helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

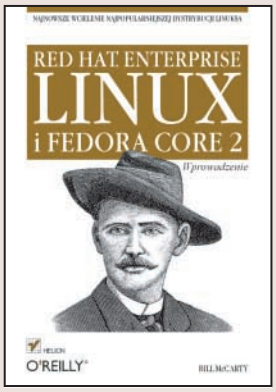
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Red Hat Enterprise Linux i Fedora Core 2. Wprowadzenie



Autor: Bill McCarty
Tłumaczenie: Grzegorz Kowalczyk
ISBN: 83-7361-672-1
Tytuł oryginału: Learning Red Hat Enterprise Linux & Fedora
Format: B5, stron: 360

Najnowsze wcielenie najpopularniejszej dystrybucji Linuksa

- Instalacja i konfiguracja systemu
- Środowiska graficzne i tryb konsoli
- Administracja systemem

Chyba wszyscy użytkownicy komputerów słyszeli o Linuksie. System ten, dzięki swojej elastyczności, stabilności, niezawodności i ogromnej wydajności jest instalowany na coraz większej liczbie komputerów. Znaczna ich część to serwery sieciowe, ale coraz częściej na instalację Linuksa decydują się również użytkownicy komputerów domowych i biurowych. Producent najpopularniejszej chyba dystrybucji Linuksa – firma RedHat Inc. – dokonała podziału swoich produktów na dwie kategorie: profesjonalne i domowe. Efekty tego podziału – dystrybucje RedHat Enterprise Linux oraz Fedora Core – są przeznaczone dla różnych grup odbiorców i różnią się zasadami dystrybucji.

Książka „Red Hat Enterprise Linux i Fedora Core 2. Wprowadzenie" omawia obie dystrybucje; jest przeznaczona dla użytkowników, którzy nie mieli dotychczas kontaktu z systemami operacyjnymi z rodziny Linuksa. Wiadomości w niej zawarte koncentrują się głównie na programach, narzędziach i aplikacjach pracujących w środowisku graficznym. W książce znajdziesz także omówienie bardziej zaawansowanych zagadnień, m.in. opis instalacji i konfiguracji serwerów usług sieciowych. Wszystko to przedstawione jest jednak w formie zrozumiałej dla początkujących użytkowników Linuksa.

- Historia Linuksa
- Przygotowanie do instalacji
- Instalacja i konfiguracja systemu
- Struktura plików i katalogów w systemie
- Środowiska graficzne Gnome i KDE
- Praca w środowisku tekstowym
- Instalacja nowego oprogramowania za pomocą narzędzia RPM Package Manager
- Połączenie z siecią internet
- Konfiguracja usług sieciowych – serwera plików i serwera WWW
- Programowanie skryptów powłoki

Wiadomości zawarte w niniejszej książce rozwieją obawy każdego, kto czuje respekt przed Linuksem.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Przedmowa **7**

1. Dlaczego Linux? **15**
Dlaczego Red Hat Enterprise Linux i Fedora? 15
Czym jest Linux? 16
Linux, czy nie Linux — oto jest pytanie 32

2. Przygotowania do instalacji systemu Linux **37**
Minimalne wymagania sprzętowe systemu Linux 37
Gromadzenie informacji o konfiguracji komputera 40
Przygotowanie dysku twardego do instalacji systemu Linux 46

3. Instalacja systemu Linux **55**
Instalacja systemu operacyjnego oraz aplikacji 55
Uruchamianie procesu instalacji 56
Pierwsze uruchamianie systemu 92
Logowanie do systemu pracującego w trybie graficznym 97
Rozwiązywanie problemów 100

4. System Linux — jak to właściwie działa? **103**
Konta użytkowników 103
Sposób organizacji danych w systemie Linux 104
Korzystanie z systemu X Window 110

5. Środowiska graficzne GNOME i KDE **117**
Środowisko graficzne GNOME 118
Środowisko graficzne KDE 131

6. Korzystanie z aplikacji dostępnych w systemie Linux **143**
OpenOffice.org 143
Evolution — klient poczty elektronicznej 152
Obsługa urządzeń typu PDA 155
Druid CD 159

7. Praca z powłoką systemu Linux 161
Uruchamianie poleceń powłoki 162
Praca z wierszem poleceń powłoki systemu Linux 166
Przydatne polecenia systemu Linux 186

8. Instalacja oprogramowania przy użyciu menedżera pakietów RPM Package Manager 191
Menedżer pakietów RPM 191
Polecenie redhat-install-packages 195
Polecenie rpm 196
Zapytania do bazy pakietów RPM 197
Instalacja pakietów RPM 198
Odinstalowanie pakietu 201
Aktualizacja pakietu 201
Odświeżenie pakietu 202
Zaawansowane zastosowania programu rpm 202
Aktualizowanie systemu Red Hat Enterprise Linux 203
Aktualizacja systemu Fedora Core 209

9. Zarządzanie i konfiguracja systemu Linux 211
Konfiguracja systemu Linux przy użyciu narzędzi z menu Ustawienia systemowe 211
Zarządzanie systemem Linux przy użyciu narzędzi z menu Narzędzia systemowe 227
Zarządzanie usługami systemowymi 235

10. Podłączenie do sieci Internet 241
Podstawy sieci komputerowych 241
Konfiguracja połączenia internetowego 242
Przeglądarka internetowa Mozilla 254
gFTP — Klient FTP 257
Korzystanie z wvdial 258
Konfiguracja systemu Linux do korzystania z modemu kablowego lub modemu DSL 259

11. Konfiguracja usług sieciowych 261
Konfiguracja hosta 262
Samba 264
Serwer Apache 275
Bezpieczna powłoka SSH 280
Zastosowanie zapory sieciowej 285
Kontrola bezpieczeństwa systemu przy użyciu programu Nmap 287
Bezpieczeństwo sieci komputerowych 288

12. Zastosowanie skryptów i innych zaawansowanych mechanizmów powłoki 291
Powłoka systemu UNIX — potężne narzędzie w rękach użytkownika 291
Mapowanie nazw plików 293
Aliasy poleceń powłoki 294
Zastosowanie konsoli wirtualnych 295
X Window kontra powłoka systemu Linux 297
Skrypty powłoki 298
Skrypty powłoki — techniki zaawansowane 306

A Drzewo katalogów systemu Linux 315

B Najważniejsze pliki systemu Linux 319

C Zarządzanie procesem uruchamiania systemu Linux 323
Uruchamianie systemu Linux 323
Dyskietka startowa systemu Linux 324
Program ładujący GRUB 325
Parametry uruchomieniowe systemu 328

D Lista poleceń systemu Linux 339

Skorowidz 345

ROZDZIAŁ 7.

Praca z powłoką systemu Linux

System Linux udostępnia dwa rodzaje interfejsów użytkownika: graficzny interfejs użytkownika (ang. *GUI — Graphical User Interface*), działający pod kontrolą serwera X, oraz stary, dobry, tekstowy interfejs wiersza poleceń (ang. *CLI — Command Line Interface*), nazywany w skrócie powłoką systemu.

Użytkownicy mający pewne doświadczenia ze znanym z systemu Windows oknem wiersza poleceń MS-DOS z pewnością zauważą spore podobieństwo do niego powłoki systemu Linux, gdzie użytkownik wpisuje z klawiatury polecenia dla systemu i może bezpośrednio na ekranie oglądać wyniki ich działania. Należy tutaj jednak przyznać, że porównywanie powłoki systemu Linux z oknem wiersza poleceń MS-DOS jest wielce dla tej pierwszej krzywdzące, głównie ze względu na fakt, że powłoka systemu Linux daje użytkownikowi o wiele większe możliwości. Krótko mówiąc, w tym konkretnym przypadku określenie *stary interfejs* wcale nie oznacza *gorszy* — a wręcz przeciwnie.

Graficzne interfejsy użytkownika są ostatnimi czasy bardzo modne, głównie ze względu na łatwość, z jaką użytkownik może się nimi posługiwać. Nie zawsze jednak są one najbardziej efektywnym sposobem komunikacji użytkownika z komputerem. Doświadczony, dysponujący odpowiednią wiedzą użytkownik może często pracować daleko bardziej efektywnie w trybie tekstowym niż jego konkurent wykorzystujący interfejs GUI. Co więcej, użytkownik pracujący z interfejsem graficznym może wykonywać tylko takie operacje, jakie zostały przewidziane i zaimplementowane przez jego twórców. Z kolei powłoka systemu jest o wiele bardziej elastyczna i podatna na zwiększanie swojej funkcjonalności; przykładowo, użytkownik może wykonywać zupełnie nowe, nietypowe zadania, bazując na odpowiednio dobranych sekwencjach istniejących poleceń.

Bez żadnej przesady można powiedzieć, że prawdziwa siła systemu Linux leży właśnie w jego powłoce. Wynika stąd jasno, że jeżeli użytkownik chce na poważnie pracować z systemem Linux, to powinien solidnie i rzetelnie zapoznać się z możliwościami i obsługą powłoki tego systemu. Nie każdemu użytkownikowi niezbędna będzie wiedza o systemie na poziomie prawdziwego guru, niemniej jednak nawet bardzo podstawowe wiadomości o funkcjonowaniu powłoki z pewnością będą bardzo procentowały w praktyce. Wiele książek, publikacji, witryn internetowych i innych źródeł zajmujących się zagadnieniami systemu Linux domyślnie zakłada, że użytkownik dysponuje przynajmniej elementarną wiedzą o powłoce systemu; co więcej, w pewnych sytuacjach, jak np. kłopoty z uruchomieniem środowiska X, usunięcie przyczyny danego problemu bez znajomości poleceń powłoki staje się bardzo utrudnione, jeśli nie wręcz niemożliwe.

System Linux obsługuje wiele różnych rodzajów powłok; jedną z najbardziej popularnych jest powłoka *bash*. W niniejszym rozdziale będziemy omawiali zagadnienia dotyczące właśnie tej powłoki; omówimy sposoby wpisywania i wykonywania poleceń powłoki, zasady używania poleceń powłoki do zarządzania plikami i katalogami, sposoby pracy z wymiennymi nośnikami pamięci masowych oraz metody uruchamiania programów i aplikacji. Na zakończenie zajmiemy się omówieniem edytora *nano*, prostego, ale bardzo przydatnego narzędzia do przeglądania i edycji plików tekstowych.

Uruchamianie poleceń powłoki

W praktyce najczęściej wykorzystywanym sposobem interakcji użytkownika z powłoką systemu jest uruchomienie okna terminala, tak jak to zostało omówione wcześniej w rozdziałach 5 i 6. Należy jednak pamiętać o tym, że okno terminala nie jest jedyną metodą dostępu do powłoki systemu Linux — użytkownik może również skorzystać z konsoli wirtualnej, co zostało bardziej szczegółowo omówione w sekcji „Using virtual consoles" w rozdziale 4.

Zatem, uruchamiamy okno terminala — i do dzieła! Pierwsze polecenie, z którym zapoznamy użytkownika, jest bardzo krótkie — jego nazwa składa się zaledwie z jednej litery: `w`. Po otwarciu okna terminala, wpisujemy polecenie `w` i naciskamy klawisz *Enter*. Odpowiedź powłoki powinna wyglądać mniej więcej tak:

```
[helion@localhost helion]$ w
12:16:20 up  2:12,  3 users,  load average: 2,63, 2,45, 2,40
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
helion   tty3     -                12:14    6.00s  1.15s  1.15s -bash
root     :0       -                Tue23   ?xdm?  33:05  12.41s gnome-session
root     pts/4    -                12:08    1.00s 13.02s  0.42s w
```

Zadaniem polecenia `w` jest wyświetlenie bieżącego statusu systemu operacyjnego oraz listy wszystkich aktualnie zalogowanych użytkowników. Z przedstawionych powyżej wyników działania użytkownik może się dowiedzieć, że: obecnie jest godzina 12:16, system jest włączony od 2 godzin i 12 minut, oraz że obecnie jest 2 użytkowników zalogowanych do systemu. Warto zwrócić uwagę, że wyniki działania polecenia `w` są bardzo „upakowane" i zawierają stosunkowo dużą ilość informacji w zaledwie kilku wierszach na ekranie. Tego typu zwięzłość informacji nie jest niczym szczególnym w systemie Linux, można nawet śmiało zaryzykować twierdzenie, że jest to zupełnie naturalny sposób porozumiewania się powłoki systemu z użytkownikiem. Niezbyt doświadczeni użytkownicy mogą na początku odnieść wrażenie, że wyniki działania poleceń powłoki prezentowane na ekranie są bardzo nieczytelne i wręcz niezrozumiałe, aczkolwiek zazwyczaj po pewnym czasie takie nastawienie diametralnie się zmienia i użytkownicy zaczynają doceniać zwięzłość i treściwość komunikatów generowanych przez system.

Wspomniana już wcześniej zwięzłość wyników działania poszczególnych poleceń nie jest w żaden sposób wynikiem przeoczenia ani tym bardziej zaniedbania projektantów systemu Linux. Przyczyna jest bardziej prozaiczna — polecenia powłoki systemu Linux zostały zaprojektowane tak, aby ich wyniki działania mogły być przetwarzane równie łatwo przez inne programy, jak i przez użytkowników systemu. Obecna struktura wyników działania poleceń powłoki zdecydowanie ułatwia życie programistom piszącym programy wykorzystujące inne polecenia powłoki.

Jak nietrudno się domyślić, poza poznanym już poleceniem `w` system Linux oddaje użytkownikowi do dyspozycji całe mnóstwo innych poleceń — jest ich tak wiele, że zapamiętanie wszystkich nazw poleceń, ich składni i przeznaczenia stanowi naprawdę poważny problem. Na szczęście ilość poleceń, z których przeciętny użytkownik musi korzystać na codzień, jest relatywnie mała i przy odrobinie wprawy, korzystanie z nich staje się niemal drugą naturą użytkownika.

Kolejnym poleceniem, które tutaj omówimy, jest polecenie `date`:

```
[helion@localhost helion]$ date
pią sie 13 12:25:53 CEST 2004
```

Jak widać, wykonanie polecenia `date` powoduje wyświetlenie na ekranie bieżącej daty i czasu.

Jeżeli dla użytkowników pracujących wcześniej z systemem Windows korzystanie z okna poleceń MS-DOS wydawało się bardzo uciążliwe i niemal wbrew naturze, to z pewnością nieco podobne, ambiwalentne uczucia będą im towarzyszyły w pierwszych dniach pracy z powłoką systemu Linux. Obiektywnie jednak patrząc na całe zagadnienie, naprawdę warto poświęcić chwilę czasu i spróbować przyzwyczaić się do specyfiki takiej pracy. Powłoka systemu Linux udostępnia użytkownikowi wiele mechanizmów, które powodują, że praca z nią jest o wiele przyjemniejsza i co najważniejsze, o wiele bardziej wydajna niż to miało miejsce w przypadku pracy z oknem MS-DOS w systemie Windows.

Poprawianie błędnie wprowadzonych poleceń

Nie będzie w tym nic dziwnego, jeżeli od czasu do czasu pracując z powłoką systemu Linux użytkownik popełni błąd podczas wpisywania nazwy polecenia. Przykładowo, załóżmy, że podczas wpisywania polecenia `date` użytkownik pomylił się i polecenie zostało wpisane jako `dat`:

```
[helion@localhost helion]$ dat
bash: dat: command not found
```

W takim przypadku użytkownik powinien dokładnie sprawdzić nazwę wpisanego polecenia i spróbować wprowadzić je ponownie, tym razem wpisując poprawną nazwę. Jeżeli błąd zostanie zauważony jeszcze przed naciśnięciem klawisza *Enter*, to użytkownik powinien użyć klawisza *Backspace* bądź klawiszy kursora (klawisz strzałka w lewo) do przesunięcia punktu wstawiania w miejsce, do którego zakradł się błąd, a następnie powinien poprawić błędnie wpisane znaki. Różnica pomiędzy wspomnianymi klawiszami polega na tym, że naciśnięcie klawisza *Backspace* powoduje usunięcie znaku znajdującego się na lewo od kursora (punktu wstawiania), natomiast klawisz strzałka w lewo powoduje wyłącznie przesuwanie punktu wstawiania o jeden znak w lewo za każdym naciśnięciem tego klawisza. Do kasowania niepotrzebnych, błędnie wpisanych znaków można również użyć klawisza *Delete*, którego naciśnięcie powoduje usunięcie znaku zlokalizowanego na prawo od kursora (miejsca wstawiania).

Podobnie jak przeglądarka sieciowa przechowuje historię witryn odwiedzanych przez użytkownika, tak powłoka *bash* przechowuje historię wydawanych poleceń. Użytkownik może przeglądać listę uprzednio wykonywanych poleceń korzystając z klawiszy strzałka w górę i strzałka w dół; posługując się naszą analogią, klawisze te spełniają podobną rolę jak przyciski *Naprzód* i *Wstecz* w przeglądarce sieciowej. Aby ponownie wykonać dane polecenie, należy używając klawiszy kursora „przewinąć" listę poleceń tak, aby żądane polecenie pojawiło się w wierszu poleceń, a następnie nacisnąć klawisz *Enter*. W razie potrzeby, użytkownik przed wykonaniem takiego polecenia może dokonać jego modyfikacji. Podczas wpisywania

poleceń, użytkownik ma do dyspozycji swego rodzaju miniedytor, nieco przypominający edytor DOSKEY znany z systemu MS-DOS. Miniedytor daje użytkownikowi do dyspozycji szereg funkcji uruchamianych naciśnięciami odpowiednich kombinacji klawiszy, przy użyciu których można modyfikować komendy wpisywane w wierszu poleceń. W tabeli 7.1 przedstawiono zestawienie najbardziej użytecznych funkcji miniedytora interpretowanych przez powłokę systemu. Wspomniana wcześniej funkcja przechowywania historii poleceń umożliwia wywołanie do 500 (!) ostatnio wykonywanych poleceń. Historia poleceń powłoki *bash* jest przechowywana w specjalnym pliku o nazwie *.bash_history*, zlokalizowanym w katalogu domowym danego użytkownika.

Tabela 7.1. Zestawienie najbardziej użytecznych funkcji miniedytora poleceń powłoki systemu

Kombinacja klawiszy	Przeznaczenie
Strzałka w górę	Wyświetla na ekranie poprzednie polecenie z listy historii poleceń
Strzałka w dół	Wyświetla na ekranie następne polecenie z listy historii poleceń
Strzałka w lewo	Przesuwa punkt wstawiania (kursor) o jeden znak w lewo
Strzałka w prawo	Przesuwa punkt wstawiania (kursor) o jeden znak w prawo
Backspace	Usuwa znak na lewo od punktu wstawiania (kursora)
Tab	Dopełnia nazwę polecenia, pliku lub katalogu na podstawie wpisanych znaków
Alt+B	Przesuwa punkt wstawiania (kursor) o jedno słowo w lewo
Alt+D	Usuwa słowo znajdujące się na prawo od punktu wstawiania (kursora)
Alt+F	Przesuwa punkt wstawiania (kursor) o jedno słowo w prawo
Ctrl+A	Przesuwa punkt wstawiania (kursor) na początek wiersza
Ctrl+D	Usuwa znak na prawo od punktu wstawiania (kursora)
Ctrl+E	Przesuwa punkt wstawiania (kursor) na koniec wiersza
Ctrl+K	Usuwa wszystkie znaki znajdujące się na prawo od punktu wstawiania (kursora)
Ctrl+L	Czyści ekran i ustawia punkt wstawiania (kursor) w pierwszym wierszu na ekranie
Ctrl+U	Usuwa wszystkie znaki znajdujące się na lewo od punktu wstawiania (kursora)
Ctrl+Y	Przywraca ostatnio usunięty element
Esc+.	Wstawia ostatnie słowo z poprzednio wykonanego polecenia
Esc+?	Wyświetla listę możliwych dopełnień danego polecenia (a także nazwy pliku lub nazwy katalogu)

Jedną z najbardziej przydatnych funkcji miniedytora powłoki jest automatyczne dopełnianie nazw po naciśnięciu klawisza *Tab*. Jeżeli użytkownik wpisze początkowy fragment nazwy polecenia, nazwy pliku lub katalogu, a następnie naciśnie klawisz *Tab*, to powłoka próbuje odszukać nazwę pasującego elementu i automatycznie ją dopełnić. Po dopełnieniu nazwy użytkownik może nacisnąć klawisz *Enter* i wykonać dane polecenie, bądź też dopisać dodatkowe elementy (przełączniki poleceń, argumenty, parametry itp.) i dopiero wtedy wykonać polecenie. Jak widać, mechanizm automatycznego dopełniania nazw powoduje, że korzystanie z powłoki systemu staje się o wiele łatwiejsze.

Oprócz wymienionych wcześniej kombinacji klawiszy służących do wywoływania funkcji miniedytora poleceń, powłoka systemu potrafi również interpretować naciśnięcia kilku dodatkowych kombinacji klawiszy, sterujących realizacją aktualnie wykonywanych poleceń (programów). Zestawienie kombinacji klawiszy sterujących realizacją poleceń przedstawiono

w tabeli 7.2. Przykładowo, naciśnięcie kombinacji klawiszy *Ctrl+C* powoduje przerwanie wykonywania polecenia (programu). Jak łatwo sobie wyobrazić, jest to bardzo użyteczna funkcja, choćby w sytuacji kiedy wykonywanie danego polecenia trwa już zbyt długo i użytkownik chce zamiast niego wykonać inne polecenie.

Tabela 7.2. Zestawienie kombinacji klawiszy sterujących realizacją poleceń powłoki

Kombinacja klawiszy	Przeznaczenie
Ctrl+C	Wysyła sygnał żądania przerwania realizacji aktualnie wykonywanego polecenia; w większości przypadków polecenie reaguje natychmiastowym zakończeniem pracy.
Ctrl+D	Wysyła sygnał końca pliku (ang. *EOF — End-of-File*) do aktualnie wykonywanego polecenia; ta kombinacja klawiszy jest przydatna w sytuacji, kiedy użytkownik chce zasygnalizować zakończenie procesu wprowadzania danych z konsoli.
Ctrl+Z	Zawiesza realizację aktualnie wykonywanego polecenia (programu). Użytkownik może wznowić realizację polecenia (programu), wpisując polecenie `fg`.

W tabeli 7.3 przedstawiono zestawienie znaków specjalnych, które sterują działaniem powłoki systemu. Znaki # oraz ; najczęściej są wykorzystywane w skryptach powłoki; więcej informacji na ten temat przedstawimy w dalszej części rozdziału. Znak & dodany na końcu polecenia powoduje, że po naciśnięciu klawisza *Enter* powłoka nie czeka z powrotem znaku zachęty do momentu zakończenia realizacji polecenia; znak zachęty pojawia się natychmiast po uruchomieniu polecenia, a samo polecenia jest wykonywane w tzw. tle (jako proces drugoplanowy — *przyp. tłum.*), dzięki czemu użytkownik może od razu przystąpić do wpisywania kolejnych poleceń. Z kolei zastosowanie znaku przekierowania strumienia danych , | (ang. *pipe redirector*) zostanie szerzej omówione w sekcji zatytułowanej „Wyświetlanie zawartości katalogów".

Tabela 7.3. Zestawienie znaków specjalnych sterujących działaniem powłoki systemu

Znak sterujący	Przeznaczenie
#	Wstawienie znaku # w danym wierszu skryptu powłoki powoduje, że jest on traktowany jako komentarz i ignorowany podczas realizacji skryptu.
;	Separator poleceń, pozwala na umieszczenie kilku poleceń w tym samym wierszu.
&	Znak & dodany na końcu polecenia powoduje, że po naciśnięciu klawisza *Enter* powłoka nie czeka z powrotem znaku zachęty do momentu zakończenia realizacji polecenia; znak zachęty pojawia się natychmiast po uruchomieniu polecenia, a samo polecenie jest wykonywane jako proces drugoplanowy.
>	Przekierowanie strumienia danych ze standardowego wyjścia danego polecenia do pliku; w składni polecenia po znaku > powinna znajdować się nazwa pliku, do którego będą zapisywane dane.
<	Przekierowanie strumienia danych z pliku na standardowe wejście danego polecenia; w składni polecenia po znaku < powinna znajdować się nazwa pliku, z którego będą odczytywane dane.
\	Znak używany na końcu danego wiersza poleceń; oznacza, że dalsza część polecenia została umieszczona w kolejnym wierszu.
|	Znak przekierowania strumienia danych ze standardowego wyjścia danego polecenia na standardowe wejście innego polecenia.

Praca z wierszem poleceń powłoki systemu Linux

Wszystkie polecenia powłoki systemu Linux mają wspólną, spójną i prostą strukturę. W niniejszej sekcji będziemy zajmować się zagadnieniami związanymi właśnie ze strukturą poleceń powłoki oraz opiszemy, w jaki sposób użytkownik może uzyskać dodatkowe informacje o przeznaczeniu, sposobie wykorzystania, działaniu oraz składni poszczególnych poleceń.

Polecenia i argumenty poleceń

Ogólna składnia wszystkich poleceń powłoki systemu Linux jest następująca:

```
nazwa_polecenia [opcje] [argumenty]
```

gdzie **`nazwa_polecenia`** określa operację, jaka zostanie wykonana przez powłokę systemu, natomiast `[opcje]` oraz `[argumenty]` pozwalają na sterowanie przebiegiem operacji oraz na przekazywanie wymaganych parametrów. Zarówno `[opcje]`, jak i `[argumenty]` polecenia są opcjonalne, tzn. mogą, ale nie muszą występować w wielu poleceniach; w formalnym zapisie składni polecenia elementy opcjonalne oznaczane są przy użyciu nawiasów kwadratowych.

Bardzo często **`nazwa_polecenia`** odnosi się do nazwy pliku programu, który powinien zostać uruchomiony po wydaniu takiego polecenia; takie komendy nazywane są *poleceniami zewnętrznymi*. W systemie Linux takie pliki poleceń zewnętrznych przechowywane są zazwyczaj w katalogach */bin*, */usr/bin* oraz */usr/local/bin*. Polecenia zewnętrzne, służące do zarządzania systemem, są zazwyczaj przechowywane w katalogach */sbin* lub */usr/sbin*; obydwa katalogi są domyślnie dołączane do ścieżki systemowej użytkownika *root*. Kiedy uruchamiane jest polecenie zewnętrzne, powłoka przekazuje wszystkie podane w wierszu polecenia argumenty i opcje do wywoływanego programu, który następnie je analizuje i w zależności od podanych opcji, odpowiednio modyfikuje swój sposób działania.

Niektóre komendy nie są poleceniami zewnętrznymi, ale wbudowanymi poleceniami powłoki systemu Linux; za ich realizację nie jest odpowiedzialny żaden dodatkowy program, ale bezpośrednio sama powłoka. Należy jednak pamiętać o tym, że różne powłoki mogą posiadać różne zestawy wbudowanych poleceń. W dalszej części rozdziału omówimy bardziej szczegółowo wybrane polecenia powłoki *bash*.

Nazwy poleceń powłoki systemu Linux niemal zawsze składają się z małych liter oraz cyfr. Większość poleceń pozwala na dodawanie różnego rodzaju opcji i argumentów, aczkolwiek jak już wspomnieliśmy wcześniej, nie zawsze jest to wymagane. Przykładowo, wpisanie polecenia `w` bez żadnych dodatkowych opcji i argumentów powoduje wyświetlenie na ekranie statystyki systemu Linux oraz listy użytkowników aktualnie zalogowanych do systemu.

Należy pamiętać o tym, że w poleceniach powłoki systemu Linux rozróżniane są małe i duże litery; podczas wpisywania poleceń należy więc zwracać baczną uwagę na ich pisownię.

Opcje umożliwiają modyfikację sposobu działania poleceń powłoki systemu Linux. Opcje poleceń mają zazwyczaj postać pojedynczej litery poprzedzonej znakiem `-`, aczkolwiek nie jest to regułą. Wiele poleceń pozwala na podawanie w wierszu wywołania kilku opcji; jeżeli taka sytuacja ma miejsce, to poszczególne opcje powinny być od siebie porozdzielane jedną

bądź kilkoma spacjami. Przykładowo, jeżeli do polecenia `w` dodamy opcję `-h`, to na ekranie nie zostanie wyświetlony wiersz nagłówka, zawierający informacje o bieżącym czasie oraz nazwy poszczególnych pól:

```
[helion@localhost helion]$ w -h
helion   tty3     -                12:14    3:35   1.43s  1.43s -bash
root     :0       -                Tue23   ?xdm?  40:01  17.76s gnome-session
root     pts/5    -                12:22    1.00s 11.98s  0.40s w -h
```

Argumentami polecenia mogą być m.in. nazwy plików i katalogów, nazwy użytkowników, jak również inne elementy, od których zależy sposób bądź wynik działania danego polecenia. Przykładowo, jako argument polecenia `w` można podać nazwę użytkownika, co spowoduje wyświetlenie informacji tylko o tym użytkowniku.

```
[helion@localhost helion]$ w helion
 12:29:04 up  2:25,  3 users,  load average: 1,80, 1,95, 2,13
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
helion   tty3     -                12:14    4:00   1.43s  1.43s -bash
```

Niektóre polecenia pozwalają na podawanie całej serii opcji i argumentów; należy tylko pamiętać o tym, że poszczególne argumenty powinny być od siebie oddzielone spacjami. Przykładowo, wykonanie przedstawionego poniżej polecenia spowoduje wyświetlenie informacji o użytkowniku *helion*, bez wiersza nagłówka:

```
[helion@localhost helion]$ w -h helion
helion   tty3     -                12:14    5:45   1.43s  1.43s -bash
```

Jeżeli podczas wpisywania polecenia podamy kilka opcji i argumentów, to może się zdarzyć, że takie polecenie nie zmieści się w jednym wierszu ekranu. Nie należy się tym jednak przejmować, ponieważ w takiej sytuacji powłoka automatycznie przeniesie dalszą część polecenia do następnego wiersza. Jeżeli z jakiegoś powodu użytkownik nie chce, aby powłoka automatycznie dzieliła polecenie na kolejne wiersze, to zbliżając się do końca wiersza użytkownik powinien wpisać znak `\` (ang. *backslash*), nacisnąć klawisz *Enter* i kontynuować wpisywanie polecenia w następnym wierszu. Znak `\` jest tzw. znakiem kontynuacji wiersza polecenia; powłoka systemu Linux „widzi" kolejne wiersze polecenia połączone znakiem `\` tak, jakby całe polecenie zostało wpisane w jednym wierszu. Należy jednak pamiętać o tym, aby po znaku `\` nic już nie zostało wpisane — w przeciwnym wypadku mechanizm łączenia wierszy może nie działać poprawnie.

Korzystanie z pomocy systemowej

Z względu na oczywisty fakt, że system Linux daje użytkownikowi do dyspozycji ogromną ilość poleceń, z których każde posiada kilka, kilkanaście bądź nawet kilkadziesiąt (!) opcji i argumentów, trudno się raczej spodziewać, że ktoś będzie w stanie to wszystko zapamiętać. Aby zatem użytkownik nie musiał borykać się z setkami stron przeróżnych „wydruków, podręczników, helpów i innych manuali", Linux udostępnia dwa bardzo wygodne polecenia, `man` oraz `apropos`, które umożliwiają dostęp do bazy pomocy systemowej, gdzie znajdują się opisy poszczególnych poleceń oraz ich opcji.

Zastosowanie polecenia man

Opis każdego polecenia systemu Linux znajduje się w specjalnym, osobnym zbiorze, nazywanym *stroną podręcznika man* (ang. *manual page*; w literaturze spotyka się również określenie *manpage* — *przyp. tłum.*) dla danego polecenia. Poszczególne pliki stron podręcznika *man*

przechowywane są w szeregu podkatalogów, tworzących razem bazę danych pomocy systemowej. Dostęp do tej bazy zapewnia właśnie polecenie *man*, którego działanie nieco przypomina funkcjonowanie znanego z systemu MS-DOS polecenia *help*. Przykładowo, aby wyświetlić na ekranie treść pomocy dla polecenia `w`, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ man w
```

Na rysunku 7.1 przedstawiono wyniki działania takiego polecenia — na ekranie pojawi się pierwsza strona podręcznika *man* dla polecenia `w`. Należy zwrócić uwagę, że w lewym dolnym narożniku ekranu pojawia się znak dwukropka, który oznacza, program oczekuje na dalsze polecenia użytkownika. Aby wyświetlić kolejną stronę podręcznika *man*, użytkownik powinien nacisnąć klawisz spacji; aby wyświetlić poprzednią stronę podręcznika, należy nacisnąć klawisz *b*. Aby zakończyć działanie polecenia *man* i powrócić do powłoki systemu, należy nacisnąć klawisz *q*.

```
W(1)                        Linux User's Manual                        W(1)

NAME
       w - Show who is logged on and what they are doing.

SYNOPSIS
       w - [husfV] [user]

DESCRIPTION
       w  displays  information  about the users currently on the machine, and
       their processes.  The header shows, in this order,  the  current  time,
       how  long  the  system  has  been running, how many users are currently
       logged on, and the system load averages for the past 1, 5, and 15  min-
       utes.

       The  following entries are displayed for each user: login name, the tty
       name, the remote host, login time, idle time, JCPU, PCPU, and the  com-
       mand line of their current process.

       The  JCPU  time  is the time used by all processes attached to the tty.
       It does not include past background jobs, but  does  include  currently
       running background jobs.

       The  PCPU  time  is  the time used by the current process, named in the
:_
```

Rysunek 7.1. Wygląd typowej strony podręcznika man

Wszystkie strony podręcznika *man* są zorganizowane w jeden, spójny sposób. Na początku każdej strony można znaleźć nazwę danej strony podręcznika oraz numer sekcji bazy danych, z której pochodzi dana strona (numer sekcji jest podany w nawiasach). Przykładowo, na rysunku 7.1 w lewym górnym oraz w prawym górnym narożniku ekranu widnieje ciąg znaków *W(1)*. Oznacza to po prostu, że w chwili obecnej wyświetlana jest odpowiednia strona podręcznika dla polecenia `w`, która należy do pierwszej sekcji bazy danych.

W kolejnej części podręcznika znajduje się nazwa oraz krótki opis polecenia, którego dana strona podręcznika dotyczy; następnie przedstawiana jest składnia polecenia, zawierająca wszystkie dostępne opcje i argumenty danego polecenia. Jak już wcześniej wspominaliśmy, w nawiasach kwadratowych podawane są opcje i argumenty opcjonalne. W kolejnej części podręcznika umieszczony jest szczegółowy opis samego polecenia, po którym następuje równie szczegółowy opis poszczególnych opcji i argumentów.

Jako że na etapie poznawania systemu Linux mniej doświadczeni użytkownicy z pewnością bardzo często będą sięgać do stron podręcznika *man*, to być może warto będzie poświęcić jedno okno terminala bądź jedną konsolę wirtualną wyłącznie na potrzeby polecenia *man*.

Alternatywnym rozwiązaniem jest przeglądanie plików stron podręcznika man przy użyciu menedżera plików Konqueror. Wpisanie w pasku adresu *man://index* spowoduje, że w oknie Konquerora zostanie wyświetlony indeks dostępnych stron podręcznika *man*, z którego, posługując się wygodnymi hiperłączami, można wywołać na ekran wybraną stronę pomocy. Dzięki takiemu rozwiązaniu użytkownik może wpisywać polecenie w jednym oknie terminala bądź na jednej konsoli wirtualnej, a do okna „pomocy" przełączać się w celu przypomnienia sobie składni wybranych poleceń.

W tabeli 7.4 przedstawiono zestawienie poszczególnych sekcji bazy stron podręcznika *man*; zawartość większości sekcji będzie przydatna zwłaszcza dla programistów i projektantów. Dla użytkowników i administratorów systemu najbardziej interesujące będą zapewne sekcje 1. i 8.

Tabela 7.4. Zestawienie poszczególnych sekcji stron podręcznika man

Numer sekcji	Zawartość
1	Polecenia zewnętrzne oraz wbudowane polecenia powłoki
2	Odwołania systemowe (inicjowane przez jądro systemu)
3	Odwołania do bibliotek (inicjowane przez biblioteki systemowe)
4	Pliki specjalne (np. pliki urządzeń)
5	Formaty plików i przyjęte konwencje
6	Gry
7	Pakiety makr i przyjęte konwencje
8	Polecenia przeznaczone do zarządzania systemem
9	Niestandardowe procedury jądra systemu

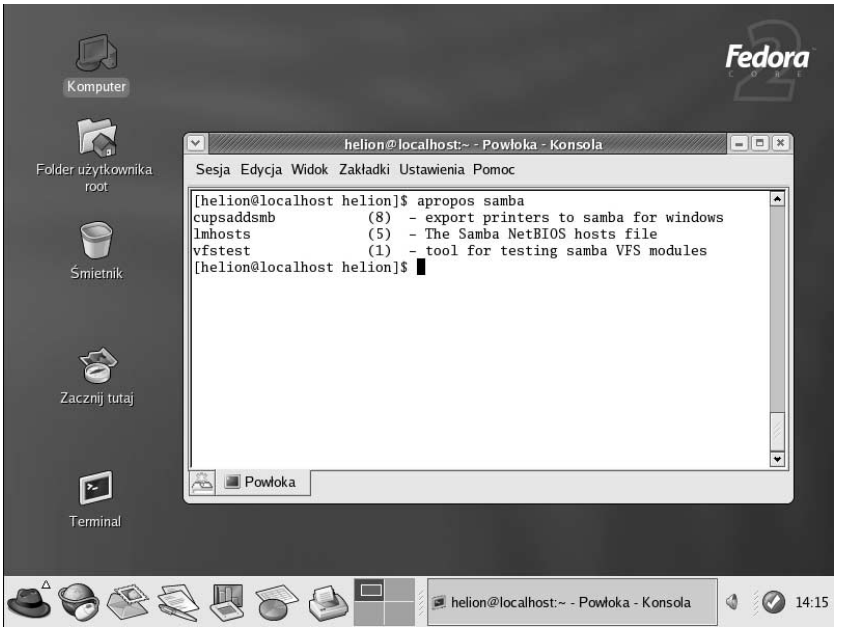
Zastosowanie polecenia apropos

Polecenie `man` pozwala użytkownikowi na przeszukiwanie stron podręcznika i wyświetlanie szczegółowych informacji o wybranym poleceniu. Polecenie `apropos` również przeszukuje bazę stron podręcznika, ale wyświetla na ekranie tylko jednowierszowe podsumowanie przeznaczenia wybranych poleceń. Argumentem, jaki przekazujemy do polecenia `apropos`, powinno być słowo kluczowe opisujące operację, jaką użytkownik chce wykonać; tak naprawdę polecenie `apropos` wyświetli na ekranie listę tych wszystkich poleceń, których wiersz podsumowania zawiera podane słowo kluczowe. Przykładowo, wykonując polecenie:

```
[helion@localhost helion]$ apropos samba
```

otrzymamy w wyniku jego działania listę poleceń, których strony podręcznika *man* zawierają słowo *samba*, jak to zostało przedstawione na rysunku 7.2.

Polecenie `apropos` jest bardzo użyteczne zwłaszcza w sytuacji, kiedy użytkownik nie może sobie przypomnieć nazwy polecenia, które miało realizować określoną czynność. Podając odpowiednie słowo kluczowe jako parametr polecenia `apropos`, użytkownik może wyświetlić na ekranie listę poleceń powiązanych z danym zagadnieniem i dzięki temu ma bardzo dużą szansę na odnalezienie właściwego polecenia.



Rysunek 7.2. Zastosowanie polecenia apropos

Do przechowywania informacji o poszczególnych poleceniach polecenie `apropos` wykorzystuje specjalną bazę danych. Zanim użytkownik będzie mógł wykonać to polecenie po raz pierwszy, musi utworzyć tę bazę danych logując się jako użytkownik *root* i wykonując następujące polecenie:

```
makewhatis
```

Wykonywanie powyższego polecenia może trwać nawet kilka czy kilkanaście minut. Po zainstalowaniu nowego pakietu oprogramowania, zawierającego polecenia zewnętrzne, użytkownik może (a nawet powinien) uaktualnić bazę danych, ponownie wykonując polecenie `makewhatis`.

Korzystanie z poleceń przeznaczonych do pracy z katalogami

Opisaliśmy już podstawowe zasady wykonywania poleceń powłoki systemu Linux, w związku z czym możemy przystąpić do omawiania poleceń przeznaczonych do pracy z katalogami systemu plików. Samo przeczytanie zawartości niniejszej sekcji może być niewystarczające; proponujemy zatem, aby Czytelnik po prostu zalogował się do swojego systemu Linux i wykonywał „na żywo" wszystkie polecenia pojawiające się w miarę czytania niniejszej sekcji. Każda okazja jest dobra do nabrania dodatkowych doświadczeń w pracy z powłoką systemu Linux.

Wyświetlanie bieżącego katalogu roboczego

Aby wyświetlić bieżący katalog roboczy, użytkownik powinien wykonać polecenie `pwd` (ang. *print working directory*). Polecenie `pwd` nie wymaga podawania żadnych opcji ani argumentów:

```
[helion@localhost helion]$ pwd
/root
```

Wykonanie polecenia `pwd` powoduje wyświetlenie na ekranie bezwzględnej ścieżki do bieżącego katalogu roboczego.

Zmiana bieżącego katalogu roboczego

Aby zmienić bieżący katalog roboczy, użytkownik powinien wykonać polecenie `cd` (ang. *change directory*), podając jednocześnie jako argument ścieżkę do nowego katalogu roboczego (podawana ścieżka może być zarówno względna, jak i bezwzględna). Przykładowo, aby zmienić bieżący katalog roboczy na katalog */bin*, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ cd /bin
[helion@localhost /bin]$
```

Warto zwrócić uwagę na fakt, że po zmianie bieżącego katalogu roboczego zmienia się wygląd znaku zachęty, jaki wyświetla powłoka po wykonaniu polecenia — znak zachęty odzwierciedla teraz nowe położenie katalogu roboczego.

Aby z dowolnej lokalizacji szybko powrócić do katalogu domowego danego użytkownika, wystarczy wykonać polecenie `cd`, nie podając żadnych argumentów:

```
[helion@localhost /bin]$ cd
[helion@localhost helion]$
```

Podobnie jak w poprzednim wypadku, po wykonaniu polecenia znak zachęty zmienił się, wskazując na nowy katalog roboczy.

Jeżeli użytkownik będzie usiłował zmienić bieżący katalog roboczy na katalog, który nie istnieje, powłoka systemu Linux wyświetli komunikat o błędzie:

```
[helion@localhost helion]$ cd abcdef
bash: cd: abcdef: Nie ma takiego pliku ani katalogu
```

Wyświetlanie zawartości katalogu

Aby wyświetlić zawartość katalogu, użytkownik może zastosować polecenie `ls` (ang. *list*). Polecenie `ls` posiada wiele bardzo przydatnych opcji, które pozwalają dopasować sposób jego działania oraz format prezentowanych wyników do indywidualnych upodobań użytkownika.

Najprostszą formą polecenia `ls` jest wykonanie go bez żadnych dodatkowych argumentów. Wykonanie takiego polecenia spowoduje wyświetlenie całej zawartości bieżącego katalogu roboczego, zarówno plików, jak i katalogów (wyniki działania tego polecenia na różnych komputerach będą wyglądały różnie, oczywiście ze względu na różną zawartość wyświetlanych katalogów).

```
[helion@localhost X11]$ ls
applnk        starthere                  Xmodmap
dm            sysconfig                  xorg.conf
fs            twm                        xorg.conf.backup
gdm           X                          Xresources
lbxproxy      xdm                        xserver
prefdm        XftConfig.README-OBSOLETE  xsm
proxymngr     xinit
serverconfig  xkb
[helion@localhost X11]$
```

Jak widać, na załączonym przykładzie nazwy plików i katalogów są wyświetlane w kolejności alfabetycznej, w trzech kolumnach. Warto zwrócić uwagę na fakt, że nazwy plików rozpoczynające się od dużej litery wyświetlane są przed nazwami plików rozpoczynającymi się od liter małych.

Bardziej wyrafinowana wersja polecenia `ls`, z dołączoną opcją `-l`, powoduje wyświetlenie na ekranie szczegółowych informacji o zawartości danego katalogu, jak to zostało zilustrowane na rysunku 7.3.

```
[root@localhost X11]# ls -l
razem 80
d rwxr-xr-x  2 root  root   4096 mar 12 05:33 applnk
d rwxr-xr-x  3 root  root   4096 sie  8 00:13 dm
d rwxr-xr-x  2 root  root   4096 sie  8 00:15 fs
d rwxr-xr-x  8 root  root   4096 mar 31 18:27 gdm
d rwxr-xr-x  2 root  root   4096 sie  8 00:02 lbxproxy
- rwxr-xr-x  1 root  root   1166 maj  7 06:29 prefdm
d rwxr-xr-x  2 root  root   4096 sie  8 00:02 proxymngr
d rwxr-xr-x  2 root  root   4096 mar 12 05:33 serverconfig
d rwxr-xr-x  2 root  root   4096 sie 07 23:49 starthere
d rwxr-xr-x  2 root  root   4096 mar 12 05:33 sysconfig
d rwxr-xr-x  2 root  root   4096 sie  8 00:02 twm
l rwxrwxrwx  1 root  root     24 sie  8 00:16 X -> ../../usr/X11R6/BIN/Xorg
d rwxr-xr-x  3 root  root   4096 sie  8 00:04 xdm
- rw-r--r--  1 root  root    372 maj  7 17:31 XftConfig.README-OBSOLETE
d rwxr-xr-x  3 root  root   4096 maj  7 17:31 xinit
l rwxrwxrwx  1 root  root     27 sie  8 00:02 xkb -> ../../usr/X11R6/lib/X11/xkb
- rw-r--r--  1 root  root    613 mar 31 09:10 Xmodmap
- rw-r--r--  1 root  root   2739 sie 10 21:48 xorg.conf
- rw-r--r--  1 root  root   2691 sie 10 21:39 xorg.conf.backup
- rw-r--r--  1 root  root    492 mar 31 09:10 Xresources
d rwxr-xr-x  2 root  root   4096 sie  8 00:02 xserver
d rwxr-xr-x  2 root  root   4096 sie  8 00:02 xsm
```

Typ | Prawa dostępu | Ilość dowiązań | Właściciel | Grupa | Rozmiar (w bajtach) | Data i czas ostatniej modyfikacji | Nazwa

Rysunek 7.3. Wyniki działania polecenia ls z dołączoną opcją -l

W pierwszym wierszu wyników działania wyświetlana jest informacja o całkowitej ilości miejsca na dysku, jakie zajmuje dany katalog razem ze swoimi podkatalogami i plikami; zajętość dysku podawana jest w KB. Pozostałe wiersze opisują poszczególne pliki i katalogi. Poniżej zamieszczono krótki opis poszczególnych kolumn:

Typ
: Wskazuje na jeden z dwóch typów obiektów: `d` oznacza katalog, natomiast `-` oznacza zwykły plik. Jak łatwo zauważyć, nazwy plików i katalogów są wyświetlane przy użyciu różnych kolorów.

Prawa dostępu
: Lista praw dostępu do poszczególnych plików i katalogów. Zagadnienia praw dostępu do plików i katalogów zostaną bardziej szczegółowo omówione w dalszej części niniejszego rozdziału.

Ilość dowiązań
: Informacja o ilości plików i katalogów dowiązanych (ang. *linked*) do danego obiektu.

Właściciel
: Nazwa użytkownika, który jest właścicielem danego pliku lub katalogu.

Grupa
: Nazwa grupy użytkowników, która jest właścicielem danego pliku lub katalogu.

Rozmiar
Rozmiar pliku lub katalogu, podany w bajtach.

Data i czas ostatniej modyfikacji
Znacznik daty i czasu ostatniej modyfikacji danego pliku lub katalogu.

Nazwa
Nazwa pliku lub katalogu.

Jeżeli w danym katalogu znajduje się duża ilość plików, to ich lista z pewnością nie zmieści się na jednym ekranie. Aby przeglądać listę plików ekran po ekranie, użytkownik powinien wykonać następujące polecenie:

```
[helion@localhost helion]$ ls | less
```

Jak łatwo zauważyć, w powyższym poleceniu użyto znaku przekierowania strumienia danych ze standardowego wyjścia polecenia `ls` na standardowe wejście polecenia `less`, które powoduje, że dane są wyświetlane po jednym ekranie naraz. Użytkownik może sterować działaniem polecenia `less`, korzystając z następujących klawiszy:

- *Spacja* — powoduje wyświetlenie kolejnego ekranu danych
- *b* — powoduje powrót do poprzednio wyświetlanego ekranu danych
- *q* lub *Q* — kończy działanie programu i zwraca sterowanie do powłoki (na ekranie pojawia się znak zachęty)

Jeżeli użytkownik chce wyświetlić zawartość katalogu innego niż bieżący katalog roboczy, to nazwa katalogu docelowego powinna zostać podana jako argument polecenia `ls`, np.:

```
[helion@localhost helion]$ ls /bin
```

Po wykonaniu powyższego polecenia, na ekranie zostanie wyświetlona zawartość katalogu */bin*, ale bieżący katalog roboczy nie ulegnie zmianie. W podobny sposób można wyświetlić na ekranie szczegółowe informacje o wybranym pliku lub katalogu — w tym celu wystarczy podać jego nazwę jako argument polecenia `ls`. Co więcej, polecenie `ls` akceptuje teoretycznie nieskończoną liczbę argumentów, dzięki czemu jako argumenty polecenia można podać całą serię nazw katalogów i w ten sposób uzyskać informacje o ich zawartości w jednym poleceniu; należy przy tym pamiętać, aby nazwy poszczególnych katalogów były od siebie oddzielone jedną bądź kilkoma spacjami.

Jeżeli nazwa katalogu lub pliku rozpoczyna się od znaku kropki (np. pliki *.bashrc*, *.bash_profile*), oznacza to, że takie pliki są *ukryte* (ang. *hidden*) i w wynikach działania polecenia `ls` wydanego w normalnym trybie są pomijane. Aby pliki ukryte były wyświetlane, należy do wiersza wywołującego polecenie `ls` dodać opcję `-a`. Przykładowo, aby wyświetlić wszystkie pliki i podkatalogi bieżącego katalogu roboczego, łącznie z plikami i katalogami ukrytymi, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ ls -a -l
```

Ciąg kolejnych opcji `-a -l` można w razie potrzeby połączyć, wpisując polecenie w następującej postaci:

```
[helion@localhost helion]$ ls -al
```

W katalogu domowym użytkownika znajduje się zazwyczaj cały szereg ukrytych plików, przechowujących informacje konfiguracyjne dla różnych programów. Przykładowo, plik *.bash_profile* zawiera informacje o konfiguracji powłoki *bash*.

Polecenie `ls` posiada jeszcze całkiem sporo innych, bardzo przydatnych możliwości, stąd w wolnej chwili naprawdę warto zajrzeć na strony podręcznika *man* polecenia `ls`.

Tworzenie nowych katalogów

Nowe katalogi mogą być tworzone przy użyciu polecenia `mkdir` (ang. *make directory*). Argumentem polecenia musi być nazwa tworzonego katalogu. Domyślnie system Linux tworzy nowy katalog jako podkatalog bieżącego katalogu roboczego. Przykładowo, wykonanie poniższego polecenia spowoduje utworzenie w katalogu domowym użytkownika *helion* podkatalogu o nazwie *redakcja*:

```
[helion@localhost helion]$ mkdir redakcja
```

Jeżeli użytkownik chce, aby nowy katalog zamiast w bieżącym katalogu roboczym został utworzony w innym katalogu docelowym, to jako argument polecenia `mkdir` musi podać bezwzględną ścieżkę do takiego katalogu. Przykładowo, jeżeli użytkownik *helion* znajduje się obecnie w swoim katalogu domowym, a chce utworzyć katalog */tmp/dokumenty*, to powinien wykonać następujące polecenie:

```
[helion@localhost helion]$ mkdir /tmp/dokumenty
```

Nazwa tworzonego katalogu musi spełniać kilka warunków i ograniczeń. Przykładowo, nazwa katalogu nie może zawierać znaku `/` (ang. *slash*). Nazwy poszczególnych katalogów i plików zazwyczaj składają się z dużych i małych liter, cyfr, kropek oraz znaków podkreślenia (_). W nazwach można również używać innych znaków, takich jak spacje i znak minus (-), niemniej jednak takie nazwy mogą powodować pewne problemy, gdyż powłoka systemu Linux interpretuje takie znaki w specjalny sposób. Jeżeli jednak z takich czy innych powodów użytkownik MUSI użyć znaków specjalnych w nazwach plików czy katalogów, to aby uniknąć problemów cała nazwa musi zostać ujęta w znaki pojedynczego cudzysłowu ' (ang. *single quote*). Znaki cudzysłowu nie są brane pod uwagę jako część nazwy. Opisana technika jest bardzo przydatna zwłaszcza w sytuacji, kiedy użytkownik podłącza się do zasobów dyskowych systemu plików Windows, gdzie nazwy katalogów i plików zawierające spacje (np. *Moje dokumenty*) są bardzo popularne.

Większość nazw plików w systemie MS-DOS zawiera znak kropki, co już nie jest regułą w systemie Linux. W systemie MS-DOS kropka oddziela główną część nazwy pliku od jej tzw. rozszerzenia, które używane było w zasadzie do rozróżniania typów poszczególnych plików. Przykładowo, w systemie MS-DOS plik o nazwie *memo.txt* prawie na pewno był plikiem tekstowym. Większość aplikacji w systemie Linux po prostu ignoruje rozszerzenia plików, stąd rozszerzenia nazw plików nie są w tym systemie wymagane. Z drugiej jednak strony, jeżeli zamierzamy wysłać dany plik komuś, kto używa innego systemu operacyjnego niż Linux, to zawsze warto takie rozszerzenie do nazwy dołączyć (np. *.txt* dla pliku tekstowego).

Usuwanie katalogu

Aby usunąć dany katalog, użytkownik powinien skorzystać z polecenia `rmdir` (ang. *remove directory*). Przykładowo, aby usunąć katalog o nazwie *archiwum*, znajdujący się w bieżącym katalogu roboczym, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ rmdir archiwum
```

Jeżeli użytkownik chce usunąć katalog zlokalizowany w innym miejscu niż bieżący katalog roboczy, to jako argument polecenia `rmdir` musi podać bezwzględną lub względną ścieżkę do takiego katalogu.

Polecenia `rmdir` nie można użyć do usunięcia katalogu, w którym nadal są przechowywane jakieś pliki bądź podkatalogi; użytkownik musi najpierw usunąć całą zawartość danego katalogu, a dopiero potem może przystąpić do usuwania samego katalogu. Małą niespodziankę mogą tutaj sprawić pliki ukryte, które mogą rezydować w danym katalogu choć nie są widoczne; zanim będziemy w stanie usunąć dany katalog, **wszystkie** pliki i podkatalogi — włączając w to pliku i katalogi ukryte — muszą zostać z niego usunięte.

Korzystanie z poleceń przeznaczonych do pracy z plikami

Jak łatwo się zorientować, katalogi mogą przechowywać zarówno pliki, jak i kolejne podkatalogi. W niniejszej sekcji omówimy kilka zagadnień związanych z poleceniami przeznaczonymi do pracy z plikami.

Wyświetlanie zawartości pliku

Pliki wykorzystywane w systemie Linux (podobnie zresztą jak w systemie Windows) mogą przechowywać tekst bądź też informacje binarne. O ile przeglądanie zawartości plików binarnych może mieć znaczenie dla wąskiej grupy specjalistów (np. programiści), o tyle zwykły użytkownik może w bardzo prosty sposób przeglądać pliki tekstowe. Aby tego dokonać, wystarczy wpisać polecenie `cat`, po którym następuje nazwa pliku tekstowego, którego zawartość chcemy wyświetlić na ekranie. Przykładowo, wykonanie następującego polecenia:

```
[root@localhost root]#  cat /etc/passwd
```

spowoduje wyświetlenie na ekranie zawartości pliku */etc/passwd*, w którym znajdują się informacje o kontach poszczególnych użytkowników danego systemu.

Jeżeli przeglądany plik jest zbyt duży, aby mógł się zmieścić na jednym ekranie, to efekt wykonania polecenia `cat` sprowadzi się do tego, że pierwsza część zawartości pliku zostanie błyskawicznie przewinięta na ekranie i zniknie z naszych oczu, pozostawiając widocznych jedynie kilkadziesiąt ostatnich wierszy. Aby tego uniknąć, należy skorzystać z poznanego już wcześniej polecenia `less`:

```
[root@localhost root]# less /etc/passwd
```

Wykonanie powyższego polecenia spowoduje wyświetlenie zawartości pliku */etc/passwd* strona po stronie, w sposób nieco przypominający działanie polecenia `man`, wyświetlającego kolejne strony podręcznika danego polecenia. Do sterowania działaniem polecenia `less` można wykorzystać następujące klawisze: *spacja* (wyświetlenie kolejnego ekranu danych), *b* (powrót do poprzednio wyświetlanego ekranu danych) oraz *q* lub *Q* (kończy działanie programu i zwraca sterowanie do powłoki; na ekranie pojawia się znak zachęty).

Usuwanie plików

Aby usunąć wybrany plik, należy użyć polecenia `rm` (ang. *remove*), podając jako argument nazwę pliku przeznaczonego do usunięcia. Przykładowo, wykonanie polecenia:

```
[helion@localhost helion]$ rm dane1
```

spowoduje usunięcie z bieżącego katalogu roboczego pliku o nazwie *dane1*. Jeżeli plik przeznaczony do usunięcia jest zlokalizowany w innym katalogu, to aby go usunąć, należy jako argument polecenia `rm` podać pełną ścieżkę do tego pliku (względną lub bezwzględną).

Po usunięciu pliku w systemie Linux, jego zawartość jest tracona bezpowrotnie. Aby uniknąć przypadkowego usunięcia plików zawierających istotne dane, podczas wykonywania operacji usuwania plików należy zachować szczególną ostrożność. Nie należy również zapominać o regularnym wykonywaniu kopii bezpieczeństwa danych, co pozwoli na ich odzyskanie w razie jakiegoś nieszczęścia.

Jeżeli użytkownik dołączy do polecenia `rm` opcję `-i`, to przed usunięciem danego pliku system poprosi o potwierdzenie zamiaru wykonania takiej operacji. Jeżeli użytkownik nie jest do końca przekonany, że wie co robi, to taka opcja może być całkiem przydatna. Jeżeli zalogujemy się do systemu jako użytkownik *root*, to czy tego chcemy czy nie, Linux automatycznie dołącza opcję `-i` do każdego polecenia `rm`.

Kopiowanie plików

Aby skopiować wybrany plik, należy użyć polecenia `cp` (ang. *copy*), podając dwa argumenty: nazwę (lub ścieżkę) pliku źródłowego oraz nazwę (lub ścieżkę) pliku docelowego. Przykładowo, wykonanie następującego polecenia:

```
[root@localhost /root]# cp /etc/passwd sample
```

spowoduje utworzenie kopii pliku */etc/passwd* i zapisanie jej w pliku o nazwie *sample*, zlokalizowanym w bieżącym katalogu roboczym.

Jeżeli docelowy plik już istnieje, to podczas operacji kopiowania zostanie on automatycznie nadpisany. Z tego powodu podczas wykonywania operacji kopiowania plików należy bardzo uważać, aby nie nadpisać pliku zawierającego jakieś cenne informacje. Przed rozpoczęciem kopiowania warto sprawdzić przy użyciu polecenia `ls` czy w lokalizacji docelowej nie ma już pliku o takiej samej nazwie; alternatywnym rozwiązaniem może być również dołączenie do polecenia `cp` opcji `-i`, która spowoduje, że przed nadpisaniem pliku docelowego system poprosi o potwierdzenie wykonania takiej operacji. Jeżeli zalogujemy się do systemu jako użytkownik *root*, to podobnie jak to miało miejsce w przypadku polecenia `rm`, Linux automatycznie dołącza opcję `-i` do każdego polecenia `cp`.

Przenoszenie i zmiana nazwy plików

Aby zmienić nazwę pliku, należy skorzystać z polecenia `mv` (ang. *move*), podając jako argumenty starą nazwę (lub ścieżkę) pliku oraz nową nazwę (lub ścieżkę) pliku. Przykładowo, wykonanie polecenia:

```
[helion@localhost helion]$ mv stary1 nowy2
```

spowoduje zmianę nazwy pliku *stary1* na nazwę *nowy2*.

Jeżeli docelowy plik już istnieje, to podczas operacji zmiany nazwy bądź przenoszenia pliku zostanie on automatycznie nadpisany. Z tego powodu podczas wykonywania operacji zmiany nazwy bądź przenoszenia pliku należy bardzo uważać, aby nie nadpisać pliku zawierającego jakieś cenne informacje. Przed rozpoczęciem operacji zmiany nazwy bądź przenoszenia pliku warto sprawdzić przy użyciu polecenia `ls` czy w lokalizacji docelowej nie ma już pliku o takiej samej nazwie; alternatywnym rozwiązaniem może być również dołączenie do polecenia

mv opcji -i, która spowoduje, że przed nadpisaniem pliku docelowego system poprosi o potwierdzenie wykonania takiej operacji. Jeżeli zalogujemy się do systemu jako użytkownik *root*, to podobnie jak to miało miejsce w przypadku poleceń rm i cp, Linux automatycznie dołącza opcję -i do każdego polecenia mv.

Od czasu do czasu może się zdarzyć, że polecenie mv nie będzie w stanie przenieść danego katalogu z jednego urządzenia na inne. Jeżeli użytkownik spotka się z takim problemem, to powinien najpierw skopiować zawartość całego katalogu z lokalizacji źródłowej na docelową, a dopiero potem, po pomyślnym wykonaniu operacji kopiowania, usunąć pliki źródłowe i następnie katalog źródłowy.

Odszukiwanie zagubionych plików

Jeżeli użytkownik zna nazwę pliku, ale nie bardzo pamięta, w jakim katalogu taki plik jest (tudzież powinien być) zlokalizowany, to do jego odszukania może wykorzystać polecenie find. Przykładowo, wykonanie następującego polecenia:

```
[helion@localhost helion]$ find . -name 'instrukcja' - print
```

spowoduje próbę odszukania pliku o nazwie *zguba*, zlokalizowanego gdzieś poza bieżącym katalogiem użytkownika (.). Jeżeli próba taka zakończy się sukcesem, to na ekranie zostanie wyświetlona bezwzględna ścieżka dostępu do takiego pliku.

Jeżeli użytkownik zna jedynie fragment nazwy pliku, to poszukiwania pozostałych fragmentów można wymusić, zastępując je odpowiednio znakami gwiazdki (*), przykładowo:

```
[helion@localhost helion]$ find / -name '*truk*' - print
```

Przedstawione powyżej polecenie spróbuje odszukać dowolne pliki, których nazwy pasują do podanego wzorca poszukiwań; poszukiwania będą obejmowały cały obszar dysku zajęty przez katalog główny /, czyli tak naprawdę cały dostępny obszar dysku.

Innym, bardzo przydatnym poleceniem, które może wziąć na siebie ciężar wyszukiwania danego pliku, jest locate. Polecenie locate wykorzystuje aktualizowaną raz dziennie bazę danych lokalizacji plików — z tego powodu polecenie to nie będzie w stanie odszukać plików, które zostały niedawno utworzone bądź usunięte. Jest to w pewnym sensie wadą tego programu, ale z drugiej strony, dzięki temu cały proces wyszukiwania żądanych plików jest dużo szybszy niż w wypadku polecenia find. Aby skorzystać z polecenia locate, należy jako jego argument podać ciąg znaków będących fragmentem nazwy poszukiwanego pliku. Podany ciąg znaków nie musi być ujęty w cudzysłów. Efektem działania polecenia będzie wyświetlenie na ekranie wszystkich plików (łącznie ze ścieżkami dostępu), których nazwy zawierają podany ciąg znaków. Przykładowo, wykonanie następującego polecenia:

```
[helion@localhost helion]$ locate pass
```

spowoduje wyświetlenie na ekranie listy wszystkich plików, których nazwa zawiera ciąg znaków *pass*.

Polecenie locate do działania wykorzystuje bazę danych utworzoną przez polecenie updatedb. Zanim będzie możliwe użycie po raz pierwszy polecenia locate, użytkownik musi zalogować się do systemu jako *root* i wykonać polecenie updatedb. Polecenie updatedb powinno być również wykonywane za każdym razem, kiedy nastąpią znaczące zmiany w systemie plików, aczkolwiek zadanie to powinny realizować za użytkownika usługi *cron* oraz *anacron*. Jeżeli jednak wyniki działania polecenia locate nie są zbyt aktualne, to użytkownik może uruchomić polecenie updatedb niezależnie od *cron-a*.

Drukowanie plików

Jeżeli do danego komputera podłączona jest poprawnie działająca drukarka, to użytkownik może wydrukować zawartość danego pliku korzystając z polecenia `lpr`. Przykładowo, wykonanie następującego polecenia:

```
[root@localhost /root]$ lpr /etc/passwd
```

spowoduje wydrukowanie pliku */etc/passwd*. Więcej informacji na temat konfiguracji drukarki można znaleźć w rozdziale 9.

W czasie kiedy drukarka jest zajęta drukowaniem jednego pliku, użytkownik może już wysyłać do kolejki wydruku następne pliki. Przeglądanie kolejki plików do wydruku jest możliwe dzięki poleceniu `lpq`:

```
[root@localhost root]# lpq
HP1150 is ready and printing
Rank    Owner   Job     File(s)                         Total Size
active  root    1       passwd                          2048 bytes
```

Każdy plik oczekujący na wydrukowanie ma przydzielony osobny numer zadania wydruku. Aby usunąć dane zadanie wydruku, należy wykonać polecenie `lprm`, podając jako argument numer zadania, które powinno zostać usunięte. Przykładowo, wykonanie następującego polecenia:

```
[root@localhost root]# lprm 155
```

spowoduje usunięcie zadania wydruku o numerze 155. Należy jednak pamiętać o tym, że zadanie wydruku może usunąć wyłącznie użytkownik będący właścicielem danego zadania (lub użytkownik *root*).

Praca z narzędziami do kompresji plików

Aby zaoszczędzić całkiem sporo przestrzeni na dyskach twardych oraz przyspieszyć pobieranie plików z serwerów sieciowych, pliki danych mogą zostać poddane procesowi kompresowania. Zgodnie z przyjętą w systemie Linux konwencją, nazwy skompresowanych plików mają rozszerzenie *.gz*; należy jednak pamiętać, że jest to jedynie przyjęta konwencja — z technicznego punktu widzenia system Linux ani nie wymaga, ani nie wymusza stosowania takiego rozszerzenia.

Aby rozpakować skompresowany plik, należy użyć polecenia `gunzip`. Przykładowo, załóżmy, że plik *dane.gz* został (jak wskazuje na to jego zgodne z przyjętą konwencją rozszerzenie) uprzednio skompresowany. Aby go rozpakować, należy wykonać poniższe polecenie:

```
[helion@localhost helion]$ gunzip dane.gz
```

Wykonanie powyższego polecenia spowoduje rozpakowanie pliku *dane* oraz usunięcie pliku *dane.gz*.

Aby skompresować wybrany plik, należy użyć polecenia `gzip`. Przykładowo, aby skompresować plik o nazwie *dane*, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ gzip dane
```

Wykonanie powyższego polecenia spowoduje utworzenie skompresowanego pliku *dane.gz* oraz usunięcie pliku *dane*.

Innym poleceniem służącym do kompresowania plików jest *bzip2*, które wykorzystuje nieco inny algorytm kompresji danych; pliki pakowane przy użyciu tego programu mają zwyczajowo rozszerzenie *.bz2* i mogą być rozpakowywane przy użyciu polecenia *bunzip2*.

W praktyce bardzo często przydaje się możliwość przechowywania wielu plików (a nawet zawartości całych katalogów) spakowanych w jeden plik archiwum. Przykładowo, w opisany sposób można dosyć efektywnie tworzyć kopie bezpieczeństwa bądź kopie archiwalne danych. W systemie Linux do tworzenia plików archiwum, przechowujących wewnątrz wiele innych plików, służy polecenie `tar`. W przeciwieństwie do polecenia *gzip*, polecenie *tar* w żaden sposób nie modyfikuje oryginalnych plików. Aby utworzyć plik archiwum[1], należy wykonać następujące polecenie:

```
tar -cvf nazwa_pliku_archiwum lista-plików-lub-katalogów
```

gdzie

`nazwa_pliku_archiwum` to nazwa pliku docelowego, który chcemy utworzyć, natomiast *`lista-plików-lub-katalogów`* to lista oddzielonych od siebie spacjami nazw plików i(lub) katalogów, które chcemy umieścić w pliku archiwum. Podając nazwy plików i katalogów można posłużyć się również ścieżkami do nich (zarówno względnymi, jak i bezwzględnymi). Zgodnie z przyjętą w systemie Linux konwencją, pliki archiwum utworzone przy użyciu polecenia tar mają rozszerzenie *.tar*; należy jednak pamiętać, że podobnie jak to miało miejsce w przypadku plików *.gz*, jest to jedynie przyjęta konwencja — z technicznego punktu widzenia system Linux ani nie wymaga, ani nie wymusza stosowania takiego rozszerzenia.

Przykładowo, jeżeli chcemy utworzyć plik archiwum o nazwie *kopia_danych.tar*, który będzie zawierał całą zawartość katalogu domowego użytkownika *helion* (łącznie z zawartością wszystkich podkatalogów), to należy wydać następujące polecenie:

```
[helion@localhost helion]$ tar -cvf kopia_danych.tar /home/helion
```

Wykonanie powyższego polecenia utworzy plik *kopia_danych.tar* w bieżącym katalogu roboczym.

Aby wyświetlić na ekranie zawartość danego pliku archiwum, należy wykonać następujące polecenie:

```
tar -tvf nazwa_pliku_archiwum | less
```

Przekierowanie strumienia danych z wyjścia polecenia `tar` ma wejście polecenia `less` powoduje, że w przypadku archiwum zawierającego dużą ilość plików wewnątrz, lista plików będzie mogła być przeglądana ekran po ekranie. Jeżeli dane archiwum zawiera niewielką ilość plików, to fragment `| less` może zostać pominięty.

Aby rozpakować zawartość pliku archiwum, należy wykonać następujące polecenie:

```
tar -xvf nazwa_pliku_archiwum
```

Wykonanie powyższego polecenia spowoduje rozpakowanie plików i katalogów stanowiących zawartość archiwum do bieżącego katalogu roboczego danego użytkownika.

Jeżeli plik lub katalog o danej nazwie już istnieje, to polecenie `tar` po prostu bez ostrzeżenia nadpisze go, zastępując jego oryginalną zawartość danymi rozpakowanymi z archiwum.

Polecenie `tar` posiada cały szereg bardzo przydatnych opcji; zainteresowanych odsyłamy do strony podręcznika *man* dla tego polecenia.

[1] W literaturze angielskiej pliki archiwum utworzone przy pomocy polecenia *tar* nazywane są potocznie *tarfiles*. W literaturze polskiej spotyka się czasem określenie *tarpliki*, aczkolwiek wydaje się, że pojęcie *pliki archiwum* jest tutaj bardziej na miejscu — *przyp. tłum.*

Powszechnie stosowanym rozwiązaniem jest kompresowanie zawartości plików archiwum, co można osiągnąć podając w poleceniu `tar` ciąg opcji `-czvf` zamiast standardowego `-cvf`. Zgodnie z przyjętą w systemie Linux konwencją, skompresowane pliki archiwum otrzymują rozszerzenie *.tgz*. Aby rozpakować skompresowany plik archiwum, należy w poleceniu `tar` podać ciąg opcji `-xzvf` zamiast standardowego `-xvf`.

Podczas kompresowania danych polecenie `tar` nie korzysta ze znanej w świecie systemu Windows metody ZIP; nie zmienia to jednak w żaden sposób faktu, że w systemie Linux można bez żadnego problemu otwierać i tworzyć pliki ZIP.

Aby utworzyć archiwum ZIP przechowujące skompresowane pliki i katalogi, należy wykonać następujące polecenie:

```
zip -r nazwa_archiwum_zip lista-plików-lub-katalogów
```

gdzie

`nazwa_archiwum_zip` to nazwa docelowego pliku ZIP, który chcemy utworzyć, natomiast *`lista-plików-lub-katalogów`* to lista oddzielonych od siebie spacjami nazw plików i(lub) katalogów, które chcemy umieścić w pliku ZIP.

Aby rozpakować zawartość archiwum ZIP, należy wykonać następujące polecenie:

```
unzip nazwa_archiwum_zip
```

Tworzenie dowiązań symbolicznych

Użytkownicy, którzy mieli okazję zetknąć się wcześniej z systemem Windows, zapewne zetknęli się z tzw. skrótami do plików, katalogów czy programów, które pozwalają na odwoływanie się do takiego elementu przy użyciu kilku skrótów o różnych nazwach. Skróty pozwalają również na odwoływanie się do danego pliku z wielu różnych katalogów. W systemie Linux podobne rezultaty można osiągnąć wykorzystując polecenie `ln`, które potrafi „podłączyć" kilka różnych nazw do jednego pliku bądź katalogu. Takie „podłączenia" nazywane są *dowiązaniami symbolicznymi* (ang. *symbolic links*); spotyka się również określenia *dowiązanie miękkie* (ang. *soft link*) bądź po prostu *dowiązanie*.

Aby utworzyć dowiązanie symboliczne do danego pliku bądź katalogu, należy wykonać następujące polecenie:

```
ln -s nazwa_pliku nazwa_dowiązania
```

Przykładowo, załóżmy, że w bieżącym katalogu roboczym znajduje się plik o nazwie *helion*. Aby można było się do tego pliku odwoływać przy użyciu nazwy *wydawnictwo*, należy utworzyć odpowiednie dowiązanie symboliczne, wykonując następujące polecenie:

```
[helion@localhost helion]$ ln -s helion wydawnictwo
```

Aby sprawdzić wyniki działania powyższego polecenia, skorzystamy z polecenia `ls -l`:

```
[helion@localhost helion]$ ls -l
razem 4
-rw-r--r--  1 root   root   600 sie 14 13:09 helion
lrwxrwxrwx  1 helion helion   6 sie 14 13:09 wydawnictwo -> helion
```

Jak widać, nowoutworzony plik o nazwie *wydawnictwo* jest typu `l`, co oznacza, że jest to plik dowiązania symbolicznego. Co więcej, polecenie `ls` pomaga użytkownikowi w identyfikacji dowiązań, wyświetlając nazwę pliku, do którego odwołuje się dowiązanie (`wydawnictwo -> helion`). Warto zwrócić uwagę na rozmiar pliku *wydawnictwo*. Utworzenie dowiązania

symbolicznego jest równoznaczne z utworzeniem wskaźnika do danego pliku oryginalnego; nie ma to nic wspólnego z tworzeniem kopii pliku — dzięki takiemu rozwiązaniu można zaoszczędzić całkiem sporą ilość miejsca na dysku.

Jeżeli podczas tworzenia dowiązania pominiemy opcję `-s`, to system Linux utworzy tzw. *dowiązanie twarde* (ang. *hard link*). Dowiązanie twarde musi być przechowywane w tym samym systemie plików, gdzie jest zlokalizowany plik, do którego ono się odwołuje; jest to ograniczenie, jakiemu nie podlegają miękkie dowiązania symboliczne. Ilość dowiązań do danego pliku lub katalogu wyświetlana przez polecenie `ls` odpowiada ilości dowiązań twardych, dowiązania symboliczne są po prostu ignorowane. W praktyce dowiązania twarde są używane bardzo rzadko, główne ze względu na fakt, że dowiązania symboliczne są o wiele bardziej elastyczne i nie podlegają tak wielu ograniczeniom.

Nadawanie praw dostępu do plików

Jak już wspominaliśmy w rozdziale 4., prawa dostępu do plików odpowiadają za to, jakie operacje na pliku bądź katalogu może wykonywać dany użytkownik. W tabeli 7.5 przedstawiono listę możliwych uprawnień oraz zamieszczono krótkie opisy każdego z nich. W rozdziale 4. wspominaliśmy również, że przypadku katalogów prawa dostępu działają nieco inaczej niż w przypadku plików. Przykładowo, `r` — czyli prawo do odczytu (ang. *read*) — w przypadku katalogu oznacza możliwość wyświetlenia listy przechowywanych w tym katalogu plików, natomiast w przypadku pliku oznacza możliwość przeczytania zawartości pliku. Zarówno dla katalogów, jak i plików można przydzielić po kilka różnych praw dostępu; użytkownik może wykonywać tylko i wyłącznie takie operacje, do jakich wykonywania dostał uprawnienia. Przykładowo, użytkownik, który do danego pliku ma prawa `rw` — czyli prawa do odczytu i zapisu (ang. *read write*) — może odczytywać zawartość pliku, może także modyfikować i ponownie zapisywać jego zawartość, ale nie będzie mógł wykonać tego pliku jako np. skryptu, na co wskazuje brak prawa `x` (ang. *eXecute;* prawo do wykonywania pliku). Warto w tym miejscu wrócić do rysunku 7.3 i przypomnieć sobie, w jaki sposób polecenie `ls` wyświetla prawa dostępu do plików i katalogów.

Tabela 7.5. Zestawienie praw dostępu do plików i katalogów

Oznaczenie prawa dostępu	Efektywne prawa dla katalogu	Efektywne prawa dla pliku
`r`	Wyświetlanie listy plików przechowywanych w katalogu	Odczytywanie zawartości pliku
`w`	Tworzenie lub usuwanie plików	Zapisywanie zawartości pliku
`x`	Dostęp do plików i podkatalogów	Wykonywanie pliku (np. skryptów)

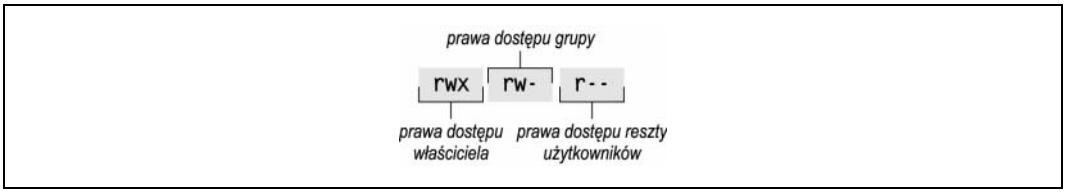
Prawa dostępu do plików i katalogów definiowane są indywidualnie dla każdego z trzech poziomów dostępu:

Użytkownik (Właściciel)
: Określa prawa dostępu dla właściciela pliku lub katalogu.

Grupa
: Określa prawa dostępu dla wszystkich użytkowników będących członkami grupy, która posiada prawo własności danego pliku lub katalogu.

Inni
: Określa prawa dostępu dla wszystkich pozostałych użytkowników.

Jak to zostało przedstawione na rysunku 7.3, polecenie `ls` wyświetla prawa dostępu w drugiej kolumnie (licząc od lewej). Zestawienie praw dostępu składa się z 9 znaków: pierwsze trzy określają prawa dostępu dla właściciela pliku lub katalogu, kolejne trzy definiują prawa dostępu dla grupy użytkowników posiadającej prawa własności do tego pliku lub katalogu, wreszcie ostatnie trzy znaki określają prawa dostępu do danego pliku lub katalogu dla wszystkich pozostałych użytkowników (rysunek 7.4).



Rysunek 7.4. Przykładowe prawa dostępu dla pliku lub katalogu

Użytkownik może ustawiać prawa dostępu dla plików i katalogów korzystając z polecenia `chmod`, którego składnia jest następująca:

```
chmod nnn nazwa_katalogu_lub_pliku
```

Argument *nnn* oznacza trzycyfrową liczbę, której kolejne cyfry definiują prawa dostępu odpowiednio dla właściciela, grupy oraz pozostałych użytkowników. W tabeli 7.6 przedstawiono zestawienie dozwolonych wartości i odpowiadających im uprawnień. Przykładowo, argument o wartości 751 reprezentuje prawa dostępu postaci `rwxr-x--x`, które dają właścicielowi pełne prawa, grupie prawa odczytu i wykonywania a pozostałym użytkownikom tylko prawa wykonywania.

Tabela 7.6. Numeryczne odpowiedniki poszczególnych praw dostępu

Wartość	Prawa
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwx

Jeżeli użytkownik jest właścicielem danego pliku lub katalogu (bądź też loguje się jako użytkownik *root*), to może zmienić prawa własności danego pliku lub katalogu; można tego dokonać korzystając z polecenia `chown`. Przykładowo, wykonanie poniższego polecenia powoduje, że użytkownik *helion* staje się właścicielem pliku *rozdzial7*:

```
[root@localhost helion]$ chown helion rozdzial7
```

Jeżeli użytkownik jest właścicielem danego pliku lub katalogu (bądź też loguje się jako użytkownik *root*), to może zmienić grupę, która będzie miała prawa własności dodanego pliku lub katalogu; można tego dokonać korzystając z polecenia `chgrp`. Przykładowo, wykonanie poniższego polecenia powoduje, że grupa *helion* otrzymuje prawa własności pliku *rozdzial7*:

```
[root@localhost helion]$ chgrp helion rozdzial7
```

Grupa, która zostaje przypisana do danego pliku lub katalogu, musi oczywiście zostać najpierw utworzona przez użytkownika *root*; dodatkowo, jeżeli polecenie `chgrp` jest wykonywane przez użytkownika innego niż *root*, to taki użytkownik musi być członkiem grupy, do której zostaje przypisany dany plik. Definicje grup są zapisane w pliku */etc/group*, do którego modyfikacji ma prawo tylko użytkownik *root*. Użytkownik *root* może przypisać danego użytkownika do jednej lub kilku grup (w zależności od potrzeb). Po zalogowaniu się do systemu użytkownik jest domyślnie dodawany do jednej z grup — tzw. grupy logowania. Aby zmienić przynależność do grupy, można użyć polecenia `newgrp`. Przykładowo, aby zmienić identyfikator bieżącej grupy użytkownika na grupę *korektorzy*, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ newgrp korektorzy
```

Jeżeli użytkownik spróbuje przełączyć się na grupę, do której nie należy bądź która nie istnieje, to takie polecenie nie zostanie wykonane. Kiedy użytkownik tworzy nowy plik lub nowy katalog, to jego prawa własności są automatycznie przypisywane do bieżącej grupy użytkownika.

Uruchamianie programów

W systemie Linux, podobnie zresztą jak w systemach MS-DOS czy Microsoft Windows, programy są przechowywane w osobnych plikach. Bardzo często, aby uruchomić dany program, wystarczy wpisać jego nazwę w wierszu poleceń i nacisnąć klawisz *Enter*. Aby jednak było to możliwe, dany program musi być zapisany w jednym z katalogów zdefiniowanych w tzw. *ścieżce systemowej* (ang. *path*). O katalogu, którego nazwa jest dołączona do ścieżki systemowej mówimy potocznie, że jest dostępny *na ścieżce systemowej*. Jeżeli użytkownik pracował wcześniej z systemem MS-DOS bądź Windows, to z pewnością szybko zorientuje się, że Linux traktuje ścieżkę systemową w niemal identyczny sposób.

Jeżeli program, który chcemy uruchomić, nie jest dostępny na ścieżce systemowej, to aby go uruchomić, trzeba będzie poprzedzić jego nazwę pełną ścieżką dostępu. Przykładowo, jeżeli program taki jest zlokalizowany w bieżącym katalogu roboczym, to przed nazwą programu należy wpisać ciąg znaków ./ , co stanowi odwołanie do katalogu bieżącego. Dzięki podaniu ścieżki do programu, system Linux będzie w stanie go uruchomić nawet wtedy, kiedy program nie będzie dostępny na ścieżce systemowej.

Przykładowo, załóżmy, że program o nazwie *symulator* jest zlokalizowany w katalogu */home/helion*; katalog ten jest katalogiem bieżącym i w dodatku znajduje się na ścieżce systemowej. Użytkownik może w takiej sytuacji uruchomić ten program, wykonując jedno z trzech przedstawionych poniżej poleceń:

```
[helion@localhost helion]$ symulator
[helion@localhost helion]$ ./symulator
[helion@localhost helion]$ /home/helion/symulator
```

Pierwsze polecenie działa przy założeniu, że program jest dostępny na ścieżce systemowej. Drugie polecenie zakłada, że program znajduje się w bieżącym katalogu roboczym; wreszcie trzecie podaje wprost dokładną lokalizację wywoływanego programu.

Montowanie i odmontowywanie napędów

Aby zamontować wybrane urządzenie lub partycję dysku, należy użyć polecenia `mount`, którego składnia jest następująca:

```
mount opcje urządzenie katalog
```

Polecenie `mount` posiada bardzo wiele opcji, aczkolwiek najczęściej używa się go w postaci domyślnej, bez żadnych dodatkowych opcji; więcej szczegółowych informacji na temat tego polecenia można znaleźć na stronach podręcznika *man* polecenia `mount`.

Przyczyną, dla której można często używać polecenia `mount` bez żadnych dodatkowych opcji, jest fakt, że w pliku */etc/fstab* przechowywane są opisy wszystkich dostępnych na danym komputerze napędów oraz informacje o systemach plików na nich przechowywanych. Kiedy po dołożeniu nowego urządzenia pamięci masowej ponownie uruchomimy komputer, usługa `kudzu` automatycznie wykryje takie urządzenie i zapisze jego charakterystykę w pliku */etc/fstab*. W razie potrzeby użytkownik może również przeglądać plik */etc/fstab* i ręcznie modyfikować jego zawartość.

Niezbędnymi argumentami polecenia `mount` są nazwa urządzenia, jakie chcemy zamontować, oraz nazwa katalogu, do którego będzie podpięte dane urządzenie, zwana inaczej *punktem montowania* (ang. *mount point*). Aby ułatwić i ujednolicić sposób dostępu do różnego rodzaju urządzeń, system Linux traktuje punkt montowania jak zwykły katalog; montowanie urządzenia powoduje jednoznaczne powiązanie go z daną nazwą katalogu. Przykładowo, do zamontowania napędu CD-ROM można użyć następującego polecenia:

```
[root@localhost root]# mount -t iso9660 /dev/cdrom /mnt/cdrom -o ro
```

Plik */dev/cdrom* jest niczym innym jak tylko dowiązaniem symbolicznym, które odwołuje się do pliku urządzenia reprezentującego systemowy napęd CD-ROM. Katalog */mnt/cdrom* został utworzony przez program instalacyjny systemu Linux; domyślnie właśnie ten katalog jest wykorzystywany jak punkt montowania dla napędów CD-ROM. Na większości płyt CD-ROM instalowany jest system plików *iso9660*, dlatego został on podany jako wartość argumentu `-t`. Ostatni argument polecenia, `-o ro`, oznacza, że system plików na zamontowanym urządzeniu jest przeznaczony tylko do odczytu (ang. *ro — read only*); inaczej mówiąc, system może odczytywać pliki z tego urządzenia, ale nie może na nim zapisywać. Jeżeli któryś z argumentów polecenia zostanie pominięty, to w większości przypadków system pobierze odpowiednie dane z pliku */etc/fstab*. Z tego powodu, napęd CD-ROM może zostać zamontowany przy użyciu skróconej wersji polecenia `mount`:

```
[root@localhost root]# mount /dev/cdrom
```

Po wykonaniu powyższego polecenia, użytkownik może odczytywać pliki i katalogi zapisane na dysku CD-ROM odwołując się po prostu do katalogu */mnt/cdrom*. Przykładowo, aby wyświetlić zawartość głównego katalogu dysku CD-ROM, należy wykonać następujące polecenie:

```
[root@localhost root]# ls /mnt/cdrom
```

Aby zamontować dyskietkę (zapisaną w formacie MS-DOS) włożoną do napędu `a:`, należy wykonać następujące polecenie:

```
[root@localhost root]# mount -t msdos /dev/fd0 /mnt/floppy
```

Aby odmontować dany napęd, należy podać jako argument polecenia `umount` przypisany do napędu punkt montowania (warto zwrócić uwagę, że nazwa polecenia to `umount` a nie `unmount`). Przykładowo, aby odmontować napęd CD-ROM, należy wykonać następujące polecenie:

```
[root@localhost root]# umount /mnt/cdrom
```

W zdecydowanej większości przypadków tylko użytkownik *root* może odmontować dane urządzenie. Mówimy, że „w zdecydowanej większości przypadków", ponieważ dystrybucja Red Hat Linux umożliwia montowanie i odmontowywanie urządzeń również zwykłym użytkownikom, pod warunkiem, że są zalogowani lokalnie. Niezależnie od tego, dowolne urządzenie może

być odmontowane jedynie wtedy, kiedy nie jest używane. Przykładowo, jeżeli bieżącym katalogiem roboczym danego użytkownika jest katalog zlokalizowany na danym urządzeniu, to takie urządzenie nie będzie mogło zostać odmontowane.

Jeżeli użytkownik nie może odmontować jakiegoś urządzenia, to zawsze warto sprawdzić wszystkie otwarte okna terminala, używane konsole wirtualne oraz okna aplikacji i upewnić się, czy któreś z nich nie korzysta z plików bądź katalogów zlokalizowanych na urządzeniu, które użytkownik usiłuje odmontować. Jeżeli tak się dzieje, to przed odmontowaniem należy zakończyć taką sesję terminala bądź konsoli, lub po prostu zmienić bieżący katalog roboczy na inny, nie związany z danym urządzeniem.

Formatowanie dyskietek

Zanim użytkownik będzie mógł zapisać na dyskietce jakieś pliki, musi ona najpierw zostać sformatowana. Poleceniem, przy pomocy którego system Linux można dokonać formatowania dyskietki, jest `fdformat`. Aby wykonać operację formatowania, należy wpisać nazwę polecenia `fdformat`, po której następuje argument definiujący daną stację dyskietek oraz pojemność samej dyskietki; listę dostępnych argumentów przedstawiono w tabeli 7.7.

Tabela 7.7. Zestawienie argumentów polecenia fdformat

Argument	Opis
`/dev/fd0`	Dyskietka 3,5"; napęd a: (pojemność 1,44 MB)
`/dev/fd0H1440`	Dyskietka 3,5"; napęd a: (pojemność 1,44 MB)
`/dev/fd1`	Dyskietka 3,5"; napęd b:(pojemność 1,44 MB)
`/dev/fd1H1440`	Dyskietka 3,5"; napęd b:(pojemność 1,44 MB)
`/dev/fd1H2880`	Dyskietka 3,5"; napęd b:(pojemność 2,88 MB)

Przykładowo, aby sformatować standardową dyskietkę 3,5", o pojemności 1,44 MB, włożoną do napędu `a:`, należy po zalogowaniu się jako użytkownik *root* wykonać następujące polecenie:

```
[root@localhost root]# fdformat /dev/fd0H1440
```

Polecenie `fdformat` wykonuje jedynie niskopoziomowe formatowanie dyskietki (ang. *low-level format*). Wynika stąd, że zanim sformatowana dyskietka będzie mogła zostać użyta, należy na niej umieścić jakiś system plików. Przykładowo, system plików MS-DOS może przydać się do wymiany danych pomiędzy komputerami pracującymi pod kontrolą systemu Linux a komputerami z systemem Windows. Aby umieścić system plików MS-DOS na sformatowanej dyskietce, należy wykonać następujące polecenie:

```
[root@localhost root]# mkdosfs /dev/fd0
```

Po sformatowaniu dyskietki i umieszczeniu na niej systemu plików, można przystąpić do jej montowania, a następnie zapisywania na niej plików.

Zanim dyskietka zostanie wyjęta z napędu, należy się upewnić, że została ona poprawnie odmontowana. Proces odmontowania zapewnia, że wszystkie dane, które mogły jeszcze oczekiwać na zapis na dyskietce, na pewno tam się znalazły; w innym przypadku, dane zapisywane na dyskietce mogą być niespójne, uszkodzone bądź po prostu nie dać się odczytać.

Przydatne polecenia systemu Linux

W niniejszej sekcji przedstawimy kilka programów (poleceń zewnętrznych), które mogą nieco ułatwić użytkownikowi pracę z systemem Linux. Opiszemy kilka programów, które udostępniają różne informacje na temat pracy systemu operacyjnego, a pod koniec rozdziału zajmiemy się zagadnieniami związanymi z obsługą prostego edytora tekstu *nano*.

Przeglądanie informacji o statusie systemu operacyjnego

System Linux daje użytkownikowi do dyspozycji całkiem sporo poleceń, które wyświetlają różnego rodzaju informacje o statusie systemu. W tabeli 7.8 przedstawiono zestawienie najczęściej używanych poleceń. Umiejętne zastosowanie wymienionych poleceń może pomóc w identyfikacji i rozwiązywaniu problemów z funkcjonowaniem systemu operacyjnego. Każde z wymienionych poleceń może być uruchamiane bez żadnych dodatkowych opcji, co nie zmienia faktu, że każde z nich posiada dosyć imponujący zestaw możliwych opcji i argumentów, rozszerzających i zmieniających funkcjonalność poszczególnych poleceń. Więcej informacji na ten temat można odnaleźć na stronach podręcznika *man* dla poszczególnych poleceń.

Tabela 7.8. Zestawienie przydatnych poleceń systemowych

Nazwa polecenia	Opis
`df`	Wyświetla ilość wolnego miejsca (wyrażoną w KB), jaka pozostała w poszczególnych, zamontowanych systemach plików.
`du`	Wyświetla informację o ilości miejsca na dysku (wyrażoną w KB), jakie zajmuje bieżący katalog roboczy oraz poszczególne podkatalogi. Jeżeli dodamy do polecenia opcję `-s`, to wyświetlona zostanie jedynie całkowita ilość zajmowanego przez ten katalog miejsca.
`free`	Wyświetla statystyki zajętości pamięci operacyjnej, m.in. ilość wolnej pamięci, ilość użytej pamięci, fizyczny rozmiar pamięci operacyjnej, ilość pamięci przechowywanej na partycji wymiany, ilość pamięci współdzielonej oraz ilość buforów używanych przez jądro systemu.
`ps`	Wyświetla informacje o aktywnych procesach działających w systemie, powiązanych z danymi sesjami logowania. Aby wyświetlić informacje o wszystkich działających procesach, należy dołączyć do polecenia opcję `-a`.
`top`	Wyświetla aktualizowane na bieżąco informacje o aktywnych procesach oraz wykorzystywanych przez nie zasobach systemowych. Aby zakończyć działanie polecenia, należy nacisnąć klawisz *q*.
`uptime`	Wyświetla bieżący czas, ilość czasu, jaki upłynął od zalogowania się użytkownika, liczbę zalogowanych użytkowników oraz informacje o średnim obciążeniu systemu.
`users`	Wyświetla nazwy użytkowników aktualnie zalogowanych do systemu.
`w`	Wyświetla sumaryczne informacje o obciążeniu systemu, zalogowanych użytkownikach oraz aktywnych procesach.
`who`	Wyświetla nazwy zalogowanych użytkowników, informacje, z jakich terminali korzystają, czas jaki upłynął od zalogowania się każdego z nich oraz nazwy komputerów, z których poszczególni użytkownicy logowali się do systemu.

Zastosowanie edytora nano

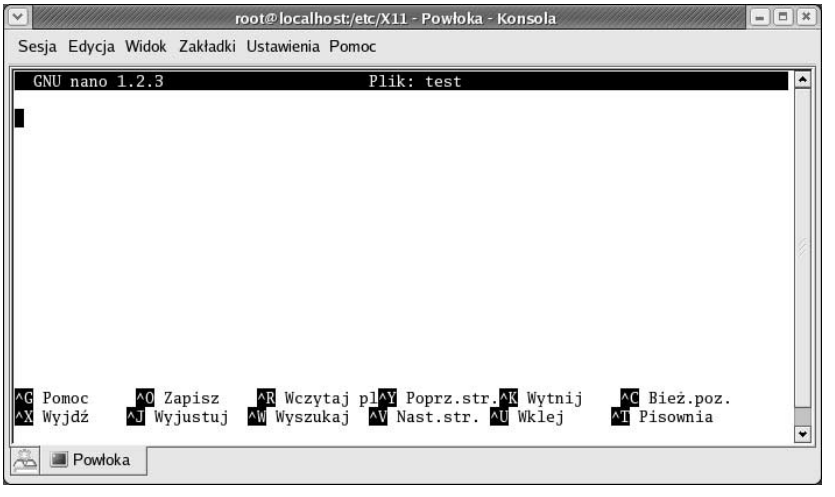
Jeżeli użytkownik korzysta ze środowiska graficznego X, to ma do wyboru cały szereg mniej lub bardziej rozbudowanych edytorów tekstu, dysponujących interfesjem GUI (ang. *Graphical User Interface*). Niestety, jeżeli pracujemy z konsolą wirtualną, to nici z uruchomienia jakiegokolwiek

edytora tekstu dysponującego interfejsem graficznym. Z pomocą przychodzi wtedy edytor `nano`, mały i prosty edytorek tekstu, który może być uważany za swego rodzaju odpowiednik znanego z systemu Windows programu *Edit*, ponieważ tak jak i on, może być uruchamiany zarówno w trybie tekstowym, jak i graficznym.

Aby uruchomić edytor `nano`, wystarczy po prostu wpisać jego nazwę w wierszu poleceń powłoki systemu Linux. Jeżeli użytkownik chce utworzyć nowy plik bądź otworzyć plik już istniejący, to w wierszu polecenia po nazwie edytora można podać nazwę pliku (jeżeli dany plik nie znajduje się w bieżącym katalogu roboczym, to dodatkowo należy podać pełną ścieżkę do takiego pliku). Przykładowo, aby utworzyć (albo otworzyć, jeżeli taki plik istnieje) plik o nazwie *moje_notatki*, należy wykonać następujące polecenie:

```
[helion@localhost helion]$ nano moje_notatki
```

Na rysunku 7.5 przedstawiono standardowy wygląd edytora `nano`. W pierwszym wierszu ekranu edytora wyświetlany jest wiersz statusu, z takimi informacjami jak wersja edytora oraz nazwa aktualnie edytowanego pliku. Jeżeli edytowany plik zostanie w jakikolwiek sposób zmodyfikowany, to w prawym górnym narożniku (w prawej części wiersza statusu) pojawi się słowo *Zmieniony*. Dwa dolne wiersze ekranu edytora wyświetlają listę dostępnych poleceń edycyjnych. Większość poleceń wymaga naciśnięcia jakiegoś dodatkowego klawisza sterującego (np. *Ctrl*), tak aby można było odróżnić polecenia od wpisywanego tekstu. Wpisywane znaki pojawiają się na ekranie od razu w miejscu, w którym znajduje się kursor (punkt wstawiania). Do zmiany położenia punktu wstawiania należy użyć klawiszy strzałek kursora, a do usuwania niepotrzebnych znaków — klawiszy *Backspace* oraz *Delete*. Niektóre polecenia wymagają podawania dodatkowych informacji, bądź same wyświetlają dodatkowe komunikaty — do tych celów wykorzystywany jest trzeci od dołu wiesz ekranu edytora.



Rysunek 7.5. Wygląd edytora tekstu nano

W tabeli 7.9 przedstawiono zestawienie poleceń edytora `nano`. Naciśnięcie kombinacji klawiszy *Ctrl+G* przywołuje na ekran tekst pomocy edytora `nano`. Niektóre polecenia i funkcje mogą być wywoływane na kilka sposobów, przykładowo, naciśnięcie klawisza *F1* daje taki sam efekt jak naciśnięcie kombinacji klawiszy *Ctrl+G* — na ekranie pojawi się tekst pomocy edytora `nano`.

Tabela 7.9. Zestawienie poleceń edytora nano

Polecenie	Opis
*Ctrl+^*	Ustawia początek zaznaczanego bloku tekstu w miejscu kursora
Ctrl+A	Przesuwa kursor do początku bieżącego wiersza
Ctrl+B	Przesuwa kursor w lewo o jeden znak
Ctrl+C lub F11	Wyświetla informacje o aktualnym położeniu kursora
Ctrl+D	Usuwa znak znajdujący się w miejscu aktualnego położenia kursora
Ctrl+E	Przesuwa kursor na koniec bieżącego wiersza
Ctrl+F	Przesuwa kursor o jeden znak w prawo
Ctrl+G lub F1	Wyświetla na ekranie tekst pomocy edytora `nano`
Ctrl+I	Wstawia tabulator w miejscu aktualnego położenia kursora
Ctrl+J lub F4	Formatuje bieżący akapit
Ctrl+K lub F9	Wycina zaznaczony blok tekstu
Ctrl+L	Odświeża ekran
Ctrl+N	Przesuwa kursor do następnego wiersza
Ctrl+O lub F3	Zapisuje do pliku aktualną zawartość okna edycyjnego
Ctrl+P	Przesuwa kursor do poprzedniego wiersza
Ctrl+R lub F5	W miejscu aktualnego położenia kursora wstawia zawartość innego pliku tekstowego
Ctrl+T	Wywołuje mechanizm sprawdzania pisowni
Ctrl+U lub F10	Wstawia tekst w miejscu aktualnego położenia kursora
Ctrl+V lub F8	Przesuwa kursor o jedną stronę tekstu do przodu
Ctrl+W lub F6	Włącza mechanizm wyszukiwania tekstu (nie zwraca uwagi na pisownię dużych i małych liter)
Ctrl+X lub F2	Kończy pracę edytora, pozwala na zapisanie do pliku zawartości okna edycyjnego
Ctrl+Y lub F7	Przesuwa kursor o jedną stronę tekstu do tyłu

Aby nabrać wprawy w posługiwaniu się edytorem `nano`, wykonajmy teraz proste ćwiczenie, które jednocześnie zapozna nas z podstawowymi funkcjami edycyjnymi tego programu. Zaczniemy od uruchomienia edytora:

```
[helion@localhost helion]$ nano
```

Na ekranie pojawi się okno edytora. Wpiszmy niewielki fragment tekstu, popełniając przy okazji kilka prostych błędów:

```
Nano to to świeetny edytor. Najczęściej wykorzystujemy go do edycji prostych
tekstowych plików. Kiedy jednak potrzebny będzie program o większych możliwościach, to
nie należy zapominać o edytorze vi!
```

Jak łatwo zauważyć, do powyższego akapitu wkradły się trzy błędy:

- Słowo *to* zostało niepotrzebnie powtórzone drugi raz.
- Słowo *świetny* zostało błędnie zapisane jako *świeetny*.
- W drugim zdaniu słowa *tekstowych* oraz *plików* zostały zamienione miejscami.

Aby poprawić pierwszy błąd, należy, posługując się klawiszami strzałek kursora, ustawić go na literze *t* drugiego słowa *to*, a następnie usunąć to słowo i następującą po nim spację naciskając trzy razy klawisz *Delete*.

Aby poprawić drugi błąd, należy, posługując się klawiszami strzałek kursora, ustawić go na literze *e* (obojętnie, pierwszej czy drugiej) słowa *świeetny*, a następnie usunąć nadmiarową literę raz naciskając klawisz *Delete*.

Teraz załóżmy, że przed słowem *świetny* chcemy dodatkowo wpisać słowo *naprawdę*. Aby tego dokonać, należy, posługując się klawiszami strzałek kursora, ustawić go na literze *ś* słowa *świetny*, następnie wpisać z klawiatury słowo *naprawdę*, dodając po nim jedną spację.

Wreszcie, aby poprawić trzeci błąd, skorzystamy z mechanizmu wycinania i wklejania tekstu do zamiany kolejności słów *tekstowych* oraz *plików*. Posługując się klawiszami strzałek kursora, przesuwamy go na pierwszą literę *t* słowa *tekstowych*. Naciskamy kombinację klawiszy *Ctrl*+^, co spowoduje rozpoczęcie procesu zaznaczania tekstu. Teraz, korzystając z klawisza strzałka w prawo, zaznaczamy całe słowo *tekstowych*, łącznie z następującą po nim spacją, i naciskając kombinację klawiszy *Ctrl+K* wycinamy zaznaczony fragment. Teraz, korzystając z klawisza strzałka w prawo, ustawiamy kursor na kropce po słowie *plików*, dopisujemy jedną spację i naciskamy kombinację klawiszy *Ctrl+U* wstawiając wycięte przed chwilą słowo *tekstowych*. Gotowe!

Teraz pozostało jedynie zapisanie poprawionej wersji tekstu w pliku na dysku. W tym celu naciskamy kombinację klawiszy *Ctrl+X*. W trzecim od dołu wierszu edytora pojawi się pytanie: *Zapisać zmieniony bufor? (ODPOWIEDŹ "Nie" SPOWODUJE UTRATĘ ZMIAN)* — aby zapisać plik naciskamy klawisz *T*. Na ekranie pojawi się komunikat *Nazwa pliku do zapisu:*. Wpisujemy żądaną nazwę pliku i naciskamy klawisz *Enter*. Aby sprawdzić, czy plik został poprawnie zapisany, możemy skorzystać z polecenia `less`.