

WYDANIE II

LINUX

WPROWADZENIE
DO WIERSZA POLECEŃ

WILLIAM SHOTTS



Tytuł oryginału: The Linux Command Line, 2nd Edition: A Complete Introduction

Tłumaczenie: Joanna Zatorska, na podstawie książki Linux. Wprowadzenie do wiersza poleceń w tłumaczeniu Joanny Zatorskiej i Przemysława Szeremioty

ISBN: 978-83-283-6762-3

Copyright © 2019 by William Shotts. Title of English-language original: The Linux Command Line, 2nd Edition: A Complete Introduction, ISBN 978-1-59327-952-3, published by No Starch Press.

Polish-language edition copyright © 2020 by Helion SA. All rights reserved.

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 Helion SA 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 Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/linwp2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

PODZIĘKOWANIA	21
WSTĘP	23

Część I **Nauka powłoki**

I	
CZYM JEST POWŁOKA?	31
Emulatory terminali	31
Pierwsze kroki	32
Historia poleceń	32
Przesuwanie kursora	32
Pierwsze polecenia	33
Kończenie sesji terminala	35
Podsumowanie	35

2

NAWIGACJA	37
Hierarchia systemu plików	37
Bieżący katalog roboczy	38
Wypisywanie zawartości katalogu	39
Zmianie bieżącego katalogu roboczego	39
Ścieżki bezwzględne	39
Ścieżki względne	40
Przydatne skróty	42
Podsumowanie	42

3

PRZEGLĄD SYSTEMU	43
Więcej zabawy z ls	43
Opcje i argumenty	44
Długi format pod lupą	45
Sprawdzanie typu pliku za pomocą polecenia type	46
Wyświetlanie zawartości pliku za pomocą polecenia less	47
Wycieczka z przewodnikiem	49
Dowiązania symboliczne	51
Dowiązania twarde	52
Podsumowanie	52

4

MANIPULOWANIE PLIKAMI I KATALOGAMI	53
Wieloznaczniki	54
mkdir — tworzenie katalogów	56
cp — kopiowanie plików i katalogów	56
Przydatne opcje i przykłady	57
mv — przenoszenie plików i zmiana ich nazw	58
Przydatne opcje i przykłady	58
rm — usuwanie plików i katalogów	58
Przydatne opcje i przykłady	59
ln — tworzenie dowiązań	60
Dowiązania twarde	60
Dowiązania symboliczne	61
Budowanie placu zabaw	61
Tworzenie katalogów	62
Kopiowanie plików	62
Przenoszenie plików i zmiana ich nazw	63
Tworzenie dowiązań twardych	64
Tworzenie dowiązań symbolicznych	65
Usuwanie plików i katalogów	66
Podsumowanie	68

5

POLECENIA	69
Czym właściwie są polecenia?	70
Identyfikowanie poleceń	70
type — wyświetlanie typu polecenia	70
which — wyświetlanie lokalizacji pliku wykonywalnego	71
Pobieranie dokumentacji polecenia	71
help — uzyskiwanie pomocy dla poleceń wbudowanych w powłokę	71
--help — wyświetlanie informacji o użyciu	72
man — wyświetlanie podręcznika programu	73
apropos — wyświetlanie odpowiednich poleceń	74
whatis — wyświetlanie jednowierszowego opisu podręcznika polecenia	75
info — wyświetlanie informacji o programie	75
README i inne pliki dokumentacji programu	76
Tworzenie własnych poleceń z wykorzystaniem polecenia alias	77
Podsumowanie	79

6

PRZEKIEROWANIA	81
Standardowy strumień wejścia, wyjścia oraz błędów	82
Przekierowanie standardowego strumienia wyjścia	82
Przekierowanie standardowego strumienia błędów	84
Przekierowanie standardowego strumienia wyjścia i standardowego strumienia błędów do jednego pliku	85
Usuwanie niepotrzebnych danych wynikowych	86
Przekierowanie standardowego strumienia wejścia	86
cat — łączenie plików	86
Potoki	88
Filtry	89
uniq — zgłaszanie lub pomijanie powtarzających się wierszy	90
wc — wypisywanie liczników wierszy, słów oraz bajtów	90
grep — wypisywanie wierszy pasujących do wzorca	91
head (tail) — zwracanie początku (końca) pliku	92
tee — pobieranie danych ze standardowego strumienia wejścia, przekazywanie ich do standardowego strumienia wyjścia i do plików	93
Podsumowanie	94

7

SPOJRZENIE NA ŚWIAT Z PUNKTU WIDZENIA POWŁOKI	95
Interpretacja poleceń	95
Interpretacja ścieżek	96
Interpretacja tyldy	98
Interpretacja wyrażeń arytmetycznych	98
Interpretacja nawiasów	99
Interpretacja parametrów	101
Podstawianie wyników poleceń	101

Cytowanie	102
Cudzysłowy podwójne	103
Cudzysłowy pojedyncze	105
Interpretowanie znaków	105
Sekwencje z lewym ukośnikiem	105
Podsumowanie	106

8

ZAAWANSOWANE SZTUCZKI ZWIĄZANE Z KLAWIATURĄ 107

Edytowanie wiersza poleceń	108
Przemieszczanie kursora	108
Modyfikowanie tekstu	108
Wycinanie i wklejanie tekstu	109
Uzupełnianie	110
Korzystanie z historii	111
Przeszukiwanie historii	112
Interpretacja historii	113
Podsumowanie	114

9

UPRAWNIENIA 115

Właściciele, członkowie grupy i wszyscy pozostali	116
Odczyt, zapis i wykonywanie	117
chmod — zmiana trybu pliku	119
Ustawianie trybu pliku z poziomu interfejsu graficznego	122
umask — ustawianie uprawnień domyślnych	123
Niektóre uprawnienia specjalne	124
Zmiana tożsamości	125
su — uruchamianie powłoki z identyfikatorem zastępczego użytkownika i grupy	126
sudo — wykonywanie polecenia jako inny użytkownik	127
chown — zmiana właściciela pliku i grupy	129
chgrp — zmiana przypisania do grupy	130
Ćwiczenia dotyczące własnych uprawnień	130
Zmiana własnego hasła	132
Podsumowanie	133

10

PROCESY 135

Jak działa proces?	136
Wyświetlanie procesów	136
Dynamiczne wyświetlanie procesów za pomocą polecenia top	138
Sterowanie procesami	139
Zatrzymywanie procesu	141
Umieszczanie procesu w tle	141

Przywracanie procesu do pierwszego planu	142
Zatrzymywanie (wstrzymywanie) procesu	142
Sygnały	143
Wysyłanie sygnałów do procesów za pomocą polecenia kill	144
Wysyłanie sygnałów do wielu procesów za pomocą polecenia killall	146
Wyłączanie systemu	146
Więcej poleceń dotyczących procesów	147
Podsumowanie	148

Część II

Konfiguracja i środowisko

11

ŚRODOWISKO	151
Co jest przechowywane w środowisku?	152
Przeglądanie środowiska	152
Niektóre ciekawe zmienne	154
W jaki sposób konfigurowane jest środowisko?	154
Czym jest plik startowy?	156
Modyfikowanie środowiska	157
Które pliki należy zmodyfikować?	157
Edytory tekstu	158
Korzystanie z edytora tekstu	158
Aktywowanie naszych zmian	161
Podsumowanie	162

12

ŁAGODNE WPROWADZENIE DO VI	163
Dlaczego należy się nauczyć vi?	163
Krótkie wprowadzenie	164
Uruchamianie i zatrzymywanie vi	164
Tryby edycji	166
Włączanie trybu edycji	167
Zapisywanie pracy	167
Zmiana położenia kursora	168
Podstawowa edycja	169
Dodawanie tekstu	169
Otwieranie wiersza	170
Usuwanie tekstu	170
Wycinanie, kopiowanie i wklejanie tekstu	172
Łączenie wierszy	173

Szukanie i zastępowanie	173
Przeszukiwanie wiersza	173
Przeszukiwanie całego pliku	174
Wyszukiwanie i zastępowanie globalne	174
Edycja wielu plików	176
Przełączanie między plikami	176
Otwieranie do edycji dodatkowych plików	177
Kopiowanie treści z jednego pliku do drugiego	178
Wstawianie treści całego pliku do drugiego pliku	179
Zapisywanie zmian	179
Podsumowanie	180

13

DOSTOSOWYWANIE ZNAKU ZACHĘTY	181
Anatomia znaku zachęty	181
Alternatywne projekty znaków zachęty	182
Dodawanie koloru	184
Przesuwanie kursora	186
Zapisywanie znaku zachęty	187
Podsumowanie	188

Część III

Popularne zadania i podstawowe narzędzia

14

ZARZĄDZANIE PAKIETAMI	191
Systemy zarządzania pakietami	192
Jak działają systemy zarządzania pakietami?	192
Pliki pakietu	192
Repozytoria	193
Zależności	193
Narzędzia zarządzania pakietami wysokiego i niskiego poziomu	194
Popularne zadania zarządzania pakietami	194
Szukanie pakietu w repozytorium	194
Instalowanie pakietu z repozytorium	195
Instalowanie pakietu z wykorzystaniem pliku pakietu	195
Usuwanie pakietu	196
Uaktualnianie pakietów z repozytorium	196
Uaktualnianie pakietów za pomocą pliku pakietu	196
Wyświetlanie zainstalowanych pakietów	197
Sprawdzanie, czy pakiet jest zainstalowany	197
Wyświetlanie informacji o zainstalowanym pakiecie	198
Sprawdzanie, który pakiet zainstalował plik	198
Podsumowanie	198

15

NOŚNIKI DANYCH	201
Montowanie i odmontowywanie urządzeń pamięciowych	202
Wyświetlanie listy zamontowanych systemów plików	203
Ustalanie nazwy urządzenia	206
Tworzenie nowych systemów plików	210
Manipulowanie partycjami z wykorzystaniem fdisk	210
Tworzenie nowego systemu plików z wykorzystaniem mkfs	212
Testowanie i naprawa systemów plików	213
Przenoszenie danych bezpośrednio do urządzeń oraz z urządzeń	214
Tworzenie obrazów dysków CD	215
Tworzenie obrazu kopii dysku CD	215
Tworzenie obrazu na podstawie zbioru plików	216
Zapisywanie obrazów CD	216
Bezpośrednie montowanie obrazu ISO	216
Opróżnianie zapisywalnego dysku CD	216
Zapisywanie obrazu	217
Podsumowanie	217
Dodatkowe informacje	217

16

ZAGADNIENIA SIECIOWE	219
Sprawdzanie i monitorowanie sieci	220
ping	220
traceroute	221
ip	222
netstat	223
Przenoszenie plików poprzez sieć	224
ftp	224
lftp — ulepszony ftp	226
wget	226
Bezpieczna komunikacja z hostami zdalnymi	226
ssh	227
scp i sftp	230
Podsumowanie	232

17

SZUKANIE PLIKÓW	233
locate — łatwy sposób szukania plików	234
find — trudny sposób wyszukiwania plików	235
Testy	236
Operatory	237
Predefiniowane akcje	240
Akcje zdefiniowane przez użytkownika	242

Zwiększanie wydajności	243
xargs	244
Powrót do placu zabaw	245
Opcje polecenia find	247
Podsumowanie	247

18

ARCHIWIZACJA I KOPIE ZAPASOWE 249

Kompresowanie plików	250
gzip	250
bzip2	252
Archiwizacja plików	253
tar	253
zip	258
Synchronizacja plików i katalogów	261
Korzystanie z polecenia rsync poprzez sieć	264
Podsumowanie	265

19

WYRAŻENIA REGULARNE 267

Co to są wyrażenia regularne?	268
grep	268
Metaznaki i literały	270
Znak dowolny	270
Kotwice	271
Wyrażenia w nawiasach i klasy znaków	272
Zaprzeczenie	273
Tradycyjne zakresy znaków	273
Klasy znaków POSIX	274
Podstawowy POSIX a rozszerzone wyrażenia regularne	277
Alternatywa	279
Kwantyfikatory	280
? — dopasowuje element zero lub jeden raz	280
* — dopasowuje element zero lub więcej razy	281
+ — dopasowuje element raz lub więcej razy	281
{} — dopasowuje element określoną liczbę razy	282
Zapręgamy wyrażenia regularne do pracy	283
Sprawdzanie listy telefonicznej za pomocą polecenia grep	283
Szukanie brzydkich nazw plików z wykorzystaniem polecenia find	284
Wyszukiwanie plików za pomocą polecenia locate	284
Wyszukiwanie tekstu za pomocą programów less i vim	285
Podsumowanie	286

20

PRZETWARZANIE TEKSTU 287

Zastosowanie tekstu	288
Dokumenty	288
Strony WWW	288
E-mail	288
Wyjście drukarki	288
Kod źródłowy programów	289
Ponowne odwiedzin u starych przyjaciół	289
cat	289
sort	291
uniq	297
Cięcie i krojenie	299
cut — usuwanie fragmentów z każdego wiersza plików	299
paste — łączenie wierszy w pliku	302
join — łączenie dwóch plików na podstawie wspólnego pola	303
Porównywanie tekstu	305
comm — porównywanie dwóch posortowanych plików wiersz po wierszu	305
diff — porównywanie plików wiersz po wierszu	306
patch — dołączanie do oryginału pliku z różnicami	309
Edycja w locie	310
tr — transliterowanie lub usuwanie znaków	310
sed — edytor strumieniowy służący do filtrowania i przekształcania tekstu	312
aspell — interaktywny program do sprawdzania pisowni	320
Podsumowanie	323
Dodatkowe informacje	323

21

FORMATOWANIE WYNIKÓW 325

Proste narzędzia formatowania	325
nl — wstawianie numerów wierszy	326
fold — zawijanie każdego wiersza do określonej długości	329
fmt — prosty program do formatowania tekstu	329
pr — formatowanie tekstu do druku	332
printf — formatowanie i wypisywanie danych	333
Systemy formatowania dokumentów	336
groff	337
Podsumowanie	341

22

DRUKOWANIE 343

Krótką historia druku	344
Drukowanie w zamierzonych czasach	344
Drukarki oparte na znakach	344
Drukarki graficzne	345

Drukowanie w systemie Linux	346
Przygotowanie plików do druku	347
pr — przekształcanie plików tekstowych przeznaczonych do druku	347
Przesyłanie zadania drukowania do drukarki	348
lpr — drukowanie plików (styl Berkeley)	348
lp — drukowanie plików (styl Systemu V)	349
Inna opcja — a2ps	350
Monitorowanie zadań drukowania i sterowanie nimi	351
lpstat — wyświetlanie informacji o stanie drukarki	353
lpq — wyświetlanie statusu kolejki drukarki	353
lprm i cancel — anulowanie zadań drukowania	354
Podsumowanie	354

23

KOMPILOWANIE PROGRAMÓW 355

Czym jest kompilowanie?	356
Czy wszystkie programy są skompilowane?	357
Kompilowanie programu w języku C	358
Uzyskiwanie kodu źródłowego	358
Sprawdzanie zawartości drzewa źródłowego	360
Budowanie programu	361
Instalowanie programu	365
Podsumowanie	366

Część IV

Tworzenie skryptów powłoki

24

PISANIE PIERWSZEGO SKRYPTU 369

Czym są skrypty powłoki?	369
Jak napisać skrypt powłoki?	370
Format pliku skryptu	370
Uprawnienia do wykonywania	371
Lokalizacja pliku skryptu	371
Dobre lokalizacje dla skryptów	373
Więcej trików formatowania	373
Długie nazwy opcji	373
Wcięcia i kontynuacja wierszy	374
Podsumowanie	375

25

ROZPOCZYNIANIE PROJEKTU	377
Pierwszy etap — minimalny dokument	377
Drugi etap — dodawanie pewnych danych	380
Zmienne i stałe	380
Przypisywanie wartości do zmiennych i stałych	383
Dokumenty włączone	384
Podsumowanie	387

26

PROJEKTOWANIE ZSTĘPUJĄCE	389
Funkcje powłoki	390
Zmienne lokalne	393
Utrzymywanie działania skryptów	394
Podsumowanie	397

27

STEROWANIE PRZEPLÝWEM — ROZGAŁĘZIENIA IF	399
Instrukcje if	400
Status wyjścia	401
Korzystanie z testu	402
Funkcje plikowe	402
Funkcje tekstowe	405
Funkcje liczbowe	406
Nowocześniejsza wersja programu test	407
(()) — przeznaczone dla liczb całkowitych	409
Łączenie wyrażeń	410
Operatory sterowania — inny sposób rozgałęziania	412
Podsumowanie	413

28

ODCZYT WEJŚCIA Z KLAWIATURY	415
read — odczyt danych ze standardowego strumienia wejścia	416
Opcje	418
IFS	420
Weryfikacja wejścia	422
Menu	424
Podsumowanie	425
Dodatkowe informacje	426

29

STEROWANIE PRZEPŁYWEM — PĘTLE WHILE I UNTIL 427

Pętle	428
while	428
Uciezka z pętli	430
until	432
Odczyt plików za pomocą pętli	432
Podsumowanie	433

30

USUWANIE BŁĘDÓW 435

Błędy składniowe	435
Brakujące cudzysłowy	436
Brakujące lub niespodziewane tokeny	437
Nieprzewidziane interpretacje	437
Błędy logiczne	439
Programowanie defensywne	439
Uwaga na nazwy plików	441
Weryfikacja wejścia	442
Testowanie	442
Przypadki testowe	443
Debugowanie	444
Znalezienie miejsca problemu	444
Śledzenie	445
Sprawdzanie wartości podczas wykonywania	447
Podsumowanie	447

31

STEROWANIE PRZEPŁYWEM — ROZGAŁĘZIENIA CASE 449

Polecenie case	449
Wzorce	451
Wykonywanie wielu operacji	453
Podsumowanie	454

32

PARAMETRY POZYCYJNE 455

Dostęp do wiersza poleceń	455
Ustalanie liczby argumentów	457
shift — uzyskiwanie dostępu do wielu argumentów	457
Proste programy	459
Korzystanie z parametrów pozycyjnych wraz z funkcjami powłoki	459
Masowa obsługa parametrów pozycyjnych	460
Bardziej kompletne programy	462
Podsumowanie	465

33

STEROWANIE PRZEPEŁYWEM — PĘTLA FOR	469
for — tradycyjna forma powłoki	469
for — forma języka C	473
Podsumowanie	474

34

ŁAŃCUCHY TEKSTOWE I LICZBY	477
Interpretacja parametrów	477
Podstawowe parametry	478
Interpretacje służące do zarządzania pustymi zmiennymi	478
Interpretacje, które zwracają nazwy zmiennych	480
Operacje na łańcuchach tekstowych	481
Zmiana wielkości liter	484
Interpretacja wyrażeń arytmetycznych	486
Liczby o różnej podstawie	486
Operatory jednoargumentowe	487
Prosta arytmetyka	487
Przypisanie	488
Operacje bitowe	490
Logika	491
bc — język kalkulatora dowolnej precyzji	493
Korzystanie z bc	494
Przykładowy skrypt	495
Podsumowanie	496
Dodatkowe informacje	496

35

TABLICE	497
Czym są tablice?	497
Tworzenie tablic	498
Przypisywanie wartości do tablicy	498
Dostęp do elementów tablicy	499
Operacje na tablicach	501
Wyświetlanie zawartości całej tablicy	501
Określanie liczby elementów tablicy	502
Znajdowanie indeksów wykorzystanych przez tablicę	502
Dodawanie elementów na końcu tablicy	503
Sortowanie tablicy	503
Usuwanie tablicy	504
Tablice asocjacyjne	505
Podsumowanie	505

EGZOTYKA	507
Polecenia grupowe i podpowłoki	507
Substytucja procesu	511
Pułapki	513
Wykonywanie asynchroniczne za pomocą polecenia wait	516
Potoki nazwane	518
Ustawianie potoku nazwanego	519
Korzystanie z potoków nazwanych	519
Podsumowanie	519

3

Przegląd systemu



SKORO WIEMY JUŻ, JAK ORIENTOWAĆ SIĘ W SYSTEMIE PLIKÓW, MOŻEMY WYRUSZYĆ NA PODBÓJ SYSTEMU LINUX. NAJPIERW JEDNAK POZNAMY KILKA PRZYDATNYCH POLECEŃ:

- **ls** — wypisuje zawartość katalogu,
- **file** — określa typ pliku,
- **less** — wyświetla zawartość pliku.

Więcej zabawy z ls

ls jest prawdopodobnie najczęściej wykorzystywanym poleceniem i to nie bez powodu. Polecenie to służy do wyświetlenia zawartości katalogu wraz z informacją o różnych ważnych atrybutach plików i katalogów. Jak mogliśmy się już przekonać, aby wyświetlić listę plików i katalogów znajdujących się w bieżącym katalogu roboczym, wystarczy po prostu wpisać polecenie ls:

```
[me@linuxbox ~]$ ls
Desktop Documents Music Pictures Public Templates Videos
```

Nie musimy się ograniczać do bieżącego katalogu roboczego — możemy też wyświetlić zawartość innego katalogu podanego w następujący sposób:

```
me@linuxbox ~]$ ls /usr
bin games include lib local sbin share src
```

Możemy nawet uzyskać listę zawartości kilku katalogów. W tym przykładzie wyświetlimy zawartość katalogu domowego użytkownika (reprezentowanego przez symbol `~`) oraz katalogu `/usr`:

```
[me@linuxbox ~]$ ls ~ /usr
/home/me:
Desktop Documents Music Pictures Public Templates Videos
/usr:
bin games include lib local sbin share src
```

Możemy też zmienić format wyświetlonej listy i wzbogacić ją o więcej szczegółów:

```
[me@linuxbox ~]$ ls -l
total 56
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Desktop
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Documents
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Music
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Pictures
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Public
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Templates
drwxrwxr-x 2 me me 4096 2017-10-26 17:20 Videos
```

Po uruchomieniu polecenia z opcją `-l` uzyskaliśmy listę w długim formacie.

Opcje i argumenty

W tym momencie należy sobie zdać sprawę z bardzo ważnej zasady działania większości poleceń. Po wpisaniu polecenia zazwyczaj umieszcza się *opcję* lub kilka *opcji*, które modyfikują jego zachowanie. Następnie występuje *argument* lub kilka *argumentów*, czyli elementów przetwarzanych przez polecenie. Zgodnie z tym większość poleceń ma następującą składnię:

polecenie *-opcje argumenty*

Większość poleceń przyjmuje opcje składające się z jednego znaku poprzedzonego kreską, na przykład `-l`. Jednak wiele poleceń, także tych wchodzących w skład projektu GNU, akceptuje również *długie opcje* składające się ze słowa poprzedzonego dwiema kreskami. Ponadto w przypadku wielu poleceń kilka

krótkich opcji możemy zapisać w postaci jednego łańcucha. W poniższym przykładzie uruchamiamy polecenie `ls` z dwiema opcjami. Opcja `l` odpowiada za wyświetlenie listy w długim formacie, natomiast opcja `t` sortuje wynik na podstawie czasu modyfikacji pliku:

```
[me@linuxbox ~]$ ls -lt
```

Dodajmy teraz długą opcję `--reverse`, aby odwrócić kolejność sortowania:

```
[me@linuxbox ~]$ ls -lt --reverse
```

UWAGA

W systemie Linux w opcjach poleceń, podobnie jak w przypadku nazw plików, istotna jest wielkość liter.

Polecenie `ls` przyjmuje bardzo dużą liczbę opcji. Najczęściej wykorzystywane przedstawiono w tabeli 3.1.

Tabela 3.1. Opcje polecenia `ls`

Opcja	Długa opcja	Opis
-a	--all	Wypisuje wszystkie pliki, nawet te o nazwie rozpoczynającej się od kropki, które zwykle nie są wyświetlane (są to pliki ukryte).
-A	--almost-all	Działa jak opcja -a, ale nie wyświetla katalogu <code>.</code> (bieżący katalog), ani <code>..</code> (katalog nadrzędny).
-d	--directory	Zwykle jeśli podajemy nazwę katalogu, <code>ls</code> wyświetli zawartość katalogu, pomijając informacje o samym katalogu. Gdy użyjemy tej opcji razem z opcją <code>-l</code> , wyświetlone zostaną szczegóły samego katalogu, zamiast jego zawartości.
-F	--classify	Ta opcja dodaje na końcu każdej wyświetlonej nazwy znak wskaźnika. Przykładowo doda ukośnik (<code>/</code>), jeśli nazwa dotyczy katalogu.
-h	--human-readable	Na liście o długim formacie nie wyświetla rozmiarów plików w bajtach, lecz w formacie wygodnym do zrozumienia przez ludzi.
-l		Wyświetla listę w długim formacie.
-r	--reverse	Wyświetla listę w odwrotnej kolejności. Zwykle <code>ls</code> wyświetla wyniki w kolejności alfabetycznej rosnąco.
-S		Sortuje listę na podstawie rozmiaru plików.
-t		Sortuje listę na podstawie czasu modyfikacji.

Długi format pod lupą

Jak widzieliśmy, opcja `-l` sprawia, że polecenie `ls` wyświetla listę w długim formacie. Format ten zawiera dużą ilość przydatnych informacji. Poniżej przedstawiona jest lista zawartości katalogu *Examples* w systemie Ubuntu:

```

-rw-r--r-- 1 root root 3576296 2017-04-03 11:05 Experience ubuntu.ogg
-rw-r--r-- 1 root root 1186219 2017-04-03 11:05 kubuntu-leaflet.png
-rw-r--r-- 1 root root 47584 2017-04-03 11:05 logo-Edubuntu.png
-rw-r--r-- 1 root root 44355 2017-04-03 11:05 logo-Kubuntu.png
-rw-r--r-- 1 root root 34391 2017-04-03 11:05 logo-Ubuntu.png
-rw-r--r-- 1 root root 32059 2017-04-03 11:05 oo-cd-cover.odf
-rw-r--r-- 1 root root 159744 2017-04-03 11:05 oo-derivatives.doc
-rw-r--r-- 1 root root 27837 2017-04-03 11:05 oo-maxwell.odt
-rw-r--r-- 1 root root 98816 2017-04-03 11:05 oo-trig.xls
-rw-r--r-- 1 root root 453764 2017-04-03 11:05 oo-welcome.odt
-rw-r--r-- 1 root root 358374 2017-04-03 11:05 ubuntu Sax.ogg

```

Przeanalizujemy poszczególne pola jednego z plików i sprawdzimy ich opis w tabeli 3.2.

Tabela 3.2. Pola długiej listy polecenia *ls*

Pole	Znaczenie
-rw-r--r--	Uprawnienia dostępu do pliku. Pierwszy znak reprezentuje typ pliku. Znak kreski na początku oznacza zwykły plik, natomiast litera d oznacza katalog. Następne trzy znaki dotyczą uprawnień dostępu dla właściciela pliku, kolejne trzy znaki dotyczą uprawnień członków grupy przypisanej do pliku, a trzy ostatnie — uprawnień wszystkich pozostałych użytkowników. Pełne znaczenie tych znaków zostanie omówione w rozdziale 9.
1	Liczba twardych dowiązań do pliku. Więcej informacji na temat dowiązań znajduje się w podrozdziałach „Dowiązania symboliczne” i „Dowiązania twarde”.
root	Nazwa użytkownika (właściciela) pliku.
root	Nazwa grupy przypisanej do pliku.
32059	Rozmiar pliku w bajtach.
2017-04-03 11:05	Data i czas ostatniej modyfikacji pliku.
oo-cd-cover.odf	Nazwa pliku.

Sprawdzanie typu pliku za pomocą polecenia *type*

Podczas naszej wyprawy odkrywczej w głąb systemu przyda się nam wiedza o zawartości pliku. Do tego celu służy polecenie *file*, które pozwala określić typ pliku. Jak już wcześniej wspomniano, nazwy plików w Linuksie nie muszą odzwierciedlać zawartości pliku. Weźmy na przykład plik *picture.jpg*. Chociaż jego nazwa sugeruje zawartość w postaci skompresowanego obrazu JPEG, w Linuksie ta reguła nie obowiązuje. Polecenie *file* można uruchomić następująco:

```
file nazwa_pliku
```

Po uruchomieniu polecenie `file` wyświetli krótki opis zawartości pliku. Na przykład:

```
[me@linuxbox ~]$ file picture.jpg
picture.jpg: JPEG image data, JFIF standard 1.01
```

Istnieje wiele rodzajów plików. Właściwie jedną z podstawowych idei leżących u podstaw systemów uniksowych, takich jak Linux, jest to, że „wszystko jest plikiem”. Wraz z postępem naszej nauki przekonamy się, ile prawdy zawiera to twierdzenie.

Chociaż wiele plików w naszym systemie wygląda znajomo, na przykład pliki MP3 i JPEG, to jednak wiele pozostałych już nie, niektóre zaś mogą wydawać się dość dziwne.

Wyświetlanie zawartości pliku za pomocą polecenia `less`

Polecenie `less` to program służący do wyświetlania zawartości plików tekstowych. W systemie Linux istnieje wiele plików zawierających tekst zrozumiały dla człowieka. Program `less` pozwala na sprawdzenie ich zawartości w wygodny sposób.

Dlaczego mielibyśmy sprawdzać treść plików tekstowych? Dlatego, że w tym formacie jest przechowywanych wiele plików zawierających ustawienia systemu (są to *pliki konfiguracyjne*); możliwość ich odczytu daje nam wgląd w działanie systemu. Ponadto w formacie tekstowym jest zapisanych wiele programów wykorzystywanych przez system (zwanymi *skryptami*). W dalszych rozdziałach nauczymy się edytowania plików tekstowych, co pozwoli nam zmienić ustawienia systemu i pisać własne skrypty. Teraz tylko przejrzymy ich zawartość.

Polecenie `less` ma następującą składnię:

```
less nazwa_pliku
```

CZYM JEST „TEKST”?

Informacje na komputerze można zapisać na rozmaite sposoby, które wymagają zdefiniowania zależności między informacjami i liczbami służącymi do ich reprezentowania. Ostatecznie przecież komputery rozumieją tylko zapis liczbowy i wszystkie dane są przekształcane na reprezentację numeryczną.

Niektóre systemy reprezentacji są bardziej złożone (takie jak skompresowane pliki wideo), podczas gdy inne są dość proste. Jeden z najwcześniejszych i najprostszych systemów zwany jest *tekstem w formacie ASCII*. ASCII (wymawiany „as-ki”) jest skrótem od słów *American Standard Code for Information Interchange*.

Ten prosty schemat kodowania był na początku używany w dalekopisach w celu odwzorowania znaków na klawiszach na liczby.

Tekst jest prostym mapowaniem znaków na liczby „jeden do jednego”. Mapowanie to jest bardzo zwarte. Pięćdziesiąt znaków tekstu jest przekładane na pięćdziesiąt bajtów danych. Trzeba zrozumieć, że tekst zawiera zwykle mapowanie znaków na liczby. Nie jest to taki sam tekst, jak w dokumencie utworzonym przez procesor tekstu, taki jak Microsoft Word lub program Writer z pakietu LibreOffice. Pliki te w odróżnieniu od zwykłego tekstu ASCII zawierają wiele elementów niebędących tekstem, które służą do opisu ich struktury i formatowania. Pliki zwykłego tekstu ASCII zawierają same znaki oraz kilka podstawowych kodów sterowania, takich jak tabulatory, znaki powrotu karetki i nowego wiersza.

W systemie Linux wiele plików jest przechowywanych w formacie tekstowym, a wiele narzędzi linuksowych przetwarza pliki tekstowe. Nawet system Windows docenia wagę tego formatu. Dobrze znany program NOTEPAD.EXE jest edytorem plików zawierających zwykły tekst ASCII.

Po uruchomieniu programu `less` zawartość pliku tekstowego możemy przewijać w dół i w górę. Aby na przykład sprawdzić zawartość pliku, który definiuje wszystkie konta użytkowników systemu, należy wpisać następujące polecenie:

```
[me@linuxbox ~]$ less /etc/passwd
```

Gdy uruchomimy program `less`, możemy przeglądać treść pliku. Jeśli plik jest obszerniejszy niż jedna strona, możemy przewijać treść w górę i w dół. Aby zakończyć działanie programu `less`, należy nacisnąć klawisz `q`.

W tabeli 3.3 znajdziemy najpopularniejsze skróty klawiszowe programu `less`.

Tabela 3.3. Polecenia programu `less`

Polecenie	Akcja
PAGE UP lub b	Przewija o jedną stronę w górę.
PAGE DOWN lub klawisz spacji	Przewija o jedną stronę w dół.
Klawisz strzałki w górę	Przewija o jeden wiersz w górę.
Klawisz strzałki w dół	Przewija o jeden wiersz w dół.
G	Przenosi na koniec pliku tekstowego.
1G lub g	Przenosi na początek pliku tekstowego.
/znaki	Szuka w przód kolejnego wystąpienia słowa <i>znaki</i> .
n	Szuka kolejnego wystąpienia poprzedniego wyszukiwania.
h	Wyświetla ekran pomocy.
q	Kończy działanie programu <code>less</code> .

LESS IS MORE — MNIEJ ZNACZY WIĘCEJ

Program `less` został zaprojektowany jako ulepszony następca wcześniejszego programu uniksowego o nazwie `more`. Nazwa `less` pochodzi od powiedzenia „mniej znaczy więcej” (ang. *less is more*) — jest to motto modernistycznych architektów i projektantów.

`less` należy do klasy programów umożliwiających przeglądanie długich dokumentów tekstowych w przystępny sposób, strona po stronie (tzw. *pages*). Podczas gdy program `more` pozwalał jedynie na wyświetlanie kolejnych stron, program `less` umożliwia przeglądanie stron w przód i wstecz, a także posiada wiele innych funkcji.

Wycieczka z przewodnikiem

Układ plików w Linuksie przypomina ich układ w innych systemach uniksowych. Jest on zgodny ze standardem opublikowanym pod nazwą *Linux Filesystem Hierarchy Standard* (standard hierarchii systemu plików w Linuksie). Niektóre dystrybucje Linuksa nie przestrzegają ściśle zasad tego standardu, lecz większość dystrybucji odzwierciedla go dość dobrze.

PAMIĘTAJMY O SZTUCZCE „KOPIUJ I WKLEJ”!

Jeśli używamy myszy, kliknijmy dwukrotnie nazwę pliku, aby go skopiować, a następnie naciśnijmy środkowy przycisk myszy, aby wkleić nazwę pliku do polecenia.

Zwiedzimy teraz system plików, aby się przekonać, co znajduje się u podstaw Linuksa. Przy okazji przećwiczymy swoje umiejętności nawigacyjne. Przekonamy się, że wiele z interesujących nas plików ma format zwykłego tekstu zrozumiałego dla człowieka. W trakcie tej wycieczki warto skorzystać z następujących atrakcji:

- Przejść do podanego katalogu za pomocą polecenia `cd`.
- Wyświetlić zawartość katalogu za pomocą polecenia `ls -l`.
- Jeśli zainteresuje nas jakiś plik, określimy jego zawartość za pomocą polecenia `file`.
- Jeśli wydaje się nam, że może to być plik tekstowy, spróbujemy wyświetlić jego treść za pomocą polecenia `less`.

Jeśli przypadkiem wyświetlimy plik, który nie zawiera tekstu i jego zawartość zaśmiesi okno terminala, możemy temu zaradzić wykonując polecenie `reset`.

W trakcie wycieczki nie należy się obawiać sprawdzania różnych rzeczy. Regularni użytkownicy w dużej mierze nie mają uprawnień pozwalających na zepsucie czegośkolwiek. To zadanie administratorów systemu! Jeśli polecenie wyświetla jakiegoś komunikaty, po prostu przejdźmy do czegoś innego. Warto poświęcić nieco

czasu na rozejrzenie się. System należy do nas i tylko czeka na odkrycie. Pamiętajmy — w świecie Linuksa nie ma tajemnic!

W tabeli 3.4 wypisano niektóre katalogi, które warto przejrzeć. W zależności od dystrybucji Linuksa katalogi te mogą się nieco różnić. Nie musimy się do nich ograniczać!

Tabela 3.4. Katalogi znajdujące się w systemie Linux

Katalog	Komentarze
/	Katalog główny, w którym wszystko ma swój początek.
/bin	Zawiera pliki binarne (programy), które są niezbędne do uruchomienia i działania systemu.
/boot	Zawiera jądro Linuksa, wstępny obraz dysku RAM (ze sterownikami potrzebnymi w czasie rozruchu) oraz program rozruchowy. Do ciekawych plików należą <i>/boot/grub/grub.conf</i> lub <i>menu.lst</i> , służące do konfiguracji programu rozruchowego, a także <i>/boot/vmlinuz</i> (lub podobny) — jądro Linuksa.
/dev	Jest to katalog specjalny zawierający <i>węzły urządzeń</i> . Zasada „wszystko jest plikiem” dotyczy także urządzeń. W tym miejscu jądro utrzymuje listę wszystkich rozpoznawanych przez siebie urządzeń.
/etc	Katalog <i>/etc</i> zawiera wszystkie ogólnosystemowe pliki konfiguracyjne. Zawiera też zestaw skryptów powłoki, które uruchamiają usługi systemowe w czasie rozruchu. Wszystkie pliki znajdujące się w tym katalogu powinny mieć format czytelnego tekstu. Co prawda cała zawartość katalogu <i>/etc</i> jest ciekawa, lecz poniższe pliki są moimi ulubieńcami: <i>/etc/crontab</i> — plik definiujący czas uruchomienia zadań automatycznych; <i>/etc/fstab</i> — tablica urządzeń pamięciowych i związanych z nimi punktów montowania; <i>/etc/passwd</i> — lista kont użytkowników.
/home	W normalnej konfiguracji systemu każdy użytkownik otrzymuje własny katalog domowy w katalogu <i>/home</i> . Zwykli użytkownicy mogą zapisywać pliki tylko we własnym katalogu domowym. To ograniczenie chroni system przed niewłaściwym działaniem użytkowników.
/lib	Zawiera pliki współdzielonych bibliotek używanych przez główne programy systemu. Są one podobne do bibliotek DLL w systemie Windows.
/lost+found	Katalog ten zawiera każda sformatowana partycja lub urządzenie wykorzystujące system plików Linuksa taki jak <i>xt3</i> . Katalog ten jest wykorzystywany w przypadku częściowego odzyskiwania po uszkodzeniu systemu plików. O ile z systemem nie stanie się nic złego, ten katalog pozostanie pusty.
/media	We współczesnych systemach linuksowych katalog <i>/media</i> będzie zawierał punkty montowania dla nośników wymiennych, takich jak napędy USB, CD-ROM itd., które zostały zamontowane automatycznie w momencie podłączenia.
/mnt	W starszych systemach linuksowych katalog <i>/mnt</i> zawiera punkty montowania dla nośników wymiennych, które zostały zamontowane ręcznie.
/opt	Katalog <i>/opt</i> służy do instalowania „opcjonalnego” oprogramowania. Przechowywane są w nim głównie komercyjne produkty, które mogą być zainstalowane w systemie.
/proc	Katalog <i>/proc</i> ma specjalne znaczenie. Nie jest to zwykły system plików w tym sensie, że nie służy do przechowywania plików na dysku twardym. Jest to raczej wirtualny system plików utrzymywany przez jądro Linuksa. „Pliki”, które się w nim znajdują, odgrywają rolę wizjera do wnętrza samego jądra. Pliki te można odczytać i dzięki temu uzyskać dobry obraz obraz, jak jądro widzi nasz komputer.
/root	Jest to katalog domowy głównego użytkownika.
/sbin	Ten katalog zawiera pliki binarne „systemu”. Są to programy wykonujące niezbędne zadania systemowe, które są generalnie zarezerwowane dla użytkownika uprzywilejowanego.

Tabela 3.4. Katalogi znajdujące się w systemie Linux — ciąg dalszy

Katalog	Komentarze
<code>/tmp</code>	Katalog <code>/tmp</code> jest przeznaczony do przechowywania tymczasowych plików tworzonych przez różne programy. Niektóre konfiguracje systemu odpowiadają za opróżnianie tego katalogu podczas każdego uruchamiania systemu.
<code>/usr</code>	Drzewo katalogu <code>/usr</code> jest prawdopodobnie największe w systemie Linux. Zawiera wszystkie programy wraz z potrzebnymi im plikami wykorzystywane przez zwykłych użytkowników.
<code>/usr/bin</code>	<code>/usr/bin</code> zawiera programy wykonywalne zainstalowane w naszej dystrybucji Linuksa. W katalogu tym są często tysiące programów.
<code>/usr/lib</code>	Współdzielone biblioteki dla programów z katalogu <code>/usr/bin</code> .
<code>/usr/local</code>	Katalog <code>/usr/local</code> zawiera ogólnosystemowe programy wykonywalne, które nie wchodzi w skład naszej dystrybucji. Programy skompilowane ze źródeł są zwykle instalowane w katalogu <code>/usr/local/bin</code> . W nowo zainstalowanym systemie Linux to drzewo katalogów istnieje, lecz pozostaje puste, dopóki administrator systemu nie umieści w nim jakiegos programu.
<code>/usr/sbin</code>	Zawiera więcej programów administracyjnych.
<code>/usr/share</code>	<code>/usr/share</code> zawiera wszystkie współdzielone dane wykorzystywane przez programy znajdujące się w katalogu <code>/usr/bin</code> . Są to pliki konfiguracyjne, ikony, tła pulpitu, pliki dźwiękowe itd.
<code>/usr/share/doc</code>	Większość pakietów zainstalowanych w systemie zawiera jakiś rodzaj dokumentacji. W katalogu <code>/usr/share/doc</code> znajdziemy pliki dokumentacji zebrane w pakiety.
<code>/var</code>	Katalogi, które przeglądaliśmy dotychczas, z wyjątkiem katalogów <code>/tmp</code> i <code>/home</code> , pozostają dość statyczne; oznacza to, że ich zawartość się nie zmienia. Drzewo katalogu <code>/var</code> to miejsce przechowywania danych, które podlegają zmianom. Umieszczone są tu bazy danych, pliki kolejek, poczta użytkownika itd.
<code>/var/log</code>	Katalog <code>/var/log</code> zawiera <i>pliki logów</i> rejestrujące różną aktywność systemu. Są to bardzo ważne pliki i należy je od czasu do czasu przeglądać. Najprzydatniejszy jest <code>/var/log/messages</code> i <code>/var/log/syslog</code> . Pamiętajmy, że ze względów bezpieczeństwa, w niektórych systemach tylko użytkownik uprzywilejowany może przeglądać pliki logów.

Dowiązania symboliczne

Rozglądając się po systemie, prawdopodobnie zauważymy na liście zawartości katalogu element wyglądający jak poniżej:

```
lrwxrwxrwx 1 root root 11 2018-08-11 07:34 libc.so.6 -> libc-2.6.so
```

Zauważmy, że pierwsza litera to *l*; wydaje się, że element ma dwie nazwy. Jest to specjalny rodzaj pliku zwany *dowiązaniem symbolicznym* (inaczej *dowiązanie miękkie*, ang. *symlink*). W większości systemów uniksowych do jednego pliku może się odnosić wiele nazw. Chociaż teraz możemy tego nie doceniać, jest to naprawdę przydatna funkcja.

Wyobraźmy sobie następujący scenariusz — program wymaga pewnego rodzaju współdzielonych zasobów, zawartych w pliku o nazwie *foo*, jednak *foo* podlega częstym zmianom. Warto uwzględnić w nazwie pliku numer wersji, aby administrator lub inni zainteresowani użytkownicy mogli łatwo ustalić, która wersja *foo* jest zainstalowana. Jest to problematyczne. Jeśli zmienimy nazwę współdzielonego zasobu, musimy odszukać wszystkie programy, które z niego korzystają, i za każdym razem, gdy zainstalujemy nową wersję, zmodyfikować je tak, aby wyszukiwały zasób na podstawie nowej nazwy. Nie brzmi to zbyt dobrze.

Tutaj na ratunek przychodzą dowiązania symboliczne. Załóżmy, że instalujemy wersję 2.6 pliku *foo*, która ma nazwę *foo-2.6*, a następnie tworzymy dowiązanie symboliczne zwane po prostu *foo*, które wskazuje na plik *foo-2.6*. Oznacza to, że gdy program otworzy plik *foo*, w rzeczywistości otwiera plik *foo-2.6*. Teraz wszyscy są szczęśliwi. Programy, które wymagają pliku *foo*, mogą go znaleźć, a my nadal mamy możliwość sprawdzenia zainstalowanej wersji. Gdy nadejdzie czas uaktualnienia pliku do wersji 2.7, możemy umieścić plik w naszym systemie, usunąć dowiązanie symboliczne *foo* i utworzyć nowe, wskazujące nową wersję pliku. To nie tylko rozwiązuje problem uaktualniania wersji, lecz także pozwala nam przechowywać na komputerze obie wersje pliku. Wyobraźmy sobie, że *foo-2.7* zawiera błąd (niech szlag trafi programistów!) i chcemy powrócić do wcześniejszej wersji. Ponownie wystarczy usunąć dowiązanie symboliczne wskazujące nową wersję i utworzyć nowe dowiązanie symboliczne wskazujące na starą wersję.

Powyższy wiersz z listy katalogów (z katalogu */lib* w systemie Fedora) przedstawia dowiązanie symboliczne o nazwie *libc.so.6*, które wskazuje na plik współdzielonej biblioteki mający nazwę *libc-2.6.so*. Oznacza to, że programy wymagające pliku *libc.so.6* w rzeczywistości uzyskują dostęp do pliku *libc-2.6.so*. W kolejnym rozdziale dowiemy się, jak tworzyć dowiązania symboliczne.

Dowiązania twarde

Omawiając dowiązania, należy też wspomnieć o ich drugim rodzaju, czyli o *dowiązaniach twardych*. Dowiązania twarde również pozwalają na tworzenie wielu nazw plików, jednak w inny sposób. Więcej informacji na temat różnic między dowiązaniem symbolicznym a twardym znajduje się w następnym rozdziale.

Podsumowanie

Dobiegła końca wycieczka, podczas której dowiedzieliśmy się wiele o naszym systemie. Poznaliśmy różnego rodzaju pliki i katalogi oraz ich zawartość. Powinniśmy zapamiętać, jak bardzo otwarty jest system. W Linuksie istnieje wiele istotnych plików, które zawierają zwykły tekst w formacie czytelny dla ludzi. W przeciwieństwie do wielu systemów własnościowych w systemie Linux wszystko można przejrzeć i przeanalizować.

PROGRAM PARTNERSKI

— GRUPY HELION —



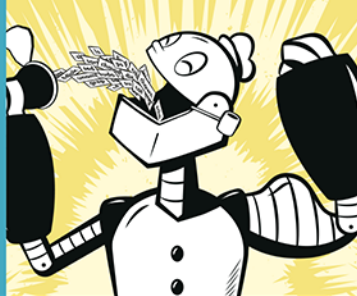
1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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

GRUPA
Helion 



ODZYSKAJ KONTROLĘ NAD SWOIM KOMPUTEREM — ZAPOMNIJ O MYSZY!

Komputery są wszędzie, a ich użyteczność stale rośnie. Łącząca je sieć ogromnie zwiększa ich możliwości. Dziś niemal każdy ma dostęp do komputera. Nadeszła era niezwykłego równouprawnienia i kreatywnej wolności. Jednak od jakiegoś czasu widoczny jest odwrotny trend: kilka wielkich korporacji coraz więcej z niej przejmuje i chce dyktować, co wolno, a czego nie wolno wykonać na komputerze. Oprogramowanie open source, zwłaszcza system Linuks, można traktować jako obronę szczególnego rodzaju wolności: możliwości decydowania o tym, co dokładnie robi komputer, bez jakichkolwiek tajemnic i ukrytych wątków. Ta wolność wymaga głębokiej wiedzy i pewnych umiejętności. Ta wolność wymaga posługiwania się wierszem poleceń.

Książka jest kolejnym wydaniem bestsellerowego podręcznika, dzięki któremu już za chwilę możesz zacząć wpisywać pierwsze polecenia w terminalu, aby niespostrzeżenie przejść do tworzenia złożonych skryptów w najnowszej wersji basha 4.x, najpopularniejszej powłoki Linuksa. Przedstawiono tu nowe operatory przekierowania i mechanizmy interpretacji w powłoce. Uaktualniono część dotyczącą skryptów oraz sposoby unikania typowych, potencjalnie groźnych błędów. Dzięki książce zdobędziesz cenne umiejętności, takie jak nawigacja w systemie plików, konfigurowanie

środowiska, łączenie poleceń oraz dopasowywanie wzorców za pomocą wyrażeń regularnych. Przekonasz się, że bez myszy możesz kontrolować swój komputer na nieosiągalnym dotychczas poziomie!

Najciekawsze zagadnienia:

- tworzenie i usuwanie plików, katalogów i dowiązań symbolicznych
- administrowanie systemem, w tym zarządzanie procesami
- stosowanie standardowego wejścia i wyjścia, przekierowań oraz potoków
- praca z vi

William Shotts programuje od ponad 30 lat. Od ponad 20 lat jest entuzjastycznym użytkownikiem systemu Linux. Ma rozległą wiedzę o tworzeniu oprogramowania, zapewnianiu jakości i wsparcia technicznego, a także tworzeniu dokumentacji. Założył portal LinuxCommand.org, na którym znajdują się informacje o nowościach, recenzje oraz sekcje poświęcone wsparciu dla użytkowników wiersza poleceń systemu Linux.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 93 63
helion@helion.pl

Sprawdź nasze szkolenia!

SZKOLENIA



AKADEMIA IT & BUSINESS

HELIONSZKOLENIA.PL

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-283-6762-3



9 788328 367623

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 99,00 zł

