

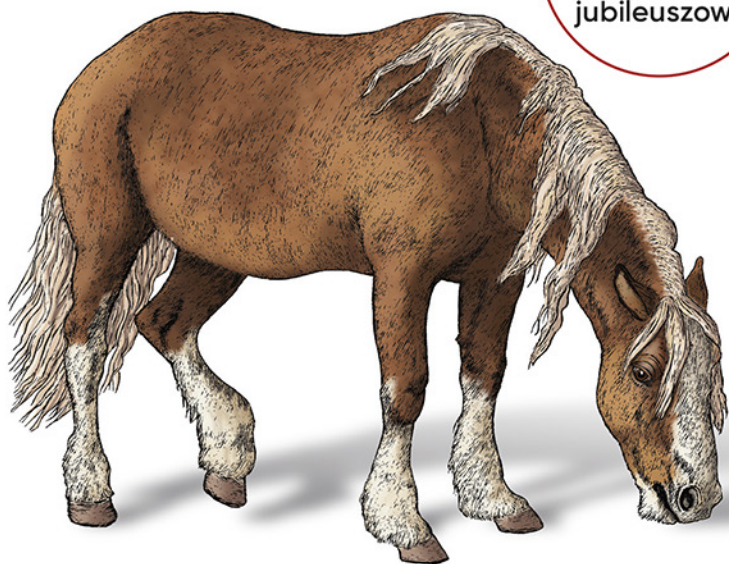
O'REILLY®

Wydanie IV

Linux

Leksykon kieszonkowy

20.
rocznica
wydanie
jubileuszowe



Daniel J. Barrett

Helion 

Tytuł oryginału: Linux Pocket Guide: Essential Commands, 4th Edition

Tłumaczenie: Robert Górczyński, na podstawie poprzedniego wydania
w przekładzie Przemysława Szeremioty

ISBN: 978-83-289-1971-6

© 2025 Helion S.A.

Authorized Polish translation of the English edition of *Linux Pocket Guide 4E*
ISBN 9781098157968 © 2024 Daniel J. Barrett.

This translation is published and sold by permission of O'Reilly Media, Inc.,
which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any
form or by any means, electronic or mechanical, including photocopying, recording
or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu
niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą
kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym,
magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź
towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje
były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich
wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych
lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności
za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/link4>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- **Kup książkę**
- **Poleć książkę**
- **Oceń książkę**

- **Księgarnia internetowa**
- **Lubię to! » Nasza społeczność**

Spis treści

Rozdział 1. Podstawowe koncepcje	15
Gdzie szukać pomocy?	56
Rozdział 2. Operacje na plikach	59
Podstawowe operacje na plikach	59
Operacje na katalogach	66
Przeglądanie plików	68
Tworzenie i edytowanie plików	76
Właściwości plików	82
Lokalizacja plików	95
Operacje na plikach tekstowych	104
Kompresowanie i pakowanie plików	123
Porównywanie plików	131
Konwersja plików na inne formaty	137
Pliki PDF i PostScript	147
Drukowanie	151
Sprawdzanie pisowni	153
Rozdział 3. Podstawy administrowania systemem	155
Jak zostać superużytkownikiem?	155
Przeglądanie procesów	157
Kontrola procesów	163
Planowanie zadań	168
Logowanie, wylogowanie i kończenie pracy	174
Użytkownicy i ich środowisko	177
Zarządzanie kontami użytkowników	181

Zarządzanie grupami	185
Instalowanie oprogramowania	188
Instalowanie oprogramowania z kodu źródłowego	200
Rozdział 4. System plików	205
Dyski i systemy plików	205
Tworzenie i modyfikowanie systemów plików	211
Macierz RAID oferująca nadmiarowość	215
Woluminy logiczne dla elastycznej pamięci masowej	221
ZFS — nowoczesny i wszechstronny system plików	227
Kopie bezpieczeństwa i zdalne przechowywanie	233
Rozdział 5. Sieć	241
Informacje o komputerze	241
Umieszczenie komputera	244
Połączenia sieciowe	248
Poczta elektroniczna na co dzień	254
Przeglądanie stron WWW	261
Rozdział 6. Wykonywanie codziennych zadań	267
Wyświetlanie na ekranie	267
Kopiowanie i wklejanie	273
Obliczenia matematyczne	275
Czas i data	280
System kontroli wersji	282
Kontenery	287
Wyświetlanie i przetwarzanie obrazów	291
Audio i wideo	294
Programowanie skryptów powłoki	301
Posłowie	318

Rozdział 1.

Podstawowe koncepcje

Czym jest Linux?

Linux jest popularnym systemem operacyjnym o otwartych źródłach, konkurującym z systemami Microsoft Windows i Apple macOS. Większość serwerów w internecie działa pod kontrolą Linuksa. Ponadto Linux znajduje się w tle każdego smartfona z systemem Android, komputera Chromebook oraz w milionach urządzeń sieciowych, takich jak routery, zapory sieciowe i automatyczne systemy dojenia krów (serio). Świetnie sprawdza się również jako system dla laptopa bądź komputera biurkowego.

Linux składa się z czterech głównych elementów (zob. rysunek 1.1), którymi są:

Jądro (kernel)

Jest to system operacyjny niskiego poziomu, zajmujący się obsługą sprzętu komputerowego oraz funkcji podstawowych, takich jak zarządzanie procesami i siecią komputerową. Większość użytkowników rzadko zauważa jego istnienie.

Dostarczone programy

Są tysiące programów przeznaczonych do obsługi plików, edycji tekstów, programowania, przeglądania zasobów internetu, obróbki audio i wideo, szyfrowania, obliczeń matematycznych... wybór jest niemal nieograniczony. Te programy komunikują się z jądrem. Programy uruchamiane w powłoce są nazywane *poleceniami*.

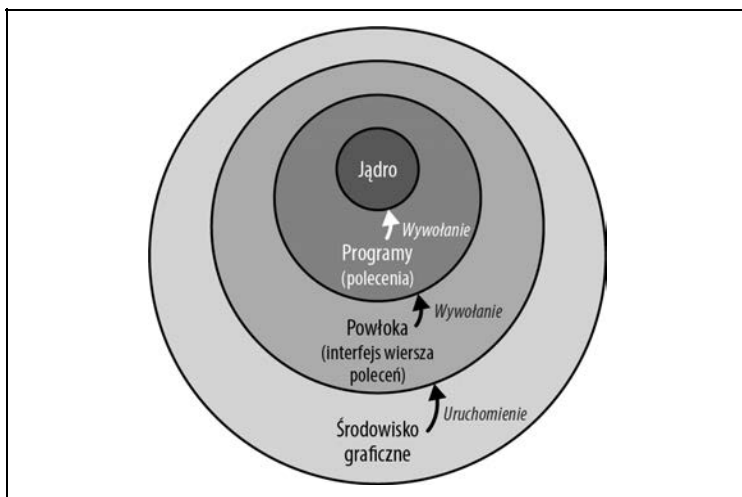
Powłoka (shell)

To interfejs użytkownika, pozwalający wpisywać polecenia, wykonywać je i wyświetlać wyniki. Linux oferuje całą gamę różnych powłok oferujących różne możliwości. W tej książce zajmiemy się

przede wszystkim powłoką **bash** (*Bourne Again Shell*), która bardzo często stosowana jest domyślnie w kontach użytkowników. Dostępnych jest wiele innych powłok, m.in. dash, fish, ksh (Korn shell), tcsh (TC shell lub T shell), zsh (Z shell), a także, do pewnego stopnia, busybox. Trzeba jednak pamiętać, że wszystkie istniejące powłoki oferują bardzo podobne funkcje, nawet jeśli praca z daną powłoką odbywa się nieco inaczej.

Środowisko graficzne (opcjonalnie)

Jest to system graficzny, pozwalający na obsługę okien, menu, ikon, myszy i innych elementów typowych dla interfejsów GUI (*Graphical User Interface* — Graficzny Interfejs Użytkownika). Najbardziej znane z nich to GNOME i KDE Plasma. Większość aplikacji opracowana dla GNOME może działać również w środowisku KDE i na odwrót¹.



Rysunek 1.1. Graficzne przedstawienie czterech głównych elementów tworzących Linuksa. Działające na niskim poziomie funkcje jądra są wywołane przez programy, które działają w powłoce, którą z kolei można uruchomić w środowisku graficznym

¹ GNOME, KDE i inne środowiska zostały zbudowane na podstawie systemu okien X Window lub Wayland. Aby sprawdzić, który z tych systemów jest używany, należy wydać polecenie `echo $XDG_SESSION_TYPE`.

W książce skupiamy się na drugim i trzecim z wymienionych elementów: dostarczonych programach i powłoce. Komputery z systemami Windows i macOS również mogą być obsługiwane z wiersza poleceń (w Windowsie za pomocą polecenia `cmd` i `powershell`, w macOS odpowiednikiem jest program Terminal), jednak większość ich użytkowników ogranicza się do używania interfejsu graficznego i może obejść się bez wpisywania poleceń. W Linuksie jednak powłoka jest elementem krytycznym. Jeżeli używasz Linuksa bez powłoki, dużo tracisz.

Linux to niesłychanie konfigurowalny system, zawierający tysiące programów. W rezultacie pojawiły się rozmaite odmiany Linuksa, które zaspokajają różne potrzeby i odpowiadają różnym gustom. Każda odmiana nosi nazwę *dystrybucji*. Mimo że mogą one wyglądać odmiennie i zawierać różne programy czy pliki, korzystają ze wspólnych komponentów. Do najpopularniejszych dystrybucji należą m.in. Mint, Ubuntu, Manjaro, Arch, Gentoo, Red Hat i OpenSUSE. W książce opisano zagadnienia wspólne dla nich wszystkich.

Uruchamianie powłoki

Skąd bierze się powłoka? Czasami Linux uruchamia ją automatycznie dla użytkownika. Tak się często dzieje po nawiązaniu połączenia z komputerem z systemem Linux za pośrednictwem sieci komputerowej z użyciem `ssh` bądź podobnego narzędzia. Wówczas po wejściu zobaczymy od razu powłokę gotową do wprowadzania poleceń.

W innych sytuacjach konieczne będzie ręczne uruchomienie powłoki. Na przykład użytkownicy zasiadający przed własnym komputerem najczęściej są witani przez pulpit graficzny z mnóstwem ikon i pozycji menu; próżno wtedy szukać powłoki na wierzchu. Wówczas potrzebna jest aplikacja graficzna, nazywana *terminalem* lub *programem terminala*, która pozwala uruchamiać powłokę w oknie. W znajdującej się nieco dalej ramce zatytułowanej „Powłoka kontra terminal” poznasz dokładną różnicę między powłoką i terminalem.

Każda dystrybucja wyposażona w środowisko graficzne zawiera również co najmniej jeden program terminala, choć być może trzeba będzie samemu go zlokalizować. Poszukaj aplikacji, ikony lub elementu menu

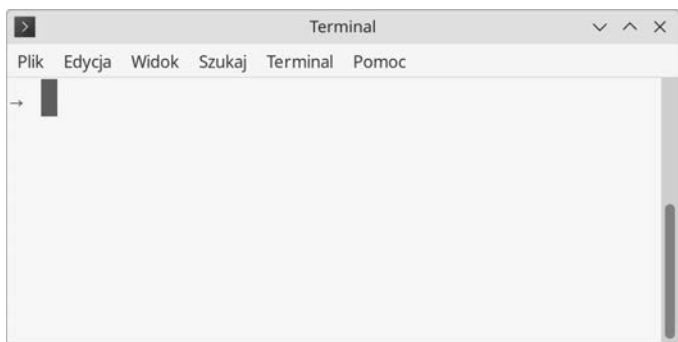
o nazwie Terminal, Konsole, xterm, gnome-terminal, uxterm lub czegoś podobnego. Uruchomienie tego programu otworzy okno powłoki. Ponadto możesz spróbować nacisnąć klawisze *Ctrl+Alt+t* (trzymając wciśnięte klawisze *Ctrl* i *Alt*, naciśnij klawisz *t*), co w niektórych środowiskach spowoduje uruchomienie terminala.

POWŁOKA KONTRA TERMINAL

Powłoka to interfejs wiersza poleceń przeznaczony do wykonywania poleceń systemu Linux wpisywanych w postaci zwykłego tekstu. Powłoka wyświetla znak zachęty i czeka na wpisanie polecenia:

→

Z kolei terminal to program wyświetlający okno, w którym została uruchomiona powłoka, jak możesz zobaczyć na rysunku 1.2. Jest to więc rodzaj graficznego opakowania dla powłoki. Terminal dodaje menu, paski przewijania, funkcjonalność kopiowania i wklejania, a także inne funkcje graficznego interfejsu użytkownika, które wspomagają działanie powłoki.



Rysunek 1.2. Program terminala wyświetla okno, w którym została uruchomiona powłoka

Rozgrzewka

Aby zasmakować możliwości wiersza poleceń, wypróbujemy teraz 10 prostych poleceń. Pamiętajmy o *dokładnym* ich wpisaniu, łącznie z wielkimi i małymi literami, odstępami i innymi symbolami (z wyjątkiem znaku

zachęty powłoki). Na końcu każdego polecenia należy nacisnąć klawisz *Enter*².

Kalendarz na listopad 2023 r.:

→ **cal lis 2023**

```
Listopad 2023
ni po wt śr cz pi so
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

Wykaz zawartości katalogu */bin* (zawierającego wiele poleceń systemowych):

→ **ls /bin**

```
bash      less      rm
bunzip2   lessecho rmdir
busybox   lessfile rnano
...
```

Zliczenie widocznych elementów katalogu domowego (reprezentowanego przez zawartość specjalnej zmiennej *HOME*, o której powiemy sobie więcej później):

→ **ls \$HOME | wc -l**

8 *U Ciebie ta wartość może być inna*

Sprawdzenie ilości zajętego miejsca na partycji głównego systemu plików na dysku:

→ **df -h /**

```
System plików  rozm.  użyte  dost.  % už.  zamont.  na
/dev/sdb1      78G    30G    48G    61%    /
```

Podgląd listy procesów uruchomionych na komputerze (naciśnij „q”, żeby przerwać):

→ **top -d1**

Wydrukowanie pliku */etc/hosts* (pliku z nazwami i adresami komputerów) na domyślnej drukarce (jeśli jest skonfigurowana):

→ **lpr /etc/hosts**

² Jeżeli otrzymasz komunikat błędu informujący o nieznalezieniu polecenia, nie przejmuj się, ponieważ prawdopodobnie dane polecenie nie zostało zainstalowane w Twoim systemie. W rozdziale 3. dokładnie wyjaśnimy, jak w systemie Linux odbywa się instalowanie oprogramowania.

Sprawdzenie czasu trwania sesji z systemem:

```
→ last -l $USER
```

```
kowalski pts/7 :0 Tue Nov 10 20:12 still logged in
```

Pobranie pliku o nazwie *sample.pdf* z podanej witryny internetowej i zapisanie go w katalogu bieżącym, bez konieczności sięgania po przeglądarkę internetową:

```
→ curl -O https://linuxpocketguide.com/sample.pdf
```

Sprawdzenie, kto jest właścicielem domeny *helion.pl* (wynik można przesuwać do przodu klawiszem spacji, podgląd kończymy, wpisując „q”):

```
→ whois helion.pl | less
```

```
Domain Name: helion.pl
```

```
Registrar: nazwa.pl
```

```
...
```

I na koniec wyczyszczenie zawartości terminala:

```
→ clear
```

Gratulacje, niniejszym staliśmy się użytkownikami powłoki systemu Linux!

Czym jest polecenie?

Typowe polecenie składa się z *nazwy programu* i następujących po niej *opcji* i *argumentów*, wpisywanych w wierszu poleceń powłoki, na przykład:

```
wc -l mojplik
```

Nazwa programu (*wc*, od ang. *word count*, czyli „licznik słów”) dotyczy samego programu, znajdującego się gdzieś na dysku (powłoka musi go odnaleźć i uruchomić). Opcje najczęściej rozpoczynają się znakiem myślnika, określają zachowanie programu. W omawianym przykładzie opcja (*-l*) określa, że program *wc* ma zliczać wiersze, a nie słowa. Argument (*mojplik*) wskazuje plik do otwarcia.

Polecenia mogą mieć wiele opcji i argumentów. Opcje do programu mogą być przekazywane pojedynczo albo zgrupowane za pojedynczym znakiem myślnika:

```
wc -l -w mojplik
```

Dwie opcje podane osobno

```
wc -lw mojplik
```

To samo co -l -w

Zdarza się jednak, że niektóre programy nie rozpoznają tego drugiego sposobu podawania opcji. Jednorazowe podanie wielu argumentów jest również dopuszczalne:

```
wc -l mojplik1 mojplik2      Zlicza wiersze w dwóch plikach
```

Opcje nie są w żaden sposób ustandaryzowane. W jednych programach mogą być poprzedzane pojedynczym znakiem myślnika (-), a inne programy dopuszczają również parę myślników i pełną słowną nazwę opcji (--lines); zdarzają się też jeszcze inne formaty. Co więcej, ta sama opcja (na przykład -l) w różnych programach może mieć zupełnie odmienne znaczenia. W programie wc opcja -l oznacza „wiersze w tekście”, ale już dla polecenia ls opcja -l oznacza „pełniejsze dane”. Bywa również tak, że w dwóch programach różne opcje mogą oznaczać to samo działanie, na przykład opcje -q i -s mogą oznaczać tzw. ciche działanie programu (z języka angielskiego „run quietly” lub „run silently”). Niektóre opcje mogą być uzupełniane o wartości, na przykład -s 10 (odstęp pomiędzy opcją a wartością nie zawsze jest obowiązkowy: -s10).

Argumenty poleceń też nie są ustandaryzowane; zazwyczaj odnoszą się do nazw plików wejściowych i wyjściowych, ale mogą także reprezentować inne dane, na przykład nazwy katalogów, nazwy użytkowników, nazwy hostów, adresy IP, wyrażenia regularne lub dowolne ciągi tekstowe.

Polecenie składające się z prostego programu razem z opcjami i argumentami jest nazywane *poleceniem prostym*. Spójrz na przykład tego rodzaju polecenia, wyświetlającego użytkowników zalogowanych na serwerze Linuksa³.

```
→ who
kowalski      :0      wrz 23 20:44
nowak         pts/0    wrz 15 13:51
malinowski    pts/1      wrz 22 21:15
kowalski      pts/2      wrz 22 21:18
```

Polecenia mogą uruchamiać wiele programów jednocześnie albo nawet je łączyć. Spójrz teraz na przykład polecenia łączącego dane wyjściowe programu who z danymi wejściowymi programu wc, którego działanie

³ Użytkownik kowalski został wymieniony dwukrotnie, ponieważ ma uruchomione jednocześnie dwie powłoki interaktywne.

polega na zliczaniu wierszy. W efekcie otrzymujemy liczbę wierszy w danych wyjściowych programu `who`:

```
→ who | wc -l  
4
```

Pionowa kreska, nazywana *potokiem* (ang. *pipe*), pozwala na powstanie połączenia między programami `who` i `wc`. Specjaliści od Linuksa cały czas korzystają z potoków.

Polecenia mogą zawierać również elementy znane z języków programowania, takie jak zmienne, konstrukcje warunkowe i pętle, co dokładniej będzie omówione w rozdziale 6. Dzięki temu polecenie może oznaczać „uruchom program, zapisz wynik działania do wskazanego pliku, a w przypadku wystąpienia błędów wyślij mi wiadomość e-mail z rezultatem uruchomienia programu”.

Użytkownicy i superużytkownicy

Linux jest wielodostępnym systemem operacyjnym, co oznacza, że wielu użytkowników może jednocześnie korzystać z tego samego komputera. Na danym komputerze każdy użytkownik identyfikowany jest za pomocą unikatowej *nazwy użytkownika*, takiej jak *kowalski* lub *fajnykolega*. Każdy z użytkowników posiada na własność (do pewnego stopnia) część systemu, w której może prowadzić własne prace (więcej informacji na ten temat znajdziesz w dalszej części rozdziału).

Poza tym w systemie istnieje specjalny użytkownik o nazwie *root*, czyli *superużytkownik*, *użytkownik uprzywilejowany* lub *administrator*, który posiada uprawnienia do wykonania dowolnej operacji. Może więc on tworzyć, modyfikować i usuwać dowolne pliki w systemie oraz uruchamiać każdy program. Możliwości działania zwykłych użytkowników są ograniczane: mogą oni co prawda uruchamiać większość programów, ale zazwyczaj nie mogą pracować z plikami należącymi do innych użytkowników.

Niektóre z poleceń opisywanych w tej książce mogą być wywoływane jedynie przez użytkownika uprzywilejowanego. Takie polecenia będziemy poprzedzali specjalnym poleceniem `sudo`:

```
→ sudo      Tutaj polecenie wymagające specjalnych uprawnień
```

Ostrzeżenie

Polecenie `sudo` zapewnia potężne możliwości, dzięki którym można nawet zniszczyć system Linux.

Polecenie `sudo` omówimy osobno w sekcji „Uprawnienia superużytkownika” w rozdziale 3. Na razie wystarczy zapamiętać, że polecenie `sudo` daje użytkownikowi supermoce (czasami też wymaga podania hasła). Spróbujmy dla przykładu policzyć wiersze tekstu w chronionym systemowym pliku `/etc/shadow` z użyciem i bez użycia `sudo`:

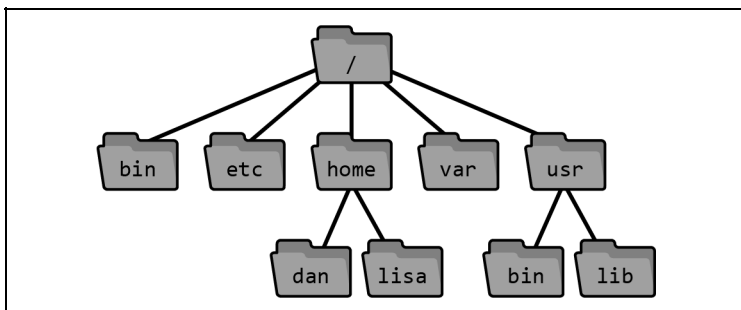
```
→ wc -l /etc/shadow      Nie uda się uruchomić polecenia
wc: /etc/shadow: Permission denied
→ sudo wc -l /etc/shadow  Tym razem się uda
[sudo] password: xxxxxxxx
51 /etc/shadow             Zadziałało!
```

System plików

Aby skutecznie używać Linuksa, trzeba dobrze zapoznać się z plikami systemowymi i katalogami (zwanymi też folderami), które razem są określane mianem *systemu plików*. W systemach okienkowych pliki i katalogi są widoczne na ekranie. W systemach opartych na linii poleceń, takich jak powłoka Linuksa, te same pliki i katalogi oczywiście istnieją, ale nie są stale widoczne, dlatego czasem użytkownik musi pamiętać, w którym katalogu się „znajduje” i jakie jest jego miejsce w hierarchii katalogów. Do przemieszczania się między katalogami oraz lokalizowania położenia służą polecenia `cd` (ang. *change directory*, pol. *zmień katalog*) i `pwd` (ang. *print working directory*, pol. *wyświetl bieżący katalog roboczy*).

Teraz garść informacji na temat terminologii. Jak wspomniano wcześniej, pliki w systemie Linux przechowywane są w *katalogach*. Katalogi tworzą hierarchię lub *drzewo* (rysunek 1.3): jeden katalog może zawierać inne katalogi, nazywane *podkatalogami*, a one z kolei mogą zawierać inne katalogi lub pliki i tak dalej, aż do nieskończoności. Najwyższy katalog nazywany jest *katalogiem głównym* lub *korzeniem* (*root directory*) i oznaczany jest znakiem ukośnika (`/`)⁴.

⁴ W systemie Linux *wszystkie* pliki i katalogi wywodzą się z korzenia. Jest to koncepcja odmienna niż w systemach Windows, w których poszczególne urządzenia oznaczane są literami napędów.



Rysunek 1.3. System plików Linuksa (fragment). Katalog główny znajduje się u góry. Pełna ścieżka dostępu do katalogu daniel to `/home/daniel`

Pliki i katalogi opisujemy za pomocą sekwencji nazw i ukośników, które składają się w ciąg znaków, nazywany *ścieżką* (ang. *path*). Na przykład następująca ścieżka:

`/jeden/dwa/trzy/cztery`

oznacza katalog główny zawierający katalog o nazwie *jeden*, który z kolei zawiera katalog o nazwie *dwa*, który zawiera katalog o nazwie *trzy*, który zawiera katalog lub plik o nazwie *cztery*. Ścieżki zaczynające się od katalogu głównego nazywane są *ścieżkami bezwzględnymi*.

Nie wszystkie ścieżki muszą być ścieżkami bezwzględnymi; można też posługiwać się ścieżkami osadzonymi w katalogu innym niż katalog główny. Na rysunku 1.3 widzimy dwa różne katalogi o nazwie *bin*, których ścieżki bezwzględne to `/bin` i `/usr/bin`. Jeśli odnosimy się po prostu do „katalogu *bin*”, nie do końca wiadomo, o który z nich nam chodzi (w systemie mogłoby zresztą występować również wiele innych katalogów o takiej nazwie). Potrzebny jest dodatkowy kontekst. Za każdym razem, kiedy posługujemy się ścieżką niezaczynającą się od znaku ukośnika, np. *bin*, posługujemy się *ścieżką względną*.

Aby dookreślić położenie ścieżki względnej, musimy wiedzieć, w jakim katalogu „znajdujemy się” obecnie w systemie plików; katalog ten to tzw. bieżący katalog roboczy albo po prostu *katalog roboczy*.

Zawsze gdy uruchamiana jest powłoka, pracuje ona w jakimś katalogu (to oczywiście przenośnia), a kiedy wydawane są w niej polecenia, działają one względem tego właśnie katalogu. Na przykład jeżeli powłoka

znajduje się w katalogu `/usr` i wydane zostanie polecenie związane z katalogiem `bin`, to w rzeczywistości przetwarzany będzie katalog `/usr/bin`. Ogólnie względna ścieżka `a/b/c` dla katalogu roboczego `/jeden/dwa/trzy` oznaczać będzie rzeczywistą ścieżkę `/jeden/dwa/trzy/a/b/c`.

Powłoka rozpoznaje dwa specjalne katalogi: `.` (pojedyncza kropka) i `..` (podwójna kropka). Pierwszy z nich oznacza katalog aktualny, a drugi — katalog *nadrzędny*, czyli katalog o jeden poziom wyżej. W związku z tym, jeżeli aktualnym katalogiem jest `/jeden/dwa/trzy`, to zapis `.` oznacza ten właśnie katalog, a zapis `..` oznacza katalog `/jeden/dwa`.

Powłokę można „przesuwać” z jednego katalogu do drugiego za pomocą polecenia `cd`, które przechodzi do wskazanego katalogu roboczego.

→ `cd /usr/local/bin`

Przejdzie do katalogu `/usr/local/bin`

W przedstawionym tutaj poleceniu została użyta bezwzględna ścieżka dostępu. Oczywiście możliwe jest też wykonywanie zmian względnych:

→ `cd d`

Wejście do podkatalogu `d`

→ `cd ../mojkatalog`

Przejdzie do katalogu nadrzędnego, a następnie do katalogu `mojkatalog`

Nazwy katalogów i plików mogą zawierać w sobie większość z dostępnych znaków: wielkie i małe litery⁵, cyfry, kropki, myślniki, podkreślniki i większość innych symboli (z wyjątkiem znaku ukośnika, ponieważ jest on używany do oddzielania katalogów). W celu efektywnej pracy z powłoką należy unikać stosowania spacji, gwiazdek, symbolu dolara (`$`), nawiasów i innych znaków, posiadających jakieś znaczenie w powłoce. Przy zastosowaniu takich znaków konieczne jest zamykanie nazw plików w cudzysłowach lub stosowanie znaków ucieczki (ang. *escape character*), zobacz sekcję „Cytowanie”, co może być niewygodne i żmudne.

Katalogi domowe

Osobiste pliki użytkowników najczęściej znajdują się w katalogu `/home` (dla zwykłych użytkowników) lub w katalogu `/root` (dla superużytkownika). Katalog domowy użytkownika najczęściej ma postać `/home/twoja-nazwa-uzytkownika`, czyli `/home/kowalski`, `/home/nowak` itd. Istnieje kilka sposobów na zlokalizowanie i odnoszenie się do swojego katalogu domowego.

⁵ Te znaki nie są równoważne, ponieważ Linux odróżnia wielkie i małe litery.

cd

Polecenie `cd` bez żadnych argumentów powraca do katalogu domowego użytkownika (tzn. ustala aktualny katalog roboczy).

Zmienna HOME

Zmienna środowiskowa `HOME` (zobacz sekcję „Zmienne powłoki”) zawiera nazwę katalogu domowego użytkownika.

→ **echo \$HOME** *Polecenie echo wyświetla tutaj nazwę katalogu*
/home/kowalski
→ **cd \$HOME/linuxpocketguide** *Przejdź do podanego podkatalogu*

znak tylda (~)

Pojedynczy znak tyldy, jeżeli zostanie użyty w miejsce katalogu, zostanie zmieniony przez powłokę w pełną nazwę katalogu aktualnie zalogowanego użytkownika.

→ **echo ~** *Polecenie echo wyświetla tutaj nazwę katalogu*
/home/kowalski
→ **cd ~ /linuxpocketguide** *Przejdź do podanego podkatalogu*

Jeżeli za tyldą znajduje się nazwa użytkownika (na przykład `~kowalski`), powłoka rozwinie ten ciąg znaków w pełną nazwę katalogu domowego tego użytkownika:

→ **cd ~kowalski** *Przejdź do katalogu domowego użytkownika 'kowalski', o ile taki istnieje w systemie*
→ **pwd** *Polecenie wyświetla bieżący katalog roboczy*
/home/kowalski

Katalogi systemowe

Typowy system linuksowy posiada dziesiątki tysięcy katalogów systemowych. W katalogach tych znajdują się pliki systemu operacyjnego, aplikacji, dokumentacja, czyli właściwie wszystko, *oprócz* osobistych plików użytkowników (które zwyczajowo przechowywane są w katalogu `/home`).

Każdy, kto nie jest administratorem systemu, raczej rzadko będzie przeglądał katalogi systemowe — jednak posiadając pewną wiedzę, można zrozumieć lub zgadnąć przeznaczenie niektórych z nich. Nazwy tych katalogów często składają się z trzech części, tak jak na rysunku 1.4.



Rysunek 1.4. Zakres, kategoria i aplikacja katalogu

Ścieżka katalogu, część 1.: zakres

Zakres ścieżki katalogu na wysokim poziomie określa przeznaczenie całej hierarchii katalogów. Jedne z najczęściej spotykanych zakresów to:

- /** Pliki systemowe dostarczane z dystrybucją (katalog *główny*).
- /usr** Więcej plików systemowych dostarczanych z dystrybucją (katalog *user*).
- /usr/local** Pliki systemowe, które nie są dostarczane z dystrybucją. Mogą być one unikatowe dla sieci lokalnej Linuksa bądź dla danego komputera.

W praktyce nie istnieje wyraźna różnica między katalogami `/` i `/usr`, ale można domyślić się, że katalog `/` jest niejako na „niższym poziomie” i jako taki jest bliższy systemowi operacyjnemu.

Ścieżka katalogu, część 2.: kategoria

Kategoria definiuje typy plików, jakie można znaleźć w danym katalogu. Na przykład jeżeli kategorią jest *lib*, można spokojnie zakładać, że w tym katalogu będą się znajdować pliki bibliotek (ang. *library*) używane podczas programowania. Natomiast jeżeli kategorią jest *bin*, w tym katalogu zwykle znajdować będą się pliki binarne (ang. *binary*), czyli programy.

W związku z tym gdy kategorię typu *bin* poprzedzasz zakresem, otrzymujesz ścieżki takie jak `/bin`, `/usr/bin` i `/usr/local/bin`. Najbardziej podstawowe programy systemowe dystrybucji, np. `ls` i `cat`, zwykle znajdują się w katalogu `/bin`, zaś inne programy trafiają do `/usr/bin`⁶. Z kolei `/usr/local/bin` zawiera lokalnie zainstalowane programy, niewchodzące w skład dystrybucji. Oczywiście nie są to bezwzględne zasady, ale raczej typowe przypadki przechowywania programów.

⁶ W niektórych dystrybucjach Linuksa to rozróżnienie zostało już zatarte; na przykład w dystrybucji Fedora katalog `/bin` jest dowiązaniem symbolicznym do katalogu `/usr/bin`.

Oto najczęściej spotykane kategorie.

Kategorie programów

- bin** Programy (najczęściej pliki binarne).
- sbin** Programy (najczęściej pliki binarne) przewidziane do uruchamiania przez superużytkownika.
- lib** Biblioteki kodu używane przez programy.

Kategorie dokumentacji

- doc** Dokumentacja.
- info** Pliki dokumentacji wbudowanego systemu pomocy edytora Emacs.
- man** Pliki dokumentacji (strony podręcznika) wyświetlane przez program *man*; pliki są często skompresowane i przeplatane poleceniami interpretowanymi przez program *man*.
- share** Pliki związane z programem, takie jak przykłady i instrukcje dotyczące instalacji.

Kategorie konfiguracji

- etc** Pliki konfiguruje system (a także inne rozmaite rzeczy).
- init.d** Pliki konfiguracyjne wykorzystywane w czasie uruchamiania Linuksa.
- rc.d** Pliki konfiguracyjne wykorzystywane w czasie uruchamiania Linuksa; również pliki *rc1.d*, *rc2.d*, ...

Kategorie programowania

- include** Pliki nagłówkowe dla tworzonych programów.
- src** Kod źródłowy programów.

Kategorie plików sieci WWW

- cgi-bin** Skrypty i programy działające na stronach WWW.
- html** Strony sieci WWW.
- public_html** Strony sieci WWW, najczęściej umieszczane w katalogach domowych użytkowników.
- www** Strony sieci WWW.

Kategorie wyświetlania

fonts	Czcionki.
X11	Pliki systemu X Window.

Kategorie sprzętu

dev	Pliki urządzeń udostępniające dyski i inne urządzenia.
media	Pliki urządzeń pozwalające na komunikację z dyskami i pozostałym sprzętem.
mnt	Punkty montowania: katalogi umożliwiające dostęp do dysków.

Kategorie środowiska wykonania programów

var	Pliki dotyczące danego komputera, tworzone w czasie jego pracy i często uaktualniane.
lock	Pliki blokujące tworzone przez programy w celu sygnalizowania swojego działania; istnienie pliku blokującego może powstrzymać inny program lub egzemplarz tego samego programu przed działaniem lub wykonaniem jakiejś akcji.
log	Pliki dzienników przechowujące informacje o ważnych zdarzeniach w systemie, błędach, ostrzeżeniach i komunikatach informacyjnych.
mail	Skrzynki pocztowe przechowujące przychodzące listy.
run	Pliki PID przechowujące identyfikatory działających procesów; te pliki są często przeglądane w celu znalezienia lub zabicia konkretnego procesu.
spool	Pliki kolejgowane lub przesyłane, na przykład poczta wychodząca, zadania wydruku i zadania zaplanowane.
tmp	Przechowalnia plików tymczasowych dostępna dla programów i użytkowników.

Ścieżka katalogu, część 3.: aplikacja

Część aplikacji w ścieżce katalogu (zob. rysunek 1.4), o ile istnieje, najczęściej jest nazwą programu. Na przykład katalog `/etc/systemd` ma zasięg / (katalog główny), kategorię `etc` (pliki konfiguracyjne) i aplikację `systemd`. Skoro `systemd` to usługa przeznaczona do konfiguracji Linuksa, można całkiem słusznie stwierdzić, że katalog `/etc/systemd` zawiera pliki konfiguracyjne dla tej usługi.

Katalogi systemu operacyjnego

Są to katalogi obsługujące jądro systemu, najniższy poziom systemu operacyjnego Linux.

/boot

Pliki potrzebne w czasie uruchamiania systemu. Tutaj przechowywane jest jądro systemu (kernel), najczęściej w postaci pliku pod nazwą */boot/vmlinuz* albo podobną.

/lost+found

Uszkodzone pliki uratowane przez narzędzie odzyskiwania danych z dysku.

/proc

Aktualnie uruchomione procesy; dla zaawansowanych użytkowników.

/sys

Pliki związane z działaniem jądra; dla zaawansowanych użytkowników.

Pliki w katalogach */proc* i */sys* umożliwiają wgląd w pracę jądra systemu; mają też pewne ciekawe właściwości. Każdy plik z katalogu */proc* zawsze ma zerową wielkość, jest plikiem tylko do odczytu i ma aktualną datę utworzenia. Mimo to zawierają one informacje na temat jądra systemu:

```
→ ls -lG /proc/version
```

```
-r--r--r-- 1 root 0 paź 3 22:55 /proc/version
```

```
→ cat /proc/version
```

```
Linux version 5.15.0-76-generic ...
```

Pliki w katalogu */sys* również mają mylącą wielkość i tajemniczą zawartość:

```
→ ls -lG /sys/power/state
```

```
-rw--r--r-- 1 root 4096 0 lip 8 06:12 /sys/power/state
```

```
→ cat /sys/power/state
```

```
freeze mem disk
```

Pliki w katalogach */proc* i */sys* są najczęściej wykorzystywane przez różne programy, ale nic nie stoi na przeszkodzie, żeby samemu sprawdzać ich zawartość. Oto kilka przykładów.

/proc/ioports Lista sprzętu wejścia-wyjścia zainstalowanego w komputerze.

/proc/cpuinfo Informacje o procesorach komputera.

/proc/version Wersja systemu operacyjnego. Polecenie `uname` wyświetla dokładnie te same informacje.

/proc/uptime Czas działania systemu, tzn. liczba sekund, jaka upłynęła od ostatniego uruchomienia systemu. Polecenie `uptime` podaje te same informacje w znacznie czytelniejszej postaci.

/proc/*nnn* Gdzie *nnn* to liczba całkowita dodatnia, np. `/proc/13542`. Te pliki zawierają informacje o działającym w systemie procesie z identyfikatorem *nnn*.

/proc/self Informacja o aktualnie wykonywanym procesie. Jest to symboliczne, automatycznie odświeżane, dowiązanie do pliku `/proc/nnn`. Aby się o tym przekonać, wystarczy kilkakrotnie wydać polecenie

→ **ls -l /proc/self**

— wskazanie pliku `/proc/self` za każdym razem będzie inne.

Zabezpieczenia plików

W systemie Linux może być zdefiniowanych wielu użytkowników, posiadających w nim swoje konta. Aby możliwe było zapewnienie prywatności i bezpieczeństwa systemu, większości użytkowników pozwala się na dostęp tylko do *części* plików, ale nigdy do wszystkich. Ten sposób kontroli dostępu może zostać zdefiniowany za pomocą dwóch pytań:

Kto posiada uprawnienia?

Każdy plik i katalog posiada swojego *właściciela*, który ma prawo zrobić z nim cokolwiek. Najczęściej właścicielem pliku jest użytkownik, który ten plik utworzył, choć superużytkownik może zmienić właścicielstwo pliku.

Dodatkowo dostęp do pliku może posiadać zdefiniowana wcześniej *grupa* użytkowników. Są one definiowane przez administratora systemu, co jest opisane w sekcji „Zarządzanie grupami” w rozdziale 3.

Do pliku lub katalogu mogą mieć też dostęp *wszyscy użytkownicy* posiadający konto w danym systemie. Ta grupa użytkowników jest określana mianem *pozostali* lub *inni*.

Jakie uprawnienia zostały przyznane?

Właścicielom plików, grupom użytkowników i pozostałym użytkownikom można przyznać uprawnienia do *odczytu* (*read*), *zapisu* lub *modyfikacji* (*write*) i *uruchamiania* (*execute*) poszczególnych plików. Uprawnienia dotyczą również katalogów, które użytkownicy mogą

odczytywać (czyli sprawdzać listę plików w katalogu), zapisywać (tworzyć i usuwać pliki w katalogu) i uruchamiać (wejść do katalogu za pomocą polecenia `cd`).

Aby sprawdzić, kto jest właścicielem pliku *mojplik* i jakie uprawnienia mamy do pliku, należy wydać polecenie (więcej informacji na ten temat znajdziesz w następnym rozdziale):

→ **ls -l *mojplik***

`-rw-r--r-- 1 kowalski kowalski 1168 paź 28 2015 mojplik`

Aby sprawdzić, kto jest właścicielem katalogu *mojkatalog* i jakie uprawnienia mamy do tego katalogu, należy wydać polecenie:

→ **ls -ld *mojkatalog***

`drwxr-x--- 3 kowalski kowalski 4096 sty 08 15:02 mojkatalog`

Uprawnienia do pliku definiowane są przez dziesięć pierwszych znaków w ciągu wygenerowanym przez polecenie `ls`, tworzących zbiór liter `r` (odczyt), `w` (zapis), `x` (uruchomienie) i innych liter oraz myślników. Na przykład:

`-rwxr-x---`

Poniżej przedstawiono znaczenie poszczególnych liter i znaków w tym ciągu.

Pozycja	Znaczenie
1	Typ pliku: <code>-</code> = plik, <code>d</code> = katalog, <code>l</code> = dowiązanie symboliczne, <code>p</code> = potok nazwany, <code>c</code> = urządzenie znakowe, <code>b</code> = urządzenie blokowe
2 – 4	Uprawnienia właściciela pliku do jego odczytu, zapisu i uruchamiania
5 – 7	Uprawnienia grupy właściciela pliku do jego odczytu, zapisu i uruchamiania
8 – 10	Uprawnienia wszystkich użytkowników do jego odczytu, zapisu i uruchamiania

Powyższy przykład (`-rwxr-x---`) należy zatem rozumieć następująco: właściciel może odczytać, zapisać i wykonać plik; członkowie grupy mogą go odczytać i wykonać; pozostali użytkownicy nie mają do niego dostępu. Aby zmienić właściciela, grupę właściciela lub uprawnienia do pliku, należy użyć poleceń `chown`, `chgrp` i `chmod`. Zostaną one opisane w sekcji „Właściwości plików” w następnym rozdziale.

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Linuksowy adminie! Trzymaj tę książkę tuż obok swojego komputera!

Administratorzy systemów linuksowych mogą pracować za pomocą myszy, korzystając z interfejsu graficznego, jednak pełny potencjał Linuksa ujawnia się dzięki pracy z wierszem poleceń. Umiejętność używania tego narzędzia przydaje się każdemu, komu zależy na efektywnym działaniu systemu.

To czwarte, gruntownie zaktualizowane wydanie cenionego leksykonu, pomyślanego jako zwarte źródło informacji, z którego w razie potrzeby możesz szybko skorzystać. Znajdziesz tu omówienie ponad 200 poleceń systemu Linux — w tym nowych, przeznaczonych do obsługi plików, zarządzania pakietami, pracy z systemem kontroli wersji, przeprowadzania konwersji między formatami plików — i wiele więcej! Najużyteczniejsze polecenia Linuksa zostały pogrupowane według funkcjonalności, a treści ułożone tak, aby ułatwić i przyspieszyć znajdowanie potrzebnych informacji. Niezależnie od tego, czy zaliczasz się do początkujących, czy do zaawansowanych użytkowników Linuksa, ten praktyczny leksykon w każdej sytuacji będzie idealnym źródłem wiedzy o najważniejszych poleceniach tego systemu!

W książce:

- podstawowe koncepcje — polecenia, powtórki, użytkownicy i system plików
- operacje na plikach
- podstawy administrowania systemem
- zarządzanie systemem plików
- praca z hostami, połączenia sieciowe, poczta elektroniczna i strony WWW
- wykonywanie codziennych zadań

Daniel J. Barrett od ponad 30 lat zajmuje się Linuksem i powiązаныmi z nim technologiami. Jest autorem i współautorem licznych książek technicznych. Bywał inżynierem oprogramowania, administratorem systemów, wykładowcą uniwersyteckim, projektantem witryn internetowych, wokalistą w zespole heavymetalowym i satyrykiem. Obecnie pracuje w firmie Google.

Helion



helion.pl



HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej!



ISBN 978-83-289-1971-6



Cena: 59,90 zł