WYDANIE II

LINUX

WPROWADZENIE DO WIERSZA POLECEŃ

WILLIAM SHOTTS

Helion

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

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.

Spis treści

Część II
Konfiguracja i środowisko

Część IV
Tworzenie skryptów powłoki

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 `.` (bieżący katalog), ani `..` (katalog nadrzędny).
`-d`	`--directory`	Zwykle jeśli podajemy nazwę katalogu, `ls` wyświetli zawartość katalogu, pomijając informacje o samym katalogu. Gdy użyjemy tej opcji razem z opcją `-l`, wyświetlone zostaną szczegóły samego katalogu, zamiast jego zawartości.
`-F`	`--classify`	Ta opcja dodaje na końcu każdej wyświetlonej nazwy znak wskaźnika. Przykładowo doda ukośnik (/), 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 `ls` 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
```

Przeanalizujmy poszczególne pola jednego z plików i sprawdźmy 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 (zwanych *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 zwykłe 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 *znaki*.
n	Szuka kolejnego wystąpienia poprzedniego wyszukiwania.
h	Wyświetla ekran pomocy.
q	Kończy działanie programu `less`.

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. *pagers*). 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, określmy jego zawartość za pomocą polecenia `file`.
- Jeśli wydaje się nam, że może to być plik tekstowy, spróbujmy wyświetlić jego treść za pomocą polecenia `less`.

Jeśli przypadkiem wyświetlimy plik, który nie zawiera tekstu i jego zawartość zaśmieci 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 czegokolwiek. To zadanie administratorów systemu! Jeśli polecenie wyświetla jakieś 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 */boot/grub/grub.conf* lub *menu.lst*, służące do konfiguracji programu rozruchowego, a także */boot/vmlinuz* (lub podobny) — jądro Linuksa.
/dev	Jest to katalog specjalny zawierający *węzły urządzeń*. Zasada „wszystko jest plikiem" dotyczy także urządzeń. W tym miejscu jądro utrzymuje listę wszystkich rozpoznawanych przez siebie urządzeń.
/etc	Katalog */etc* 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 */etc* jest ciekawa, lecz poniższe pliki są moimi ulubieńcami: */etc/crontab* — plik definiujący czas uruchomienia zadań automatycznych; */etc/fstab* — tablica urządzeń pamięciowych i związanych z nimi punktów montowania; */etc/passwd* — lista kont użytkowników.
/home	W normalnej konfiguracji systemu każdy użytkownik otrzymuje własny katalog domowy w katalogu */home*. 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 *xt3*. 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 */media* 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 */mnt* zawiera punkty montowania dla nośników wymiennych, które zostały zamontowane ręcznie.
/opt	Katalog */opt* służy do instalowania „opcjonalnego" oprogramowania. Przechowywane są w nim głównie komercyjne produkty, które mogą być zainstalowane w systemie.
/proc	Katalog */proc* 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 tego, 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
/tmp	Katalog */tmp* jest przeznaczony do przechowywania tymczasowych plików tworzonych przez różne programy. Niektóre konfiguracje systemu odpowiadają za opróżnianie tego katalogu podczas każdego uruchamiania systemu.
/usr	Drzewo katalogu */usr* jest prawdopodobnie największe w systemie Linux. Zawiera wszystkie programy wraz z potrzebnymi im plikami wykorzystywane przez zwykłych użytkowników.
/usr/bin	*/usr/bin* zawiera programy wykonywalne zainstalowane w naszej dystrybucji Linuksa. W katalogu tym są często tysiące programów.
/usr/lib	Współdzielone biblioteki dla programów z katalogu */usr/bin*.
/usr/local	Katalog */usr/local* zawiera ogólnosystemowe programy wykonywalne, które nie wchodzą w skład naszej dystrybucji. Programy skompilowane ze źródeł są zwykle instalowane w katalogu */usr/local/bin*. W nowo zainstalowanym systemie Linux to drzewo katalogów istnieje, lecz pozostaje puste, dopóki administrator systemu nie umieści w nim jakiegoś programu.
/usr/sbin	Zawiera więcej programów administracyjnych.
/usr/share	*/usr/share* zawiera wszystkie współdzielone dane wykorzystywane przez programy znajdujące się w katalogu */usr/bin*. Są to pliki konfiguracyjne, ikony, tła pulpitu, pliki dźwiękowe itd.
/usr/share/doc	Większość pakietów zainstalowanych w systemie zawiera jakiś rodzaj dokumentacji. W katalogu */usr/share/doc* znajdziemy pliki dokumentacji zebrane w pakiety.
/var	Katalogi, które przeglądaliśmy dotychczas, z wyjątkiem katalogów */tmp* i */home*, pozostają dość statyczne; oznacza to, że ich zawartość się nie zmienia. Drzewo katalogu */var* to miejsce przechowywania danych, które podlegają zmianom. Umieszczane są tu bazy danych, pliki kolejek, poczta użytkownika itd.
/var/log	Katalog */var/log* zawiera *pliki logów* rejestrujące różną aktywność systemu. Są to bardzo ważne pliki i należy je od czasu do czasu przeglądać. Najprzydatniejszy jest */var/log/messages* i */var/log/syslog*. 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 uzyskają dostęp do pliku *libc-2.6.so*. W kolejnym rozdziale dowiemy się, jak tworzyć dowiązania symboliczne.

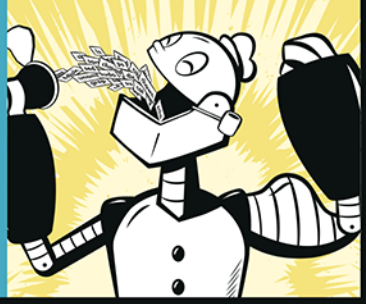
Dowiązania twarde

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 dowiązaniami symbolicznymi i twardymi 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 czytelnym 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 powłoki — zaprezentowano solidne praktyki pisania 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.

ISBN 978-83-283-6762-3