nie zostanie określona grupa, do której ma on zostać zapisany, w systemie powstanie grupa o nazwie takiej samej jak nazwa użytkownika. Informacje o członkach poszczególnych grup przechowywane są w pliku */etc/group*.

Autoryzacja użytkowników w systemie Linux przebiega w sposób następujący:

- sprawdzenie, czy użytkownik o podanej nazwie jest zarejestrowany w systemie w pliku */etc/passwd*;
- zakodowanie podanego hasła;
- porównanie z zakodowanym hasłem przechowywanym przez system w pliku */etc/shadow*.

W wyniku pomyślnej autoryzacji system uruchamia sesję użytkownika.

Każdy użytkownik systemu ma własny identyfikator **UID** (ang. *User IDentifier*) oraz identyfikator grupy, do której należy — **GID** (ang. *Group IDentifier*). Na podstawie tych danych system rozpoznaje, czy użytkownik dysponuje prawem dostępu do plików i katalogów.

W każdym systemie Linux istnieje konto administratora systemu — uprzywilejowanego użytkownika o nazwie *root*. Dysponuje on nieograniczonymi uprawnieniami. Jednym z jego zadań jest zakładanie kont nowym użytkownikom.

Zarządzanie użytkownikami umożliwiają następujące polecenia:

- `useradd` — dodaje nowego użytkownika;
- `userdel` — usuwa wybrane konto użytkownika;
- `usermod` — pozwala na edycję danych konta użytkownika;
- `passwd` — zmienia hasło użytkownika;
- `groupadd` — dodaje nową grupę użytkowników;
- `groupdel` — usuwa grupę użytkowników;
- `groupmod` — modyfikuje grupę użytkowników.

13.6.1. Uprawnienia w systemie Linux

W systemie Linux nadaje się **prawa dostępu do plików i katalogów**. Te prawa to **odczyt** (*read*), **zapis** (*write*), **wykonanie** (*execute*). Mogą być one przydzielone użytkownikowi (właścicielowi), grupie, która jest właścicielem pliku, oraz wszystkim pozostałym użytkownikom.

Poszczególne prawa oznaczane są literami: odczyt — *r*, zapis — *w*, wykonanie — *x*, jak również za pomocą liczb: odczyt — *4*, zapis — *2*, wykonanie — *1*. Sumowanie liczb pozwala na proste przedstawianie uprawnień. Tabela 13.1 prezentuje zapis uprawnień oraz ich wyjaśnienie.

Tabela 13.1. Uprawnienia w postaci liczbowej

Odczyt 4	Zapis 2	Wykonanie 1	
			0 — brak uprawnień
		X	1 — wykonanie
	X		2 — zapis
	X	X	3 — zapis, wykonanie
X			4 — odczyt
X		X	5 — odczyt, wykonanie
X	X		6 — odczyt, zapis
X	X	X	7 — odczyt, zapis, wykonanie

Każdy plik i katalog ma właściciela i grupę właściciela — dla nich oraz pozostałych użytkowników przypisywane są prawa dostępu.

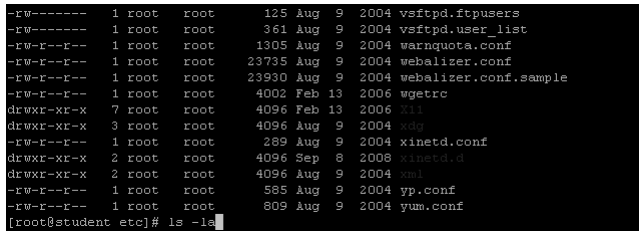
Aby wyświetlić listę plików i katalogów w konsoli tekstowej, należy skorzystać z polecenia `ls`. Opcje tej komendy, które pozwolą wyświetlić więcej informacji, to:

- `-a` — wyświetla wszystkie pliki i katalogi (w tym także ukryte);

- `-l` — wyświetla długą listę informacji o plikach (w tym czas utworzenia i prawa dostępu);
- `-s` — wyświetla rozmiar plików w blokach;
- `-sh` — wyświetla rozmiar plików w bajtach.

Po wydaniu polecenia `ls -la` wyświetlona zostanie lista wszystkich plików i katalogów wraz z prawami dostępu. Rysunek 13.1 pokazuje katalogi znajdujące się w katalogu */etc/*. Kolejne kolumny oznaczają:

- prawa dostępu,
- liczbę dowiązań,
- nazwę właściciela pliku,
- nazwę grupy,
- rozmiar pliku,
- datę modyfikacji,
- nazwę pliku.



Rysunek 13.1.
Wynik działania polecenia ls

Pierwszy znak w pierwszej kolumnie informuje o typie pozycji na liście. Przyjmuje on jedną z następujących postaci:

- *-* oznacza zwykły plik;
- *b* oznacza specjalny plik reprezentujący urządzenie blokowe;
- *c* oznacza specjalny plik reprezentujący urządzenie znakowe;
- *d* oznacza katalog;
- *l* oznacza dowiązanie symboliczne;
- *p* oznacza potok;
- *s* oznacza gniazdo.

Kolejne litery oznaczają prawa dostępu dla właściciela pliku (znaki 2 – 4), grupy (znaki 5 – 7) oraz pozostałych użytkowników (8 – 10). Przykładowy wpis:

```
drwx------ 7 user  users        4096 07-21 13:00 user
```

oznacza katalog (pierwsza litera *d*) z prawami odczytu, zapisu i wykonania dla właściciela (grupa i pozostali użytkownicy nie mają żadnych praw do tego katalogu). Właścicielem jest użytkownik *user*. Grupa, do której należy właściciel pliku, to *users*, katalog

zajmuje na dysku 4 kb. Ostatnia modyfikacja została przeprowadzona 21.07 bieżącego roku o godzinie 13:00. Nazwa katalogu to *user*.

Zmianę praw dostępu do katalogu lub pliku umożliwia polecenie `chmod`. Wymaga ono określenia, czyje uprawnienia należy zmienić, na jakie oraz jakiego pliku lub katalogu ta zmiana będzie dotyczyć. Prawa dostępu mogą być podane zarówno w postaci liczbowej, jak i znakowej. W przypadku postaci liczbowej podaje się trzy kolejne liczby reprezentujące prawa właściciela, grupy i pozostałych użytkowników. Składnia polecenia wygląda następująco: `chmod` *`kod_prawa_dostępu nazwa_pliku_lub_katalogu`*, na przykład:

```
chmod 640 info.txt
```

co oznacza, że plik *info.txt* będzie miał następujące prawa dostępu:

- dla właściciela: odczyt i zapis (4 + 2),
- dla grupy: odczyt (4),
- dla pozostałych użytkowników: brak praw (0).

Gdy prawa dostępu nadawane są w postaci znakowej, składnia wygląda nieco inaczej: `chmod` *`kto_operacja_prawo nazwa_zasobu`*, gdzie: *`kto`* określa, komu nadawane są prawa (`u` — właściciel pliku, `g` — grupa właściciela pliku, `o` — inni użytkownicy, `a` — dla wszystkich), *`operacja`* oznacza przypisanie lub odebranie prawa (`+` lub `-`), *`prawo`* oznacza prawa, które się zmienia (`w` — zapis, `r` — odczyt, `x` — wykonanie). Na przykład polecenie:

```
chmod u-w info.txt
```

oznacza, że właściciel pliku *info.txt* stracił prawo zapisu do pliku.

Zmianę właściciela plików umożliwia polecenie `chown`, ze składnią:
`chown` *`nowy_właściciel nazwa_pliku_lub_katalogu`*, na przykład:

```
chown user katalog1
```

Zmianę grupy pliku umożliwia polecenie `chgrp`, ze składnią:
`chgrp` *`nowa_grupa nazwa_pliku_lub_katalogu`*, na przykład:

```
chgrp users katalog1
```

13.7. Procesy

Podczas uruchamiania każdy proces w systemie ma nadawany unikalny numer **PID** (ang. *Process IDentifier*). Wszystkie procesy w systemie Linux są procesami potomnymi procesu *init*, który ma identyfikator 1. Jest on tworzony podczas startu systemu operacyjnego.

Do każdego procesu przypisany jest użytkownik, który go uruchomił. Na potrzeby usług takich jak serwer WWW czy serwer pocztowy tworzone są specjalne konta, które pozwalają uruchomić wybraną usługę. Usługi w systemach Linux zwane są **demonami** (ang. *daemon*).

DEFINICJA

W systemie Linux wszystkie uruchomione programy noszą miano *procesu*. Jądro systemu steruje procesami i zarządza czasem procesora, przydzielając go kolejnym procesom (wielozadaniowość). System wykonuje dany proces przez pewien czas, a następnie udostępnia procesor kolejnemu procesowi. Procesy mogą przyjmować następujące stany:

- Działający — aktualnie wykonujący jakąś operację.
- Uśpiony — proces czeka na jakieś zdarzenie systemowe, na przykład odczyt danych z dysku.
- Gotowy do wykonania — proces czeka na przydzielenie mu czasu procesora.
- Zombie — proces zakończył działanie, czeka na zakończenie go przez proces macierzysty.

Jako system wielozadaniowy Linux pozwala uruchamiać zadania w tle w trybie tekstowym. Standardowo programy uruchamiane są na pierwszym planie (następuje interakcja z terminalem). Program, który pracuje w tle, nie ma interakcji z terminalem. Aby przenieść uruchomiony program do pracy w tle, należy użyć kombinacji klawiszy *Ctrl+Z*. Kombinacja *Ctrl+C* kończy bieżący proces uruchomiony na pierwszym planie.

Aby zakończyć działające procesy z poziomu systemu operacyjnego — bez konieczności przełączania się między nimi — należy użyć komendy `kill` *`PID`*, gdzie *`PID`* jest identyfikatorem procesu. Zamknięcie wszystkich procesów danego typu (na podstawie ich nazwy, a nie identyfikatora) umożliwia polecenie `killall` *`nazwa_procesu`*.

Aby wyświetlić procesy uruchomione na serwerze, można skorzystać z polecenia `ps`, które bez parametrów wyświetla programy wybranego użytkownika. Opcje tej komendy, które pozwolą wyświetlić więcej informacji, to:

- `-A` — wyświetla wszystkie procesy, także procesy innych użytkowników (rysunek 13.2);
- `-a` — wyświetla wszystkie procesy uruchomione w aktualnym oknie terminala;
- `-l` — wyświetla długą listę informacji o procesach (w tym czas utworzenia i prawa dostępu);
- `-m` — wyświetla informację o pamięci;
- `-u` — wyświetla informację o procesach wybranego użytkownika.

```
11825 ?        00:00:00 smtpd
11848 ?        00:00:00 smtpd
11850 ?        00:00:00 cleanup
11858 ?        00:00:00 pop3-login
11861 ?        00:00:00 pop3-login
11869 ?        00:00:00 pop3-login
11872 ?        00:00:00 smtp
11957 ?        00:00:00 bounce
11958 ?        00:00:00 flush
11959 ?        00:00:00 lmtp
11963 ?        00:00:00 smtpd
11964 ?        00:00:00 pop3-login
11965 pts/1    00:00:00 ps
[root@student etc]# ps -A
```

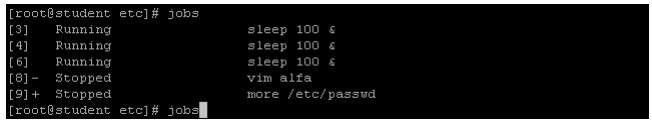
Rysunek 13.2. Wynik działania polecenia ps -A

Aby uruchomić program, który rozpocznie przetwarzanie w tle (np. kompilację kodu źródłowego), na końcu polecenia uruchamiającego należy wpisać znak &, na przykład:

```
gcc program.c &
```

To polecenie uruchomi w tle kompilację kodu źródłowego zapisanego w pliku *program.c*. Użytkownik ujrzy na monitorze informację o numerze uruchomionego procesu (PID). Po zakończeniu kompilacji system wyświetli komunikat o zamknięciu procesu.

Aby sprawdzić zadania wykonywane w tle, należy skorzystać z polecenia `jobs` (rysunek 13.3), które wyświetla numer zadania w tle, nazwę procesu, jego status — działający (ang. *running*), zatrzymany (ang. *stopped*) lub zakończony (ang. *done*).

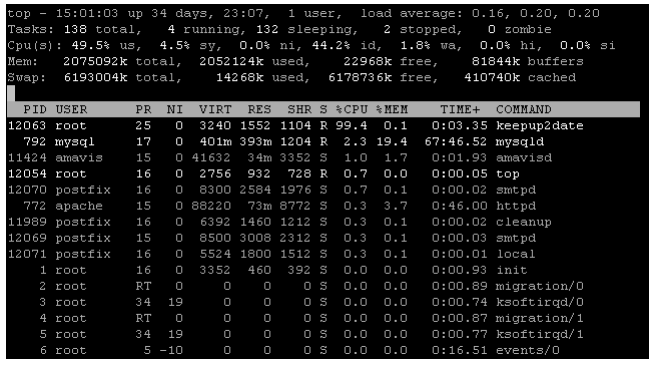


Rysunek 13.3. Wynik działania polecenia jobs

Aby umieścić wybrane polecenie ponownie na pierwszym planie, należy użyć polecenia `fg`, podając jako parametr numer zadania w tle, wyświetlany przez polecenie `jobs`. Polecenie `fg` bez parametru przeniesie na pierwszy plan zadanie, które zostało umieszczone w tle jako ostatnie.

Aby zmienić status zadania wykonywanego w tle, należy użyć komendy `bg` z numerem zadania. Polecenie to powoduje, że status zadania w tle zmieni się z zatrzymanego na działający.

Do zbadania wydajności pracy komputera służy w systemie Linux polecenie `top` (rysunek 13.4). Wyświetla ono w czasie rzeczywistym listę zadań najbardziej obciążających procesor oraz podsumowanie pracy serwera (średnie obciążenie procesora, pamięci itp.).



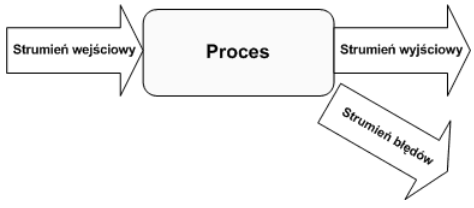
Rysunek 13.4. Wynik działania polecenia top

DEFINICJA

Z każdym procesem w systemie Linux związane jest pojęcie `strumienia`, czyli danych przekazywanych do programu i danych, które generuje dany proces. Zwykle występują trzy strumienie:

- *stdin* — strumień wejściowy domyślnie związany z klawiaturą, z której wprowadzane są dane;
- *stdout* — strumień wyjściowy domyślnie związany z ekranem, na którym wyświetlane są wyniki pracy programu;
- *stderr* — strumień wyjściowy domyślnie związany z ekranem, na którym wyświetlane są błędy generowane przez dany proces.

Zależności pomiędzy strumieniami przedstawione zostały na rysunku 13.5.



Rysunek 13.5.
Przepływy strumieni danych

Jedną z cech systemu Linux jest możliwość **przekierowania strumieni** do plików. Operatory przekierowań to:

- > — przekierowuje strumień wyjściowy do zwykłego pliku podanego jako parametr. Jeśli plik nie istnieje, zostanie utworzony, jeśli istnieje, cała zawartość zostanie zastąpiona.
- >> — przekierowuje strumień wyjściowy do pliku, dopisując dane na końcu pliku.
- < — przekierowuje na strumień wejściowy dane zawarte we wskazanym pliku.

Aby przekierować wyniki pracy wybranego programu do pliku, należy użyć konstrukcji `nazwa_polecenia [parametry] > plik_z_wynikami`, na przykład:

```
ls -la > moje_dane.txt
```

Polecenie to zapisze w pliku *moje_dane.txt* zawartość bieżącego katalogu (wynik działania polecenia `ls -la`). Użycie konstrukcji `nazwa_polecenia [parametry] >> plik_z_wynikami`, na przykład:

```
ls -la >> moje_dane.txt
```

spowoduje, że wyniki działania programu zostaną dopisane na końcu wybranego pliku.

W celu przekierowania strumienia wejściowego używa się konstrukcji `nazwa_polecenia [parametry] < plik_z_danymi`, na przykład:

```
mail uczen1999@wp.pl < informacja.txt
```

Powyższa linijka spowoduje wysłanie zawartości pliku *informacja.txt* na adres pocztowy *uczen1999@wp.pl*.

Kolejnym przykładem funkcjonalności rozwiązań związanych z pracą w trybie tekstowym są **potoki danych**. Są to strumienie wyjściowe jednego procesu przekazywane jako dane wejściowe do innego procesu. W przypadku potoków operatorem pozwalającym na przekazanie jest symbol `|`. Obowiązuje składnia: *`program_pierwszy`* `|` *`program drugi`*, na przykład:

```
ls –la | grep uczen
```

W przytoczonym przykładzie wyniki działania polecenia `ls` zostają przekazane na wejście dla programu grep, który ma za zadanie wypisanie tylko tych linii, w których znajduje się słowo *uczen*. Danymi wejściowymi programu drugiego (`grep`) jest lista plików będąca wynikiem działania programu pierwszego (`ls`).

Podczas przekazywania potoków między procesami bardzo często używana jest wspomniana komenda `grep`. Służy ona do wyświetlania tylko tych linii, które pasują (lub nie pasują) do określonego wzorca. Uproszczona składnia wygląda w sposób następujący: `grep [-v]` *`wzorzec`* `[`*`plik`*`]`, gdzie

- `-v` — oznacza opcję negacji wzorca;
- *`wzorzec`* — oznacza treść do wyszukania;
- *`plik`* — oznacza plik, którego zawartość ma być sprawdzona (gdy nie używamy potoków).

Wzorce tworzone są z wykorzystaniem wyrażeń regularnych. W tabeli 13.2 przedstawiono znaki specjalne, pozwalające na tworzenie dowolnych wyrażeń.

Tabela 13.2. Znaki specjalne wykorzystywane w wyrażeniach regularnych

Znak	Opis
`.`	Dopasuj dowolny znak.
`$`	Dopasuj poprzedzające wyrażenie do końca wiersza.
`^`	Dopasuj występujące po operatorze wyrażenie do początku wiersza.
`*`	Dopasuj zero lub więcej wystąpień znaku poprzedzającego operator.
`[]`	Dopasuj dowolny znak ujęty w nawiasy, na przykład `[abc012]`.
`[ – ]`	Dopasuj dowolny znak z przedziału, na przykład `[0-9]` — wszystkie cyfry; `[a-z]` — wszystkie małe litery; `[0-9a-zA-Z]` — wszystkie litery i cyfry.
`[^ ]`	Dopasuj znak, który nie znajduje się w nawiasach.

Aby lepiej zrozumieć mechanizm tworzenia wyrażeń regularnych, w tabeli 13.3 przedstawione zostały przykłady zastosowań.

Tabela 13.3. Wykorzystanie wyrażeń regularnych w programie grep

Polecenie	Opis
`grep 'Ala' plik`	Wypisze linie zawierające wyraz `Ala`.
`grep 'A.a' plik`	Wypisze linie zawierające wyrazy takie jak `Ala`, `Aga`, `Ara`, `A+a` itp.
`grep 'A[lg]a' plik`	Wypisze linie zawierające wyrazy `Ala` i `Aga`.
`grep '^Ala' plik`	Wypisze linie rozpoczynające się od słowa `Ala`.
`grep 'Go*gle' plik`	Wypisze linie zawierające wyrazy rozpoczynające się na literę `G`, kończące się na `gle`, które między nimi zawierają dowolną liczbę wystąpień litery `o`.

Innym poleceniem wykorzystywanym podczas przekazywania potoków jest `more`, które wyświetla dane z podziałem na strony. Gdy użytkownik zapełni ekran danymi, polecenie to pozwala kontynuować wyświetlanie. Wystarczy, że użytkownik naciśnie klawisz.

ĆWICZENIA

1. Zaloguj się na trzech konsolach systemu Linux.
2. Wyświetl strukturę katalogów znajdujących się w katalogu głównym i w katalogu */dev*.
3. Zapisz w pliku *procesy.txt* aktualnie uruchomione procesy.
4. Sprawdź średnie obciążenie procesora w systemie Linux.
5. Zapisz w pliku *users.txt* zawartość katalogu */home*.
6. Sprawdź prawa dostępu do elementów w katalogu */home*.
7. Dodaj nowego użytkownika do systemu, zmień jego hasło, zaloguj się do systemu jako nowy użytkownik.
8. Jako nowy użytkownik utwórz plik tekstowy zawierający aktualnie uruchomione procesy. Sprawdź, jakie prawa dostępu ma ten plik. Odbierz właścicielowi prawa do odczytu.
9. Zmień grupę, do której należy utworzony plik.
10. Za pomocą poleceń `find` oraz `grep` zapisz w pliku listę wszystkich plików, które w nazwie mają ciąg znaków `al.` .

PYTANIA I POLECENIA KONTROLNE

1. Czym jest wielodostępność?
2. Jak nazywa się system plików wykorzystywany w systemach Linux?
3. Wymień najważniejsze katalogi zapisywane w katalogu głównym systemu Linux. Jakie dane są w nich przechowywane?
4. W jaki sposób system Linux odwołuje się do urządzeń?
5. Wymień powłoki systemowe występujące w systemie Linux. Kiedy są one wczytywane? Gdzie zapisywana jest informacja dotycząca powłoki przypisanej danemu użytkownikowi?
6. Czym jest konsola?
7. W jaki sposób przełączać się między konsolami w systemie Linux?
8. Jakie polecenia pozwalają na podłączenie i zdalną pracę w systemie Linux?
9. Jakie informacje przechowywane są w pliku */etc/passwd*?
10. Co oznaczają skróty PID, UID, GID?
11. Jakie uprawnienia do plików występują w systemie Linux?
12. Jaka komenda pozwala na wyświetlenie plików i katalogów wraz z prawami dostępu do nich?
13. Co to są moduły jądra?
14. Jakie stany może przyjmować proces w systemie Linux?
15. Do czego służy polecenie `top`?
16. Czym różnią się potoki od strumieni?

16

Wprowadzenie do sieci komputerowych

Siecią komputerową nazywa się grupę komputerów lub innych urządzeń połączonych z sobą w celu wymiany danych lub współdzielenia zasobów, na przykład:

- korzystania ze wspólnych urządzeń (skanera, drukarki),
- korzystania ze wspólnego oprogramowania,
- korzystania z centralnej bazy danych,
- przesyłania danych (poczty elektronicznej, plików).

16.1. Rodzaje sieci

Sieci komputerowe mogą być sklasyfikowane w zależności od sposobu dostępu do zasobów oraz ze względu na obszar działania.

DEFINICJA

W zależności od sposobu dostępu do zasobów rozróżnia się dwa rodzaje sieci:

- *klient-serwer* — sieć, w której znajduje się jeden centralny serwer udostępniający dane;
- *peer-to-peer* (sieć równoprawna) — sieć, w której każde urządzenie jest równoprawne z pozostałymi.

Przykładem sieci typu **klient-serwer** może być sieć oparta na usługach katalogowych Active Directory. W sieci znajduje się centralny serwer zarządzający uprawnieniami, który może pracować jako serwer plików (udostępniający dane) lub serwer wydruku (udostępniający drukarki).

Transmisja typu klient-serwer wykorzystywana jest także w przypadku wielu usług w internecie. Dotyczy to na przykład stron WWW umieszczanych na serwerach i pobieranych za pomocą przeglądarki (czyli klienta).

Komputery pracujące w sieci **peer-to-peer** są równorzędne względem siebie, tak jak ma to miejsce w przypadku grupy roboczej w systemie Windows. Każdy komputer pełni zarówno funkcję klienta (pobierając dane z innego urządzenia), jak i serwera (dla innych urządzeń korzystających z udostępnionych zasobów).

DEFINICJA

Ze względu na obszar działania sieci komputerowej rozróżniane są sieci:

- *LAN* (ang. *Local Area Network*) — sieć lokalna działająca na niewielkim, ograniczonym obszarze;
- *MAN* (ang. *Metropolitan Area Network*) — sieć działająca na większym obszarze, na przykład miasta;
- *WAN* (ang. *Wide Area Network*) — sieć rozległa, działająca na dużym obszarze.

Przykładem sieci lokalnej jest sieć obejmująca swoim zasięgiem budynek szkoły. Sieć LAN składa się z komputerów i urządzeń peryferyjnych, mediów transmisyjnych i urządzeń sieciowych. Najczęściej spotyka się sieci zbudowane z wykorzystaniem technologii Ethernet.

Sieciami rozległymi można nazwać sieci dużych firm, obejmujące zasięgiem oddziały na terenie kraju. Pozwalają one na korzystanie z technologii transmisji danych na duże odległości, takich jak Frame Relay, ATM, DSL.

16.2. Podstawowe elementy sieci

Aby sieć mogła funkcjonować, niezbędne są elementy pozwalające na transmisję danych. Zalicza się do nich:

- **karty sieciowe** zapewniające dostęp do sieci;
- **medium transmisyjne** wraz z osprzętem, które przesyła sygnały;
- **protokół komunikacyjny** odpowiedzialny za ich przetwarzanie.

16.2.1. Okablowanie

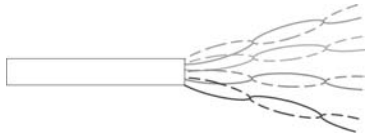
Sieci lokalne początkowo budowane były z wykorzystaniem **kabla koncentrycznego**. Składa się on z miedzianego przewodnika (rdzenia) otoczonego warstwą elastycznej izolacji, która z kolei otoczona jest splecioną miedzianą taśmą lub folią metalową działającą jak drugi przewód i ekran dla znajdującego się wewnątrz przewodnika (rysunek 16.1). Ta druga warstwa zmniejsza także ilość zewnętrznych zakłóceń elektromagnetycznych.

Rysunek 16.1. Budowa kabla koncentrycznego

Kable koncentryczne miały na końcach wtyczki typu BNC. Ten rodzaj okablowania wykorzystywany był w sieciach budowanych w topologii pierścienia lub magistrali, obecnie jednak praktycznie nie jest stosowany.

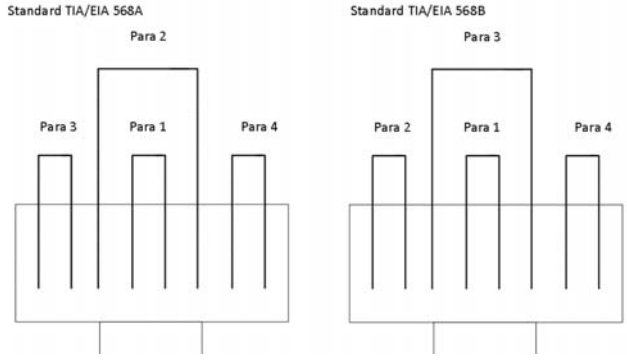
Najpopularniejszym medium transmisyjnym w sieciach lokalnych jest **skrętka nieekranowana** (ang. *Unshielded Twisted Pair — UTP*). Jest to zestaw 4 par żył miedzianych skręconych z sobą. Skręcenie przewodów pozwala na wyeliminowanie zakłóceń elektromagnetycznych. Innym rodzajem skrętki jest **skrętka ekranowana** (ang. *Shielded Twisted Pair — STP*). Podobnie jak UTP, jest ona kablem ośmiożyłowym, dodatkowo owiniętym metalowym oplotem, który pełni funkcję ekranu eliminującego zakłócenia.

W skrętce każda żyła oznaczona jest innym kolorem: zielonym, pomarańczowym, niebieskim, brązowym oraz biało-zielonym, biało-pomarańczowym, biało-niebieskim, biało-brązowym (rysunek 16.2).



Rysunek 16.2.
Kabel typu skrętka

Skrętka podłączana jest do gniazd i końcówek typu RJ-45. Standard Ethernet określa, że każdy ze styków złącza RJ-45 ma specyficzne zadanie. Karta sieciowa wysyła sygnały przez styki 1 i 2, a odbiera na stykach 3 i 6. Przewody w skrętce muszą być podłączone do odpowiednich styków na obu końcach kabla (rysunek 16.3).



Rysunek 16.3. Standardy układu przewodów we wtyczce RJ-45 (TIA/EIA 568A oraz TIA/EIA 568B)

Przy budowie sieci z wykorzystaniem technologii Ethernet stosuje się dwa rodzaje kabli:

- prosty (ang. *straight-through*),
- skrosowany (ang. *crossover*).

Wersja **prosta** służy do łączenia urządzenia końcowego (np. komputera, drukarki, itp.) z koncentratorem lub przełącznikiem. Obie końcówki kabla mają taki sam układ przewodów. Wersja **skrosowana** służy do łączenia komputerów bez pośrednictwa koncentratora lub do łączenia koncentratorów. W tym wypadku na końcach kabli zamienione są miejscami przewody 1 i 2 z 3 i 6.

Standard Ethernet określa maksymalną długość połączenia za pomocą skrętki na 100 m. Po przekroczeniu tej odległości należy użyć aktywnych urządzeń sieciowych w celu wzmocnienia sygnału.

16.2.2. Karta sieciowa

DEFINICJA

Karta sieciowa to urządzenie zapewniające komunikację z siecią komputerową. W komputerach stacjonarnych występuje ona jako karta rozszerzeń lub wbudowana jest na płycie głównej. Dostępne są również karty podłączane przez port USB i porty PCMCIA lub PCI Express dla komputerów przenośnych.

W urządzeniach peryferyjnych pracujących w sieci, takich jak drukarka sieciowa czy skaner, karty sieciowe są wbudowane. Ich konfiguracja najczęściej odbywa się za pomocą aplikacji dostarczanych przez producenta.

Użytkownik, wybierając kartę sieciową, musi uwzględniać architekturę sieci, w której będzie ona pracowała. Najczęściej spotykane są karty sieciowe pracujące w standardzie Fast-Ethernet.

Aby podłączyć urządzenie do sieci bezprzewodowej, również wymagana jest karta sieciowa, która jako medium transmisyjne wykorzystuje fale radiowe. Do komunikacji bezprzewodowej wykorzystuje się pasmo 2,4 GHz (w standardzie 802.11b oraz 802.11g) lub 5 GHz (w standardzie 802.11a). Pasmo wykorzystywane do transmisji determinuje wybór karty sieciowej.

Każda karta sieciowa ma zapisany przez producenta unikalny adres zwany **MAC** (od ang. *Media Access Control*). Adres ten wykorzystywany jest podczas transmisji w drugiej warstwie modelu OSI (więcej na ten temat w podrozdziale 16.4). Adres MAC jest adresem 48-bitowym, zapisywanym w postaci 12 liczb szesnastkowych oddzielanych znakiem - lub :.

16.2.3. Wzmacniak, koncentrator, przełącznik

W sieci zbudowanej za pomocą skrętki maksymalna dopuszczalna długość jednego segmentu wynosi 100 m. Wraz z odległością transmitowany sygnał traci moc. Aby zachować jego parametry, należy go wzmocnić poprzez zastosowanie urządzeń aktywnych. Jednym z takich urządzeń jest **wzmacniak** (ang. *repeater*), który powoduje wzmocnienie transmitowanego sygnału (wchodzącego) i przekazanie go dalej (na wyjście). W sieciach budowanych z wykorzystaniem kabla UTP wzmacniak pozwala na przedłużenie odcinka sieci o kolejne 100 m. Inne spotykane nazwy wzmacniaka to regenerator oraz wtórnik.

Koncentrator (ang. *hub*) to urządzenie łączące wiele urządzeń pracujących w sieci komputerowej w topologii gwiazdy. Okablowanie poprowadzone od poszczególnych urządzeń schodzi się w centralnym miejscu sieci, które stanowi koncentrator. Pracuje on w pierwszej warstwie modelu OSI (więcej o modelu OSI w podrozdziale 16.4). Jego zadaniem jest wzmocnienie sygnału przychodzącego i przekazanie go na wszystkie wyjścia.

Przełącznik (ang. *switch*), podobnie jak koncentrator, stanowi centralny punkt sieci zbudowanej w topologii gwiazdy, jednak sygnał wchodzący nie jest przesyłany na wszystkie wyjścia, lecz tylko do portu, do którego podłączone jest urządzenie odbierające dane. Przełącznik (rysunek 16.4) pracuje w warstwie drugiej modelu OSI, przełączanie ramek realizowane jest na podstawie adresów MAC urządzeń podłączonych do sieci.

Tablica adresów MAC tworzona jest dynamicznie podczas pracy urządzenia. Jeśli dane transmitowane są do urządzenia o nieznanym adresie, to przesyłane są one na wszystkie wyjścia w urządzeniu.

Rysunek 16.4. Przełącznik (ang. *switch*)

Wygląd zewnętrzny koncentratorów i przełączników jest bardzo podobny, najważniejsza różnica tkwi w mechanizmie propagacji płynących sygnałów.

16.2.4. Router

DEFINICJA

Router to urządzenie pracujące w warstwie trzeciej modelu OSI, bazujące na adresach IP. Routery łączą różne rodzaje sieci, pozwalają na przekazywanie pakietów między oddzielnymi sieciami logicznymi (sieciami IP), a także między sieciami zbudowanymi z wykorzystaniem różnych mediów i technologii transmisyjnych. Routery kierują pakiety do sieci docelowej, wybierając najlepszą dla nich drogę. Operacja ta nazywana jest *rutowaniem* lub *trasowaniem*.

Funkcję routera mogą pełnić komputery z dwoma kartami sieciowymi (podłączonymi do dwóch sieci) oraz z oprogramowaniem zapewniającym kierowanie pakietów do odpowiednich sieci.

Router jest urządzeniem niezbędnym do podłączenia sieci lokalnej do internetu. Rozwój technologii i dostępność łączy internetowych to przyczyny upowszechniania się niewielkich routerów domowych. Najczęściej mają one jedno łącze do sieci WAN oraz wbudowany przełącznik służący do podłączenia kilku urządzeń sieci lokalnej. Urządzenia te pełnią także dodatkowe funkcje, takie jak translacja adresów oraz serwer DHCP.

16.2.5. Punkt dostępowy sieci bezprzewodowej

DEFINICJA

Bezprzewodowa sieć lokalna (ang. *Wireless Local Area Network — WLAN*) to sieć lokalna, w której połączenia między urządzeniami sieciowymi zrealizowano bez użycia przewodów. Na infrastrukturę sieci bezprzewodowej składają się:

- karty sieciowe,
- punkty dostępowe,
- anteny wraz z okablowaniem.

Sieci WLAN pracować mogą w dwóch trybach:

- w trybie ad-hoc, w którym urządzenia łączą się bezpośrednio z sobą;
- w trybie infrastruktury, z wykorzystaniem punktów dostępowych (ang. *access point*).

Punkt dostępowy to centralny punkt sieci bezprzewodowej. Przekazuje dane między urządzeniami, pozwala także na podłączenie sieci bezprzewodowej do sieci kablowej. Punkty dostępowe mają dwa interfejsy sieciowe: interfejs bezprzewodowy (gniazdo do podłączenia anteny) i interfejs sieci kablowej (najczęściej gniazdo RJ-45 do podłączenia sieci Ethernet).

Punkty dostępowe mogą komunikować się z sobą, co pozwala na budowę złożonej infrastruktury łączącej urządzenia znacznie od siebie oddalone.

W przypadku sieci bezprzewodowych ogromne znaczenie ma bezpieczeństwo danych. Ogólnodostępne medium transmisyjne powoduje, że każde urządzenie znajdujące się w zasięgu sieci mogłoby korzystać z jej zasobów. Punkty dostępowe pozwalają na implementację procedur bezpieczeństwa polegających na filtrowaniu adresów MAC lub IP, a także na zabezpieczeniu dostępu do sieci kluczem szyfrującym. Dostępne protokoły szyfrowania dla sieci bezprzewodowych to:

- **WEP** (ang. *Wired Equivalent Privacy*) — opiera się na kluczu długości 40 lub 104 bitów.
- **WPA** oraz **WPA2** (ang. *WiFi Protected Access*) — opierają się na 128-bitowym kluczu, pozwalają na lepsze zarządzanie kluczami.

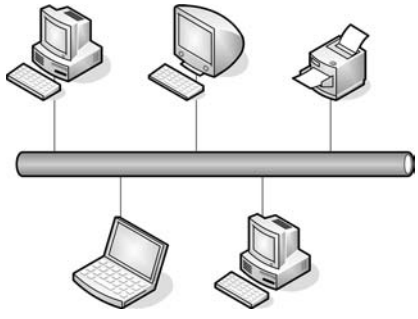
16.3. Topologie sieciowe

Topologie sieci lokalnych mogą być opisane zarówno na płaszczyźnie fizycznej, jak i logicznej. **Topologia fizyczna** określa organizację okablowania, **topologia logiczna** opisuje dostęp do medium fizycznego oraz reguły komunikacji, z których korzystają podłączone do sieci urządzenia. Obie płaszczyzny topologii są ściśle z sobą powiązane.

16.3.1. Topologia magistrali

DEFINICJA

Topologia magistrali (ang. *bus*) charakteryzuje się tym, że wszystkie elementy sieci są podłączone do jednej magistrali (zazwyczaj kabla koncentrycznego). Sieć umożliwia tylko jedną transmisję w danym momencie — sygnał nadany przez jedną ze stacji jest odbierany przez wszystkie z nich, lecz tylko adresat go interpretuje (rysunek 16.5).



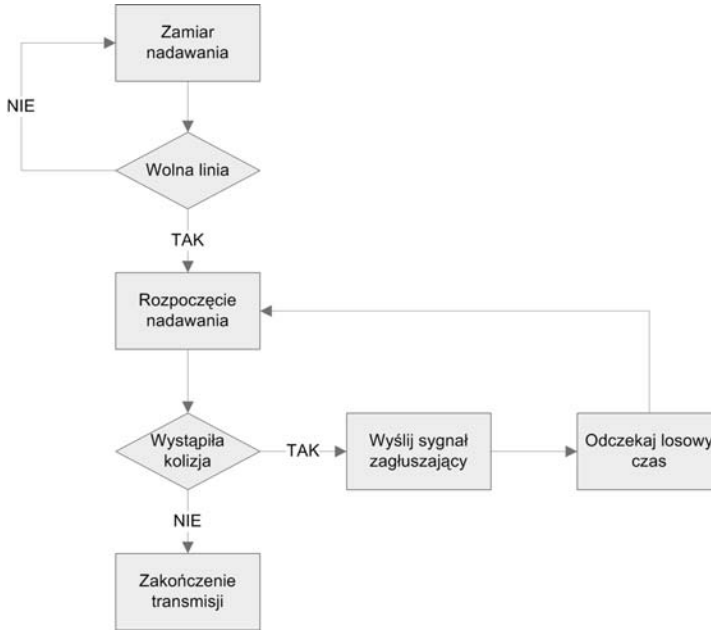
Rysunek 16.5.
Topologia magistrali

Końce magistrali wyposażone są w tzw. terminatory, których zadaniem jest wyeliminowanie odbicia sygnału od końca kabla. Odbicia te mogą zakłócać pracę sieci.

Dostęp do medium transmisyjnego realizowany jest przez protokół CSMA/CD (ang. *Carrier Sense Multiple Access / Collision Detection*). Protokół ten wykrywa, czy łącze jest dostępne, a także reaguje na występujące kolizje.

W sieci z protokołem CSMA/CD urządzenia przed nadawaniem sprawdzają, czy medium sieciowe nie jest zajęte. Jeśli węzeł wykryje, że sieć jest zajęta, będzie oczekiwał przez losowo wybrany czas, zanim ponowi próbę. Jeśli węzeł wykryje, że medium nie jest zajęte, rozpocznie nadawanie i nasłuchiwanie. Celem nasłuchiwania jest upewnienie się, że żadna inna stacja nie nadaje w tym samym czasie. Po zakończeniu transmisji danych urządzenie powróci do trybu nasłuchiwania.

Jeśli dwa urządzenia rozpoczęły nadawanie w tym samym czasie, występuje **kolizja** wykrywana przez urządzenia nadawcze. Transmisja danych zostaje wówczas przerwana. Węzły zatrzymują nadawanie na losowo wybrany czas, po którym podejmowana jest kolejna próba uzyskania dostępu do medium. Rysunek 16.6 przedstawia algorytm mechanizmu CSMA/CD.



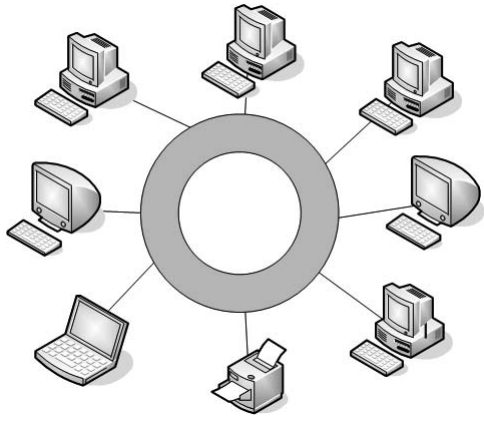
Rysunek 16.6. Algorytm blokowy działania mechanizmu CSMA/CD

Zalety sieci opartych na topologii magistrali to małe zużycie kabla, brak dodatkowych urządzeń (np. koncentratorów), niska cena sieci (pojedynczy kabel między węzłami) i łatwość instalacji. Wadą jest trudność w lokalizacji usterek, możliwość tylko jednej transmisji w danym momencie, potencjalnie duża liczba kolizji oraz fakt, że awaria głównego kabla lub rozpięcie dowolnego złącza powoduje unieruchomienie całej sieci.

16.3.2. Topologia pierścienia

DEFINICJA

W *topologii pierścienia* (ang. *ring*) wszystkie urządzenia połączone są za pomocą jednego nośnika w układzie zamkniętym — okablowanie tworzy krąg i nie ma żadnych zakończeń (rysunek 16.7). Sygnał wędruje w pętli między komputerami. Każdy komputer pełni funkcję wzmacniacza regenerującego sygnał i wysyłającego go dalej. Sieć w topologii pierścienia tworzona jest za pomocą kabla koncentrycznego lub światłowodu.



Rysunek 16.7.
Topologia pierścienia

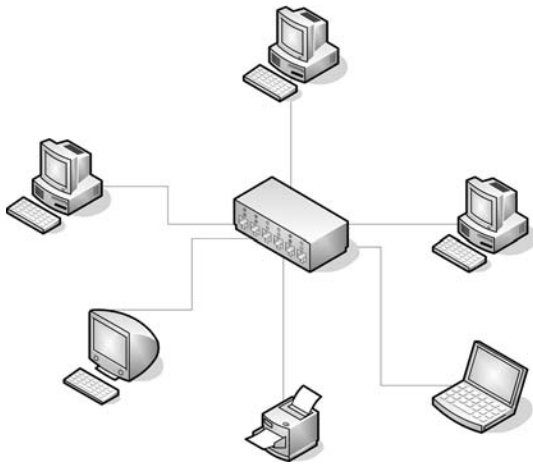
Dostęp do sieci w topologii pierścienia realizowany jest przez przekazywanie żetonu. **Żeton** (ang. *token*) dostępu jest określoną sekwencją bitów zawierających informację kontrolną. Przejęcie żetonu przez urządzenie sieciowe zezwala na rozpoczęcie transmisji danych. Każda sieć (pierścień) ma tylko jeden żeton dostępu, przekazywany między kolejnymi węzłami sieci. Jeśli komputer ma dane do wysłania, to usuwa żeton z pierścienia i rozpoczyna transmisję. Dane wędrują po kolejnych węzłach sieci, aż trafią do adresata. Komputer odbierający wysyła komunikat do komputera nadającego o odebraniu danych. Po weryfikacji komputer wysyłający tworzy nowy żeton dostępu i wysyła go do sieci.

Zaletą sieci w topologii pierścienia jest małe zużycie przewodów. Wady to łatwość uszkodzenia sieci (uszkodzenie jednego węzła powoduje zatrzymanie transmisji w całej sieci), kłopoty z lokalizacją uszkodzeń, a także utrudniona rozbudowa sieci.

16.3.3. Topologia gwiazdy

DEFINICJA

Topologia gwiazdy (ang. *star*) charakteryzuje się tym, że kable sieciowe połączone są w jednym wspólnym punkcie sieci, w którym znajduje się koncentrator lub przełącznik (rysunek 16.8).



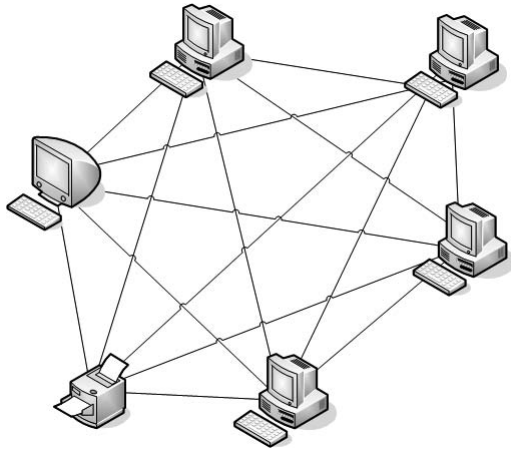
Rysunek 16.8.
Topologia gwiazdy

Zaletami sieci zbudowanej w topologii gwiazdy są duża przepustowość i łatwość diagnozowania uszkodzeń, wadą jest nadmierne zużycie kabli. Topologia gwiazdy jest obecnie najczęściej stosowana przy budowie sieci lokalnych. Opiera się na niej topologia **rozszerzonej gwiazdy,** w której pojedyncze gwiazdy są powiązane za pomocą koncentratorów lub przełączników. Takie rozwiązanie umożliwia rozszerzenie zasięgu i obszaru sieci.

16.3.4. Topologia siatki i topologia mieszana

DEFINICJA

Topologia siatki polega na zapewnieniu wszystkim urządzeniom połączeń ze wszystkimi pozostałymi urządzeniami w sieci (rysunek 16.9). Oznacza to, że każdy host ma własne połączenia z pozostałymi hostami.

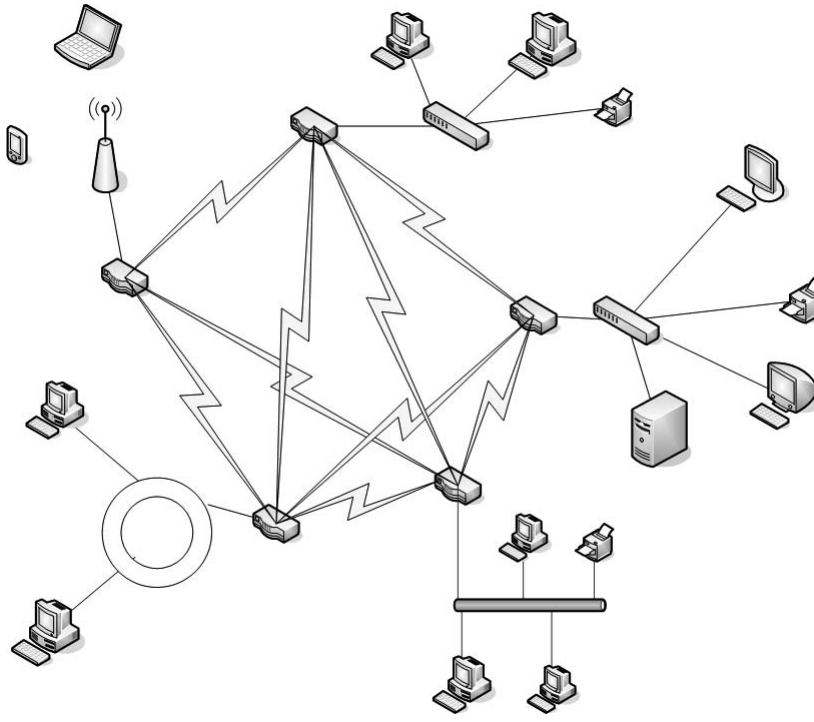


Rysunek 16.9.
Topologia siatki

DEFINICJA

Topologia mieszana łączy w sobie różne rozwiązania — jest połączeniem co najmniej dwóch innych topologii (rysunek 16.10).

Zaletami topologii siatki są niezawodność, brak kolizji i możliwość przesyłu danych wieloma ścieżkami. Wadami są wysokie koszty oraz złożoność budowy.



Rysunek 16.10. Topologia mieszana

16.4. Model OSI

DEFINICJA

Model odniesienia OSI (ang. *Open System Interconnection Reference Model*) jest traktowany jako wzorzec transmisji danych dla protokołów komunikacyjnych w sieciach komputerowych. Podstawowym założeniem modelu jest podział systemów sieciowych na 7 warstw (ang. *layers*), współpracujących z sobą w ściśle określony sposób (rysunek 16.11). Został on przyjęty przez Międzynarodową Organizację Standaryzującą ISO w 1984 roku.

Rysunek 16.11.
Model warstwowy OSI

Model odniesienia OSI jest wzorcem używanym do reprezentowania mechanizmów przesyłania informacji w sieci. Pozwala wyjaśnić, w jaki sposób dane pokonują różne warstwy w drodze do innego urządzenia w sieci, nawet jeśli nadawca i odbiorca dysponują różnymi typami medium sieciowego. Podział sieci na warstwy przynosi następujące korzyści:

- Dzieli proces komunikacji sieciowej na mniejsze, łatwiejsze do zarządzania elementy składowe.
- Tworzy standardy składników sieci, dzięki czemu składniki te mogą być rozwijane niezależnie i obsługiwane przez różnych producentów.
- Umożliwia wzajemną komunikację sprzętu i oprogramowania sieciowego różnych rodzajów.
- Zmiany wprowadzone w jednej warstwie nie dotyczą innych warstw.

Trzy górne warstwy, czyli aplikacji, prezentacji i sesji, zajmują się współpracą z oprogramowaniem wykonującym zadania zlecane przez użytkownika systemu komputerowego. Tworzą one interfejs, który pozwala na komunikację z warstwami niższymi.

Warstwa aplikacji (ang. *application layer*) zapewnia dostęp do sieci aplikacjom użytkownika. W warstwie tej zdefiniowane są protokoły usług sieciowych takich jak HTTP, FTP, SMTP.

Warstwa prezentacji (ang. *presentation layer*) odpowiada za reprezentację danych — obsługę znaków narodowych, formatów graficznych oraz kompresję i szyfrowanie.

Warstwa sesji (ang. *session layer*) zapewnia aplikacjom komunikację między różnymi systemami. Zarządza sesjami transmisyjnymi poprzez nawiązywanie i zrywanie połączeń między aplikacjami.

Warstwa transportowa (ang. *transport layer*) zapewnia połączenie między aplikacjami na różnych systemach komputerowych, dba o kontrolę poprawności przesyłanych danych. Tutaj następuje podział danych na segmenty, które są kolejno numerowane i wysyłane do stacji docelowej. Stacja docelowa po odebraniu segmentu wysyła po-

twierdzenie odbioru. Jeśli porcja danych nie dotarła do adresata, przeprowadzana jest retransmisja.

Warstwa sieciowa (ang. *network layer*) zapewnia metody ustanawiania, utrzymywania i rozłączania połączeń sieciowych. W tej warstwie obsługiwane są routing i adresacja logiczna.

Poprawną transmisję danych przez konkretne media transmisyjne zapewnia warstwa **łącza danych** (ang. *data link*). Warstwa ta operuje na fizycznych adresach interfejsów sieciowych, zapewniając łączność między dwoma bezpośrednio połączonymi urządzeniami.

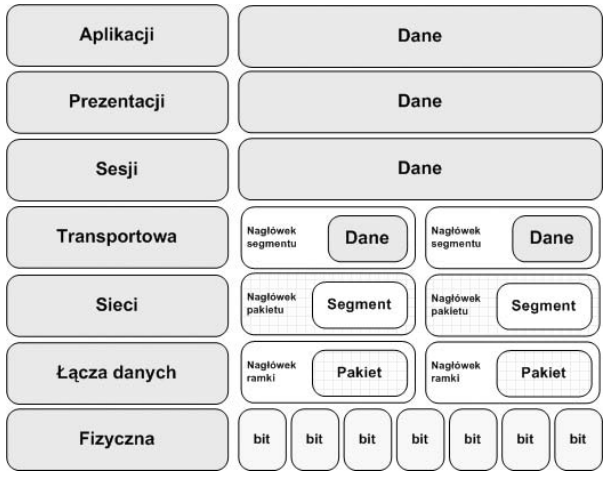
Warstwa fizyczna odbiera dane z warstwy łącza danych i przesyła je w medium transmisyjnym jako bity reprezentowane w konkretny sposób (sygnały elektryczne, impulsy świetlne).

W pierwszej warstwie modelu OSI pracują koncentratory, wzmacniaki i okablowanie. Do drugiej przypisane są przełączniki i karty sieciowe (operują na adresach MAC); warstwa trzecia poprzez routery operuje na adresach IP.

Model OSI opisuje drogę danych przesyłanych między aplikacjami, które zostały uruchomione na różnych systemach komputerowych. W przypadku większości usług w internecie transmisja między systemami realizowana jest według modelu klient-serwer, a komunikują się aplikacje klienckie (np. przeglądarka internetowa) i aplikacja serwerowa (np. serwer stron WWW).

DEFINICJA

Transmisja w modelu OSI przeprowadzana jest w dół kolejnych warstw (na komputerze klienckim), a następnie w górę (na serwerze). Proces przekazywania danych między warstwami protokołu nazywany jest *enkapsulacją* (kapsułkowaniem, rysunek 16.12).



Rysunek 16.12. Model enkapsulacji

W procesie enkapsulacji dane użytkownika (z warstwy aplikacji) dzielone są w warstwie transportu na **segmenty** i opatrywane nagłówkiem zawierającym m.in. numery portów. Tak przygotowane porcje danych wędrują do warstwy trzeciej, gdzie dodawany jest nagłówek zawierający adresy logiczne nadawcy i odbiorcy. Powstaje **pakiet**. Do pakietów w warstwie łącza danych dodawane są adresy fizyczne — tworzona jest **ramka**. Ostatnia warstwa — fizyczna — ramkę z poprzedniej warstwy przekształca do postaci pozwalającej przesłać informację medium transmisyjnym. Dane wędrują do stacji docelowej i tam są ponownie przekształcane, najpierw z bitów na ramki, następnie na pakiety i segmenty, po czym zostają zinterpretowane przez aplikację na komputerze docelowym.

16.5. Protokoły używane w sieciach LAN

16.5.1. Protokół TCP/IP

Najpopularniejszym spośród protokołów komunikacyjnych jest **protokół IP**, powszechnie używany w sieciach LAN, a także w internecie. W sieciach IP dane są wysyłane w formie bloków określanych mianem pakietów. W przypadku transmisji z wykorzystaniem protokołu IP przed jej rozpoczęciem nie jest zestawiana wirtualna sesja komunikacyjna między dwoma urządzeniami.

Protokół IP jest protokołem zawodnym — nie gwarantuje, że pakiety dotrą do adresata, że nie zostaną pofragmentowane czy też zdublowane. Ponadto dane mogą dotrzeć do odbiorcy w innej kolejności niż zostały nadane. Niezawodność transmisji danych zapewniają protokoły warstw wyższych (np. protokół warstwy transportowej — TCP).

16.5.2. Protokół IPX/SPX

Dla sieci pracujących w środowisku Novell Netware opracowany został protokół **IPX** (ang. *Internet Packet Exchange*). Nie został on wyposażony w mechanizmy kontroli transmisji i nie gwarantuje, że wszystkie pakiety dotrą na miejsce. Podobnie jak w przypadku protokołu IP, niezawodność transmisji zapewnia protokół warstwy czwartej — **SPX** (ang. *Sequenced Packet Exchange*).

Adresacja w protokole IPX składa się z dwóch części: adresu sieci i adresu hosta. Pierwszy z nich jest liczbą 32-bitową, drugi — 48-bitową i odpowiada adresowi MAC karty sieciowej.

Obecnie protokoły IPX/SPX praktycznie nie są stosowane, ponieważ zostały wyparte przez stos protokołów TCP/IP.

16.5.3. AppleTalk

AppleTalk jest protokołem opracowanym przez firmę Apple, stosowanym w sieciach komputerowych opartych na systemie operacyjnym MacOS. Protokół ten wykorzystują proste sieci równorzędne. Aktualnie protokół AppleTalk nie jest rozwijany, został zastąpiony przez protokół TCP/IP.

Protokoły IP, IPX i AppleTalk są **protokołami rutowalnymi** (ang. *routed protocol*). Oznacza to, że mogą być obsługiwane przez routery, a więc mogą przenosić dane między różnymi sieciami.

16.5.4. NetBEUI

NetBEUI to prosty protokół opracowany przez IBM i wykorzystywany jedynie w systemach operacyjnych firmy Microsoft. Protokół ten cechuje się minimalnymi wymaganiami i dużą odpornością na błędy. Sprawdza się jednak tylko w małych sieciach lokalnych — nie może być używany w internecie, gdyż nie jest protokołem rutowalnym. W najnowszych wersjach systemów Windows protokół został zastąpiony przez TCP/IP.

W sieciach LAN wyróżnia się trzy rodzaje transmisji:

- **Unicast** — pojedynczy pakiet wysyłany jest przez stację nadawczą do odbiorcy.
- **Multicast** (**transmisja grupowa**) — pojedynczy pakiet danych jest kopiowany i wysyłany do grupy stacji sieciowych (określonej przez adres multicast).
- **Broadcast** (**transmisja rozgłoszeniowa**) — pojedynczy pakiet jest kopiowany i wysyłany do wszystkich stacji sieciowych. W tym typie transmisji stacja nadawcza adresuje pakiet, używając adresu broadcast.

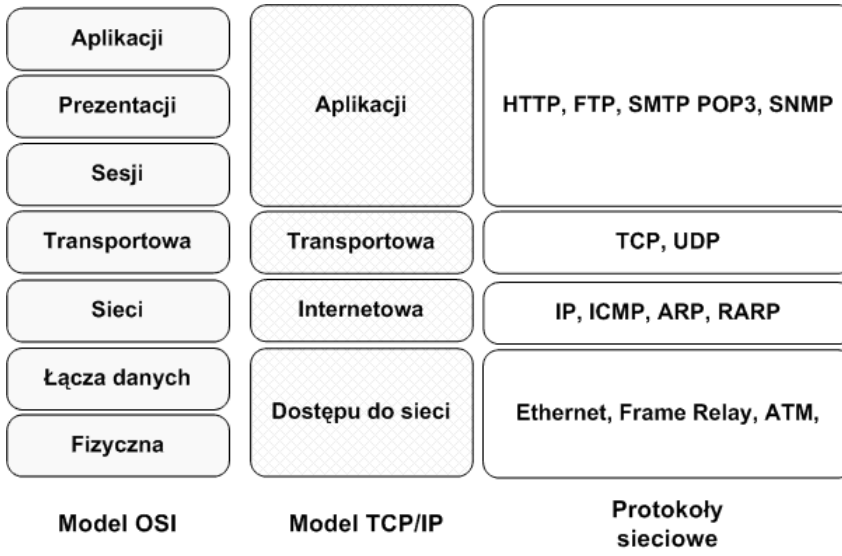
16.6. Model TCP/IP

DEFINICJA

Model TCP/IP (ang. *Transmission Control Protocol/Internet Protocol*) to teoretyczny model warstwowej struktury komunikacji sieciowej. Podstawową cechą modelu TCP/IP jest wyodrębnienie szeregu współpracujących z sobą warstw (ang. *layers*). Założenia modelu TCP/IP są zbliżone do założeń modelu OSI, jednak liczba warstw jest mniejsza i bardziej odzwierciedla prawdziwą strukturę internetu (rysunek 16.13).

Warstwa aplikacji (ang. *application layer*) to najwyższy poziom, w którym pracują aplikacje, na przykład serwer WWW czy przeglądarka internetowa. Obejmuje ona zestaw gotowych protokołów, które są wykorzystywane do przesyłania w sieci różnego typu informacji.

Warstwa transportowa (ang. *transport layer*) odpowiada za przesyłanie danych i kieruje właściwe informacje do odpowiednich aplikacji, wykorzystując porty określane dla każdego połączenia. Warstwa transportowa nawiązuje i zrywa połączenia między komputerami i gwarantuje pewność transmisji.



Rysunek 16.13. Porównanie modeli OSI i TCP/IP

Zadaniem **warstwy internetowej** (ang. *internet layer*) jest podzielenie segmentów na pakiety i przesłanie ich dowolną siecią. Pakiety trafiają do sieci docelowej niezależnie od przebytej drogi. Tą warstwą zarządza protokół IP. Tutaj określana jest najlepsza ścieżka i następuje przełączanie pakietów.

UWAGA

Związek między protokołem IP i protokołem TCP jest bardzo istotny. Protokół IP określa drogę dla pakietów, a protokół TCP zapewnia niezawodny transport.

Warstwa dostępu do sieci (ang. *network access layer*) zajmuje się przekazywaniem danych przez fizyczne połączenia między urządzeniami sieciowymi (np. karty sieciowe lub modemy). Dodatkowo warstwa ta jest wyposażona w protokoły służące do dynamicznego określania adresów IP.

16.6.1. Protokoły w warstwie dostępu do sieci

Warstwa dostępu do sieci jest odpowiedzialna za wszystkie zagadnienia związane z zestawieniem łącza fizycznego służącego do przekazywania pakietu IP do medium sieciowego. Odpowiada między innymi za odwzorowywanie adresów IP na adresy sprzętowe i za enkapsulację pakietów IP w ramki. Określa połączenie z fizycznym medium sieci w zależności od rodzaju sprzętu i interfejsu sieciowego.

W warstwie dostępu do sieci modelu TCP/IP działają sterowniki aplikacji, modemów i innych urządzeń. Definiuje ona funkcje umożliwiające korzystanie ze sprzętu siecio-

wego i dostęp do medium transmisyjnego. W sieciach lokalnych protokołem dostępu do sieci jest Ethernet, w sieciach rozległych są to m.in. protokoły ATM i Frame Relay.

Ethernet

DEFINICJA

Standard Ethernet został opublikowany w latach 80. ubiegłego wieku. Transmisja osiągała szybkość do 10 Mb/s i była realizowana przez gruby kabel koncentryczny na odległościach do 2 km. Pierwotny standard technologii Ethernet był wielokrotnie poprawiany w celu dostosowania go do potrzeb nowych mediów transmisyjnych i większych prędkości transmisji. Obecnie rodzina technologii Ethernet obejmuje trzy standardy: Ethernet (prędkość 10 Mb/s), Fast Ethernet (100 Mb/s) i Gigabit Ethernet (1000 Mb/s).

Technologie Ethernet określają sposoby ustalania przepustowości łącza sieciowego nazywane **autonegocjacją**. Interfejsy sieciowe mogą pracować w wielu trybach, w zależności od rodzaju wykorzystywanego w sieci medium. Celem autonegocjacji jest umożliwienie współpracy różnych urządzeń w trybie o najwyższej prędkości akceptowalnej przez wszystkie urządzenia w sieci.

Format ramki przyjmuje postać przedstawioną na rysunku 16.14.

Preambuła	SFD	Adres docelowy MAC	Adres źródłowy MAC	Typ ramki	Dane	Suma kontrolna

Rysunek 16.14. Ramka Ethernet

Poszczególne elementy oznaczają:

- **Preambuła** — składa się z 7 bajtów złożonych z naprzemiennych jedynek i zer.
- **SFD** (ang. *start frame delimiter*), czyli znacznik początkowy ramki w postaci sekwencji 8 bitów (1 bajt).
- **Adres MAC odbiorcy** (6 bajtów).
- **Adres MAC nadawcy** (6 bajtów).
- **Typ ramki** (2 bajty).
- **Dane** (46 – 1500 bajtów) — jeżeli dane są mniejsze niż 46 bajtów, to uzupełniane są zerami.
- **Suma kontrolna** (4 bajty).

Frame Relay

Frame Relay zapewnia komunikację połączeniową o przepływności do 45 Mb/s. Funkcjonuje na telekomunikacyjnych łączach cyfrowych dobrej jakości, odznaczających się niskim wskaźnikiem błędów. Frame Relay pozwala na łączenie sieci LAN, transmisję danych i głosu, wideo- i telekonferencje.

Sieć Frame Relay składa się z wielu urządzeń sieciowych połączonych kanałami fizycznymi, na których tworzone są połączenia wirtualne (logiczne). Mogą być one zestawiane na stałe (ang. *Permanent Virtual Circuits — PVC*) i tymczasowo (ang. *Switched Virtual Circuits — SVC*).

Frame Relay zapewnia gwarantowaną szybkość transmisji (ang. *Committed Information Rate — CIR*).

ATM

ATM jest technologią telekomunikacyjną, która umożliwia przesyłanie głosu, obrazów wideo i danych przez sieci prywatne i publiczne. W przeciwieństwie do Frame Relay, jest ona oparta na architekturze komórek, a nie ramek. Komórka ATM ma stałą długość 53 bajtów. Tworzy ją 5-bajtowy nagłówek ATM i 48 bajtów treści zasadniczej. Małe komórki o stałej długości doskonale nadają się do przesyłania głosu i obrazów wideo, ponieważ ruch ten nie toleruje opóźnień. Ruch zawierający obrazy wideo i głos nie musi czekać na przesłanie większego pakietu danych.

16.6.2. Protokoły warstwy internetowej

Zadaniem warstwy internetowej jest wybranie najlepszej ścieżki dla pakietów przesyłanych w sieci. Podstawowym protokołem działającym w tej warstwie jest **protokół IP** (ang. *Internet Protocol*). Tutaj następuje określenie najlepszej ścieżki i przełączanie pakietów.

DEFINICJA

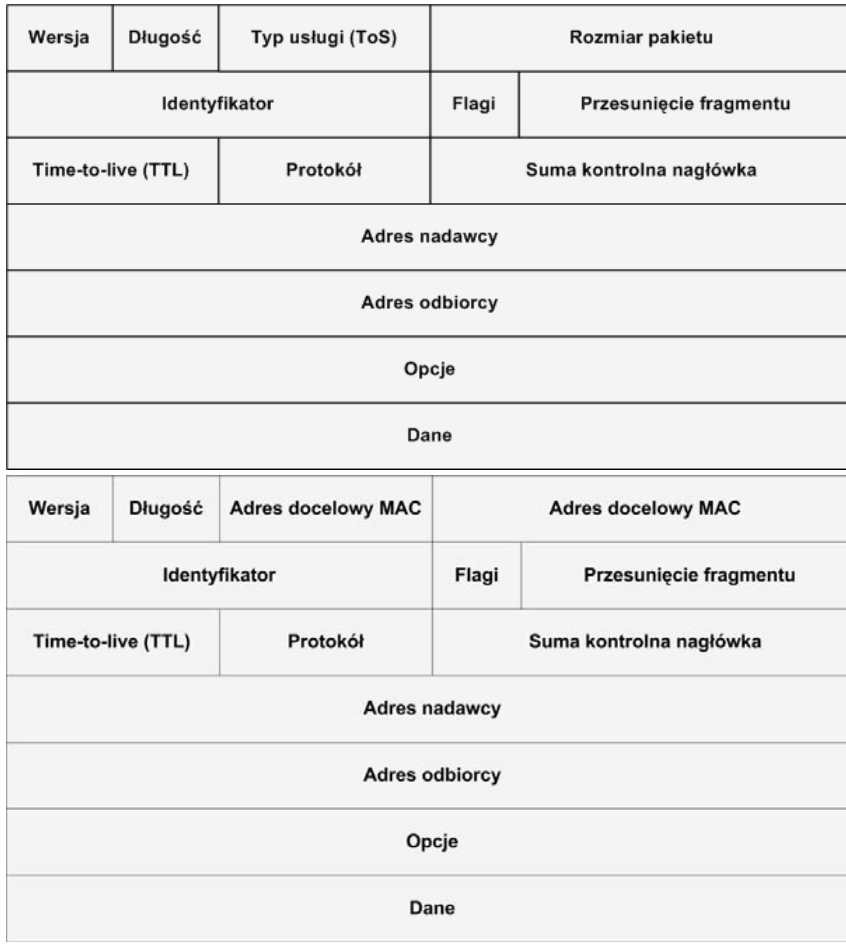
W warstwie internetowej modelu TCP/IP działają następujące protokoły:

- *Protokół IP*, który zapewnia usługę bezpołączeniowego dostarczania pakietów przy użyciu dostępnych możliwości. Protokół IP nie bierze pod uwagę zawartości pakietu, ale wyszukuje ścieżkę do miejsca docelowego.
- *Protokół ICMP* (ang. *Internet Control Message Protocol*), który pełni funkcje kontrolne i informacyjne. Jest on używany przez polecenia sprawdzające poprawność połączenia (np. polecenie `ping`).
- *Protokół ARP* (ang. *Address Resolution Protocol*), który znajduje adres warstwy łącza danych MAC dla znanego adresu IP.
- *Protokół RARP* (ang. *Reverse Address Resolution Protocol*), który znajduje adres IP dla znanego adresu MAC.

Protokół IP spełnia następujące funkcje:

- Definiuje format pakietu i schemat adresowania.
- Przesyła dane między warstwą internetową i warstwą dostępu do sieci.
- Kieruje pakiety do zdalnych hostów.

Postać, w jakiej dane są przesyłane przez pakiety IP, przedstawiona została na rysunku 16.15.



Rysunek 16.15. Format pakietu IP

Poszczególne elementy oznaczają:

- **Wersja** — wersja protokołu IP.
- **Długość nagłówka** — wielkość nagłówka datagramu opisanego w 32-bitowych słowach.
- **Typ usługi** (ang. *Type of Service — ToS*) — określa klasę usług; wykorzystywany przy zarządzaniu ruchem.
- **Rozmiar pakietu** — rozmiar całego pakietu IP podany w bajtach.
- **Identyfikator** — używany podczas łączenia fragmentów danych.

- **Flagi** — jest to 3-bitowe pole, gdzie pierwszy bit oznacza, czy dany pakiet może zostać podzielony na fragmenty; drugi — czy pakiet jest ostatnim fragmentem. Trzeci bit nie jest używany.
- **Przesunięcie fragmentu** — określa pozycję, gdzie w bieżącym pakiecie kończy się fragment datagramu z poziomu 4.
- **Time-To-Live (TTL)** — zawiera znacznik życia pakietu. Pole to jest liczbą, zmniejszaną przez każdy router, przez który przechodzi. Kiedy wartość TTL osiągnie zero, pakiet jest zatrzymywany, a nadawca zostaje poinformowany, że pakietu nie udało się dostarczyć.
- **Protokół** — oznacza kod protokołu warstwy wyższej — transportowej.
- **Suma kontrolna nagłówka** — służy do wykrywania uszkodzeń wewnątrz pakietów.
- **Adresy źródłowy i docelowy pakietu** — adres IP nadawcy i odbiorcy pakietu.
- **Opcje** — dodatkowe informacje, nie zawsze używane, mogą dotyczyć na przykład funkcji zabezpieczeń.
- **Dane** — pole przeznaczone na dane pakietu lub jego fragmentu.

16.6.3. Protokoły warstwy transportowej

DEFINICJA

Warstwa transportowa zapewnia usługi przesyłania danych z hosta źródłowego do hosta docelowego. Ustanawia logiczne połączenie między hostem wysyłającym i odbierającym. Protokoły transportowe dzielą i scalają dane wysyłane przez aplikacje wyższej warstwy w jeden strumień danych przepływający między punktami końcowymi.

Protokoły warstwy transportowej to TCP i UDP.

Protokół IP pozwala na przenoszenie pakietów między sieciami, jednak nie zapewnia, że wysłane dane dotrą do adresata. Ta cecha powoduje, że protokół IP nazywany jest **bezpołączeniowym** — dane wysyłane są tylko w jedną stronę, bez potwierdzenia.

Za niezawodność przesyłu danych odpowiedzialny jest **protokół TCP**, nazywany protokołem **połączeniowym**. To on po odebraniu każdej porcji danych wysyła potwierdzenie do nadawcy, że dane zostały odebrane. W przypadku braku potwierdzenia dane wysyłane są ponownie.

Innym protokołem działającym na rzecz protokołu IP jest **UDP** (ang. *User Datagram Protocol*). Jest on bezpołączeniowym protokołem transportowym należącym do stosu protokołów TCP/IP. Służy do wysyłania datagramów bez potwierdzania czy gwarancji ich dostarczenia. Przetwarzanie błędów i retransmisja muszą być obsłużone przez protokoły wyższych warstw (np. warstwy aplikacji).

Protokół UDP jest zaprojektowany dla aplikacji, które nie mają potrzeby składania sekwencji segmentów. Nie przesyła on informacji o kolejności, w jakiej mają być odtworzone. Taka informacja jest zawarta w nagłówku segmentów protokołu TCP.

16.6.4. Protokoły warstwy aplikacji

DEFINICJA

Warstwa aplikacji zajmuje się świadczeniem usług dla użytkownika. Protokoły warstwy aplikacji definiują standardy komunikacji między aplikacjami (programami klienckimi a serwerowymi).

Najpopularniejsze protokoły warstwy aplikacji:

- **Telnet** (ang. *Network Terminal Protocol*) — protokół terminala sieciowego, pozwalający na zdalną pracę z wykorzystaniem konsoli tekstowej.
- **FTP** (ang. *File Transfer Protocol*) — protokół transmisji plików.
- **SMTP** (ang. *Simple Mail Transfer Protocol*) — protokół wysyłania poczty elektronicznej.
- **POP** (ang. *Post Office Protocol*) — protokół odbioru poczty elektronicznej.
- **HTTP** (ang. *Hypertext Transfer Protocol*) — protokół przesyłania stron WWW.
- **SSH** (ang. *Secure Shell Login*) — protokół terminala sieciowego zapewniający szyfrowanie połączenia.
- **DNS** (ang. *Domain Name Serwer*) — serwer nazw domenowych. Odpowiada za tłumaczenie adresów domenowych na adresy IP i odwrotnie.
- **DHCP** (ang. *Dynamic Host Configuration Protocol*) — protokół dynamicznej konfiguracji urządzeń. Odpowiedzialny za przydzielanie adresów IP, adresu domyślnej bramki i adresów serwerów DNS.
- **NFS** (ang. *Network File System*) — protokół udostępniania systemów plików (dysków sieciowych); działa, wykorzystując UDP, czyli bez potwierdzenia odbioru.
- **SNMP** (ang. *Simple Network Management Protocol*) — prosty protokół zarządzania siecią. Pozwala na konfigurację urządzeń sieciowych i gromadzenie informacji na ich temat.

16.7. Narzędzia dla protokołów TCP/IP

Poprawne skonfigurowanie protokołu IP pozwala na pracę z wykorzystaniem zasobów sieciowych. Każdy sieciowy system operacyjny oferuje narzędzia pozwalające sprawdzić poprawność konfiguracji.

16.7.1. Polecenie ipconfig

W systemach Windows poleceniem, które pozwala sprawdzić adresy przypisane do poszczególnych interfejsów, jest `ipconfig`. Narzędzie to pomaga przy wykrywaniu błędów w konfiguracji protokołu IP.

WSKAZÓWKA

Najczęściej polecenie `ipconfig` jest wykorzystywane w następujący sposób:

- `ipconfig` — pokazuje skróconą informację o połączeniu.
- `ipconfig /all` — pokazuje szczegółowe dane o konfiguracji wszystkich interfejsów.
- `ipconfig /renew` — odnawia wszystkie karty.
- `ipconfig /release` — zwalnia wszystkie połączenia.
- `ipconfig /?` — wyświetla komunikat pomocy.
- `ipconfig /flushdns` — czyści bufor programu rozpoznającego nazwy DNS.

Odpowiednikiem polecenia `ipconfig` w systemie Linux jest `ifconfig`.

16.7.2. Ping

Do diagnozowania połączeń w sieciach komputerowych TCP/IP używane jest polecenie `ping`. Pozwala ono na sprawdzenie, czy istnieje połączenie między dwoma urządzeniami, umożliwia określenie jego jakości poprzez mierzenie liczby zgubionych pakietów oraz czasu ich dotarcia do celu i z powrotem. Do badania jakości połączenia `ping` korzysta z protokołu ICMP.

Polecenie `ping` dostępne jest zarówno w systemie Windows, jak i Linux. Aby sprawdzić poprawność konfiguracji połączenia IP, należy użyć składni: `ping` *`nazwa_lub_adres_do_sprawdzenia`*.

16.7.3. Tracert

Komendą służącą do badania trasy pakietów IP w systemie Windows jest `tracert`. Sprawdza ona czasy dostępu do kolejnych routerów znajdujących się na drodze do adresu docelowego (rysunek 16.16).

```
C:\>tracert wikipedia.org

Trasa śledzenia do wikipedia.org [208.80.152.2]
przewyższa maksymalną liczbę przeskoków 30

  1     1 ms     1 ms     1 ms  fav182.internetdsl.tpnet.pl [83.13.21.182]
  2     2 ms     1 ms     1 ms  fav177.internetdsl.tpnet.pl [83.13.21.177]
  3    19 ms    26 ms    19 ms  kat-ru4.idsl.tpnet.pl [213.25.2.204]
  4    20 ms    19 ms    19 ms  kat-r1.tpnet.pl [212.160.0.61]
  5    19 ms    46 ms    19 ms  kat-r2.tpnet.pl [194.204.175.174]
  6    43 ms    42 ms    44 ms  xe-2-0-3-0.ffttr2.frankfurt.opentransit.net [193.251.249.9]
  7    56 ms    55 ms    56 ms  tengige0-2-1-0.lontr1.london.opentransit.net [193.251.131.134]
  8   138 ms   137 ms   137 ms  pos0-5-1-0.nyktr1.newyork.opentransit.net [193.251.243.21]
  9   137 ms   138 ms   137 ms  te4-1.nykse1.newyork.opentransit.net [193.251.243.197]
 10   137 ms   136 ms   137 ms  206.111.13.137.ptr.us.xo.net [206.111.13.137]
 11   140 ms   137 ms   139 ms  te0-12-2-0.rar3.nyc-ny.us.xo.net [207.88.12.173]
 12   161 ms   161 ms   183 ms  te-3-0-0.rar3.washington-dc.us.xo.net [207.88.12.74]
 13   160 ms   161 ms   160 ms  te-3-0-0.rar3.atlanta-ga.us.xo.net [207.88.12.9]
 14   160 ms   158 ms   159 ms  te-4-0-0.rar3.miami-fl.us.xo.net [207.88.12.6]
 15   176 ms   175 ms   175 ms  207.88.14.58.ptr.us.xo.net [207.88.14.58]
 16   184 ms   184 ms   184 ms  w006.z207088246.xo.cnc.net [207.88.246.6]
 17   191 ms   183 ms   189 ms  rr.pmtpa.wikimedia.org [208.80.152.2]

Śledzenie zakończone.

C:\>
```

Rysunek 16.16. Wynik działania funkcji tracert

WSKAZÓWKA

Często z wyników działania programu można odczytać wędrówkę pakietów po sieci, ponieważ niektóre nazwy routerów zawierają ich lokalizację. W przykładzie podanym na rysunku 16.16 pakiety do serwera wikipedia.org pokonały trasę z Katowic (z adresu kat-ru4.idsl.tpnet.pl), przez Frankfurt (xe-2-0-3-0.ffttr2.frankfurt.opentransit.net), Londyn (tengige0-2-1-0.lontr1.london.opentransit.net), Nowy Jork (pos0-5-1-0.nyktr1.newyork.opentransit.net), Waszyngton (te-3-0-0.rar3.washington-dc.us.xo.net), Atlantę(te-3-0-0.rar3.atlanta-ga.us.xo.net) i Miami (te-4-0-0.rar3.miami-fl.us.xo.net).

16.7.4. Netstat

Polecenie `netstat` jest jednym z najbardziej rozbudowanych poleceń, pozwalającym na sprawdzanie połączeń sieciowych (rysunek 16.17). Dostępne jest zarówno dla systemu Windows, jak i Linux. Umożliwia wyświetlanie aktywnych połączeń sieciowych TCP, a także portów, na których komputer nasłuchuje, tabeli routingu, statystyk itp.

```
[root@student root]# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.100.0   *               255.255.255.0   U         0 0          0 eth0
83.16.196.34    *               255.255.255.248   U         0 0          0 eth1
192.168.3.0     wenus           255.255.255.0   UG        0 0          0 eth2
192.168.2.0     mars            255.255.255.0   UG        0 0          0 eth3
192.168.1.0     neptun          255.255.255.0   UG        0 0          0 eth3
169.254.0.0     *               255.255.0.0     U         0 0          0 eth1
127.0.0.0       *               255.0.0.0       U         0 0          0 lo
default         gate            0.0.0.0         UG        0 0          0 eth0
[root@student root]#
```

Rysunek 16.17. Przykład wykorzystania polecenia netstat — tablica routingu

Polecenie `netstat` użyte bez parametrów powoduje wyświetlenie aktywnych połączeń protokołu TCP. Inne najważniejsze parametry polecenia to:

- `-a` — służy do wyświetlania wszystkich aktywnych połączeń oraz portów nasłuchu protokołów TCP i UDP.

- `-b` — służy do wyświetlania aktywnych połączeń protokołu TCP i nazw programów, które przypisane są do obsługi danego portu.
- `-e` — wyświetla statystykę sieci Ethernet.
- `-n` — wyświetla aktywne połączenia TCP (adresy i numery portów są wyrażane numerycznie).
- `-o` — wyświetla aktywne połączenia TCP i identyfikatory procesów (PID) poszczególnych połączeń.
- `-p` *`protokół`* — ukazuje połączenia wybranego protokołu (udp, tcpv6, tcp lub udpv6).
- `-s` — służy do wyświetlania oddzielnych statystyk dla poszczególnych protokołów.
- `-r` — służy do wyświetlania zawartości tabeli trasowania protokołu IP.

16.7.5. DNS

DEFINICJA

System nazw domenowych — DNS (ang. *Domain Name System*) to system serwerów oraz protokół komunikacyjny mający za zadanie tłumaczyć adresy w postaci przyjaznej dla człowieka (nazwy mnemoniczne) na adresy IP.

Adresy DNS składają się z domen internetowych rozdzielonych kropkami. Taki podział pozwala na budowanie hierarchii nazw, na przykład *www.helion.pl* oznacza nazwę *helion* zarejestrowaną w domenie krajowej *pl*. Przedrostek *www* oznacza nazwę usługi, jest administrowany przez właściciela głównej domeny.

Hierarchiczny charakter systemu domen pozwala na tworzenie poddomen. Na przykład dla domeny krajowej *pl* utworzonych zostało wiele poddomen:

- regionalnych, jak *katowice.pl*, *zakopane.pl* czy *waw.pl*,
- funkcjonalnych, jak *com.pl*, *gov.pl* czy *org.pl*,
- należących do firm, organizacji lub osób prywatnych, jak *helion.pl*.

Kolejne nazwy w adresie oddzielane są kropkami, ostatnia z nazw jest domeną najwyższego poziomu (ang. *top level domain*).

DNS to system organizacyjny zarządzany przez dwie instytucje — IANA i ICANN. Nadzorują one ogólne zasady przyznawania nazw domen i adresów IP. Nie zajmują się jednak przydzielaniem domen dla chętnych, a jedynie rozdzielają domeny najwyższego poziomu między kraje lub organizacje, które odpowiadają za ich przyznawanie. Domeną *pl* zarządza organizacja **Naukowe i Akademickie Sieci Komputerowe** (**NASK**). Rozdziela ona poddomeny w obrębie domeny *pl* między zainteresowanych, którzy stają się ich administratorami.

Domeny najwyższego poziomu tworzą szkielet sieci DNS. Informacje o nich przechowywane są na 13 głównych serwerach systemu DNS rozsianych po świecie. Gromadzą one dane o serwerach DNS, które zarządzają zarejestrowanymi domenami. To do nich przekierowywane są zapytania dotyczące poddomen utworzonych w domenach głównych.

16.8. Zasady transmisji w sieciach TCP/IP

Urządzenia pracujące w jednej sieci mają możliwość komunikacji tylko między sobą. Aby połączyć je z inną siecią, wymagany jest **router**. Jest to urządzenie, które przekierowuje pakiet do adresata znajdującego się w innej logicznej sieci IP.

16.8.1. Brama domyślna

DEFINICJA

Komunikacja w sieciach TCP/IP pozwala na wymianę danych tylko z urządzeniami znajdującymi się w danej sieci. Aby wysłać wiadomość poza sieć, w której pracuje urządzenie, należy ustawić parametr konfiguracyjny protokołu IP — *bramę domyślną*. Adres bramy domyślnej wskazuje na router, który przechowuje informacje o tym, jak dotrzeć do wybranej sieci.

Routery to węzły sieci. Mają za zadanie przesyłać pakiety do adresata, a dokładnie do sieci, w której znajduje się jego adres IP. Pakiet zaadresowany do komputera znajdującego się w naszej sieci jest kierowany bezpośrednio do niego. Jeśli ma zostać wysłany poza sieć, trafia do routera, który sprawdza, czy jest on kierowany do sieci bezpośrednio podłączonej do niego, czy ma być przesłany do urządzenia znajdującego się poza sieciami podłączonymi do routera. Pakiety wędrują od jednego węzła (routera) do drugiego poprzez wiele węzłów pośredniczących, często mogą też być transmitowane różnymi trasami. Zadaniem routera jest wybrać najlepszą dostępną drogę pomiędzy jednym a drugim węzłem. Decyzja o wyborze trasy podejmowana jest na podstawie wpisów znajdujących się w tablicy routingu — spisie sieci podłączonych bezpośrednio do routera oraz sieci dostępnych na routerach sąsiadujących.

DEFINICJA

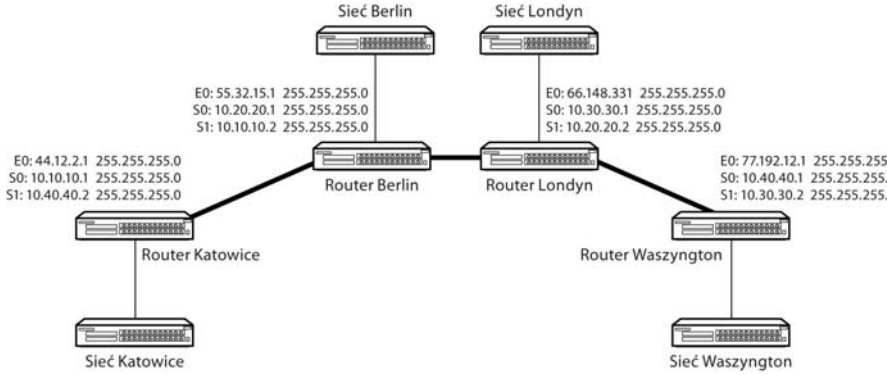
Tablica routingu może być utworzona przez administratora lub dynamicznie, przez protokoły routingu (nie mylić z protokołami rutowalnymi). *Routing* (*trasowanie*) polega na podjęciu decyzji, przez który fizyczny port lub przez którą sieć pakiety mają być wysłane, aby jak najszybciej dotarły do adresata.

Każdy wpis w tablicy routingu zawiera adres sieci docelowej oraz adres sieci lub interfejsu, przez który dana sieć docelowa jest osiągalna. Jeśli router zna więcej niż jedną

trasę do sieci docelowej, wybiera trasę najkorzystniejszą na podstawie metryki — wartości określającej jakość danej trasy.

Metryki zależne są od konkretnego protokołu routingu. Mogą opierać się tylko na liczbie routerów znajdujących się na drodze do celu, ale również na chwilowym obciążeniu łącza, jego prędkości czy opóźnieniach występujących w transmisji.

Przykład 16.1. Rysunek 16.18 przedstawia infrastrukturę sieciową dla przykładowej sieci łączącej Katowice, Berlin, Londyn i Waszyngton. W każdym z miast do zainstalowanych routerów podłączona została sieć lokalna. Informacje o dostępnych sieciach zapisane są w tablicach routingu zamieszczonych pod rysunkiem. Kolejne wpisy w tablicach oznaczają: adres sieci, adres interfejsu, przez który dana sieć jest osiągalna, oraz liczbę przeskoków do celu.



Router Katowice

10.10.10.0	Serial0	1
10.20.20.0	10.10.10.1	2
10.30.30.0	10.40.40.2	2
10.40.40.0	Serial1	1
44.12.2.0	Eth0	1
55.32.15.0	10.10.10.1	2
66.148.33.0	10.10.10.1	3
77.192.12.1	10.40.40.2	2

Router Berlin

10.10.10.0	Serial1	1
10.20.20.0	Serial0	1
10.30.30.0	10.20.20.1	2
10.40.40.0	10.10.10.2	2
44.12.2.0	10.10.10.2	2
55.32.15.0	Eth0	1
66.148.33.0	10.20.20.1	2
77.192.12.1	10.10.10.2	3

Router Londyn

10.10.10.0	10.20.20.2	2
10.20.20.0	Serial1	1
10.30.30.0	Serial0	1
10.40.40.0	10.30.30.1	2
44.12.2.0	10.30.30.1	3
55.32.15.0	10.20.20.2	2
66.148.33.0	Eth0	1
77.192.12.1	10.30.30.1	2

Router Waszyngton

10.10.10.0	10.40.40.1	2
10.20.20.0	10.30.30.2	2
10.30.30.0	Serial1	1
10.40.40.0	Serial0	1
44.12.2.0	10.40.40.1	2
55.32.15.0	10.40.40.1	3
66.148.33.0	10.30.30.2	2
77.192.12.1	Eth0	1

Rysunek 16.18. Przykładowa infrastruktura sieci

16.8.2. Protokoły routingu

Routery budują tablice routingu na podstawie informacji wymienionych z sąsiednimi routerami. Wymiana ta opiera się na **protokołach routingu**, które mają za zadanie poinformować sąsiednie węzły sieci o sieciach, do których ma dostęp. Takie rozwiązanie pozwala na dynamiczne budowanie struktury. Dołączenie kolejnej sieci do jednego z routerów nie wymaga rekonfiguracji pozostałych węzłów sieci. Zostaną one automatycznie poinformowane o zaistniałych zmianach. Protokoły routingu to: RIP, IGR, EIGRP, OSPF, BGP.

Protokoły routingu wysyłają informacje w określonych interwałach czasowych. Jeśli dana trasa nie jest dostępna (nie dotarła informacja od sąsiedniego routera), to wpis dotyczący trasy i sieci, które były osiągalne, zostaje usunięty z tablicy routingu i —

o ile to możliwe — zastępowany jest innym wpisem (np. wcześniej odrzuconym jako mniej korzystny).

Dynamiczne przekazywanie informacji o stanie sieci poprawia jej działanie. Router, który utracił bezpośrednie połączenie z sąsiadującym węzłem, może połączyć się z nim inną drogą. Routery mają możliwość zdefiniowania **routingu domyślnego** — trasy określającej dostęp do wszystkich sieci, które nie są wpisane w tablicy routingu.

Przykład 16.2. Wróćmy do przykładu z rysunku 16.18. Węzeł Katowice ma dostęp do węzła Waszyngton dwoma drogami — przez Berlin i Londyn oraz bezpośrednio. Na podstawie informacji zawartych w tablicy routingu wszystkie pakiety kierowane są do sieci bezpośrednio łączącej oba węzły. Jeśli połączenie to zostanie zerwane (nie dotrze informacja protokołu routingu), to w tablicy routingu wpis dotyczący bezpośredniej dostępności węzła Berlin jest zamieniany na trasę przez Waszyngton.

Wpisy w tablicy routera Katowice będą wyglądać następująco:

```
10.20.20.0          10.40.40.1          3
10.30.30.0          10.40.40.2          2
10.40.40.0          Serial1             1
44.12.2.0           Eth0                1
55.32.15.0          10.10.10.1          4
66.148.33.0         10.10.10.1          3
77.192.12.1         10.40.40.2          2
```

Wpis dla sieci 10.10.10.0 został usunięty — sieć ta przestała działać.

16.8.3. Gniazdo

DEFINICJA

Transmisja w sieciach TCP/IP opiera się na dwóch elementach — adresie urządzenia i numerze portu. Taka para parametrów transmisji nazywana jest *gniazdem*. Adres IP odpowiada za zidentyfikowanie pojedynczego urządzenia w sieci, numer portu oznacza, jaka aplikacja na urządzeniu docelowym ma przetwarzać przesłane dane.

Numery portów dodawane są do segmentów na poziomie warstwy czwartej (przez protokoły TCP i UDP). Numery portów zapewniają, że dane zostaną przetworzone przez konkretną aplikację. Na przykład podczas pobierania stron WWW zapytanie ze strony przeglądarki wysyłane jest na port 80 wybranego serwera WWW. Portem nadającym jest pierwszy wolny port powyżej 1023. Dane trafiają do serwera WWW na port 80 — jest to port, za którego obsługę odpowiada serwer HTTP. Serwer WWW wysyła dane (wybraną stronę) do klienta, kierując odpowiedź na port, z którego przy-

szło zapytanie. Komputer odbierający na podstawie portu kieruje odebrane dane do przetworzenia przez program, który wysłał zapytanie.

Numery portów do numeru 1023 przypisywane są znanym usługom sieciowym, na przykład 21 — FTP, 23 — Telnet, 25 — SMTP, 80 — HTTP. Numery portów powyżej 1024 są przydzielane dynamicznie programom, które korzystają z połączeń sieciowych.

ĆWICZENIA

1. Sprawdź budowę sieci, do której jesteś podłączony. Z jakiego medium korzysta? W jakiej topologii jest zbudowana?
2. Sprawdź adres IP przypisany do komputera, na którym pracujesz.
3. Sprawdź dostępność swojej bramy domyślnej.
4. Sprawdź trasę wędrowania pakietów do dowolnego adresu w domenie *com*.
5. Wyświetl tablicę routingu na komputerze.

PYTANIA I POLECENIA KONTROLNE

1. Co nazywane jest siecią komputerową?
2. Jak klasyfikuje się sieci ze względu na sposób działania?
3. Czym różnią się sieci LAN od sieci WAN?
4. Opisz budowę i przeznaczenie kabla koncentrycznego.
5. Dlaczego przewody w kablu UTP są skręcone?
6. Czym różni się kabel prosty od kabla skrosowanego? Jakie urządzenia można nimi połączyć?
7. Jakie urządzenia mają przypisany adres MAC?
8. Czym różni się koncentrator od przełącznika?
9. Jakie zadanie pełni router?
10. W jakich trybach może pracować sieć bezprzewodowa?
11. Jak nazywa się punkt styku sieci kablowej i bezprzewodowej?
12. Opisz topologię magistrali. W jaki sposób następuje dostęp do medium transmisyjnego?
13. W jakich sieciach wykorzystywany jest mechanizm przekazywania żetonu (tokenu)?
14. Scharakteryzuj topologię gwiazdy.
15. Wymień wszystkie warstwy modelu OSI. Jakie funkcje pełnią one w transmisji danych?
16. Co oznacza termin *enkapsulacja*?

PYTANIA I POLECENIA KONTROLNE ciąg dalszy

17. Jakie rodzaje transmisji występują w sieciach?
18. Czym różni się model TCP/IP od modelu OSI?
19. Jakie urządzenia działają w warstwie dostępu do sieci oraz w warstwie sieci?
20. Wymień protokoły warstwy sieci.
21. Wymień protokoły warstwy aplikacji.
22. Czym różni się protokół TCP od UDP?
23. Jakie polecenie pozwala sprawdzić konfigurację interfejsów sieciowych w systemach Windows? Jaki jest jego odpowiednik w systemie Linux?
24. Jakie polecenie pozwala śledzić trasę pakietu w systemie Windows? Jaki jest jego odpowiednik w systemie Linux?
25. W jaki sposób wyświetlić tablicę routingu dostępną w systemie Windows?
26. Jaką funkcję pełni serwer DNS?
27. Co jest zadaniem routingu?
28. Jaka jest różnica między protokołem rutowalnym a protokołem routingu?
29. Jakie wpisy zawiera tablica routingu?
30. Wymień trzy protokoły routingu. Jakie jest ich zadanie?
31. Co oznacza termin *gniazdo* w przypadku transmisji sieciowej?