helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Linux. Komendy i polecenia



Autor: Łukasz Sosna
ISBN: 83-7361-480-X
Format: B6, stron: 78

Pierwszy kontakt z Linuksem może być dla użytkownika Windows lekkim szokiem – bardziej skomplikowana procedura instalacyjna, zupełnie inne zasady korzystania z wielu funkcji i mnogość możliwości konfiguracji to najczęstsze źródła problemów. Jednak największe kłopoty sprawia praca z poleceniami i komendami.

Jeśli chcesz mieć zawsze pod ręką przydatną ściągę, sięgnij po książkę „Linux. Komendy i polecenia".

Znajdziesz w niej informacje o zastosowaniu i składni poleceń systemowych. Nauczysz się korzystać z konsoli tekstowej, poznasz polecenia pozwalające na zarządzanie systemem plików, administrowanie systemem i zarządzanie kontami użytkowników i dowiesz się, jakich parametrów wymagają poszczególne komendy. Szybko znajdziesz wszystkie potrzebne Ci wiadomości.

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

Spis treści

Wprowadzenie do systemu Linux 5

Czym jest Linux 5
Dostępne dystrybucje; jak wybrać odpowiednią dla siebie 6
Instalacja systemu 7
    Instalacja przy użyciu płyty CD 8
    Instalacja przy użyciu dyskietki 8
    Uwagi do procesu instalacji 9

Korzystanie z komputera pracującego pod kontrolą systemu Linux 11

Środowisko pracy 11
Logowanie się do systemu 12
    Logowanie się w trybie tekstowym 13
    Logowanie się w trybie graficznym 14
Bezpieczne wyłączanie i restart komputera 15
Użytkownicy systemu Linux 16
Co znajduje się w poszczególnych katalogach systemu 17
    Gdzie jest miejsce na twoje pliki 18
Dyski i partycje w systemie 19
Pomoc na stronach MAN 20

Zarządzanie zasobami komputera 20

Pliki i katalogi w systemie 20
Wyświetlanie zawartości katalogu 22
    Polecenie dir 22
    Polecenie vdir 23
    Polecenie ls 24
Przechodzenie pomiędzy katalogami 32
Tworzenie katalogów 34
Usuwanie katalogów 35
Tworzenie plików 36

Usuwanie plików ... 37
Wyświetlenie zawartości pliku ... 38
Zmiana dat modyfikacji plików i dostępu do nich ... 39
Kopiowanie plików i katalogów ... 42
Przenoszenie plików i katalogów oraz zmiana ich nazwy ... 45
Nadawanie praw dostępu do plików i katalogów ... 47
Tworzenie dowiązań symbolicznych ... 50
Tworzenie aliasów ... 51
Zmiana hasła ... 52
Zmiana powłoki ... 53
Uzyskiwanie informacji o typie pliku ... 53
Zmiana właściciela i grupy pliku ... 54
Wyszukiwanie plików i katalogów ... 56
Ustalanie zajętego i wolnego miejsca na partycjach ... 62
Ustalanie, ile miejsca zajmuje plik lub katalog ... 63
Polecenia more i less ... 64
Czyszczenie terminala ... 66
Montowanie i odmontowywanie systemów plików ... 66
Przełączanie się na konto innego użytkownika ... 68
Uzyskiwanie informacji o sprzęcie ... 69
Polecenie arch ... 69
Polecenie uname ... 69
Informacje o użytkowniku ... 71
Kto jest obecnie zalogowany ... 72
Informacje o pamięci systemowej ... 72

Administrowanie systemem ... 73

Poziom uruchomienia systemu ... 73
Demony usług ... 74
Uruchamianie i zatrzymywanie ... 74
Ustawianie demonów do startu w odpowiednim trybie ... 75
Użytkownicy ... 76
Grupy ... 78

Skorowidz ... 79

Zarządzanie zasobami komputera

Pierwszą rzeczą, którą powinieneś poznać, jest zarządzanie zasobami twojego komputera; musisz nauczyć się tworzyć i usuwać elementy na dysku, zmieniać ich atrybuty oraz wykonywać wiele innych czynności, które pomogą ci w poruszaniu się po strukturze katalogów i pozwolą w łatwy sposób dotrzeć do interesujących cię plików.

Pliki i katalogi w systemie

W Linuksie, inaczej niż w takich systemach operacyjnych jak choćby systemy z rodziny Microsoft Windows, nie jest wymagane stosowanie w nazwach plików specjalnych rozszerzeń, określających m.in. to, jaki program powinien zostać użyty do otwarcia pliku. Zawartość pliku i program, jaki należy zastosować do jego otwarcia, jest w Linuksie ustalana na podstawie nagłówka `MIME` pliku. Mimo to, jeżeli z jakiegokolwiek powodu odczuwasz potrzebę nadawania plikom rozszerzeń, możesz to robić — choć, jak wspomniałem, rozszerzenia te nie są konieczne, ich stosowanie nie jest także niewskazane.

W nazwach plików i katalogów możesz stosować dowolne znaki alfanumeryczne (litery i cyfry), a oprócz tego znak kropki (`.`), myślnika (`-`) i podkreślenia (`_`). Z wszystkimi innymi znakami postępuj ostrożnie; zazwyczaj są one zarezerwowane dla specjalnych funkcji systemu.

W nazwach plików i katalogów możesz używać także spacji i nie będzie to powodować większych problemów, będzie jednak po

prostu niewygodne. Komendy, w których będziesz odwoływać się do plików lub katalogów zawierających spacje będą po prostu dłuższe i łatwiej będzie popełnić błąd podczas wpisywania nazwy pliku.

Trzeba także pamiętać o tym, że znak kropki nie powinien być znakiem rozpoczynającym nazwę pliku czy katalogu. Napisałem wprawdzie nieco wcześniej, iż jego stosowanie jest dozwolone, trzeba jednak pamiętać o tym niezwykle ważnym zastrzeżeniu; nazwy plików i katalogów nie powinny *rozpoczynać się* od znaku kropki, w każdym innym jednak miejscu może on wystąpić. Przyczyną tego ograniczenia jest to, że w Linuksie znak kropki na początku nazwy pliku jest zarezerwowany dla ukrytych plików i katalogów — na przykład plik o nazwie *.ukryty_plik* będzie plikiem ukrytym.

Niezwykle istotna jest także wielkość stosowanych przez nas liter. W omawianym systemie wielkie i małe litery są rozpoznawane jako osobne znaki. Jeśli więc tworzysz katalog lub plik, zapamiętaj, czy jego nazwę wpisałeś wielką, czy małą literą — będzie ci to potrzebne, kiedy będziesz chciał się do niego w przyszłości odwołać.

Wyświetlany przez system znak `$` (jeśli korzystasz z konta zwykłego użytkownika) `# (jeżeli pracujesz jako administrator systemu)` jest znakiem zachęty. Oto przykład jego użycia:

```
[lukasz@linux /]$
```

Przed znakiem zachęty występują nazwa użytkownika i nazwa hosta oraz ciąg znaków określający bieżącą lokalizację w systemie plików. W tym przypadku:

- użytkownikiem jest *lukasz*,
- host, na którym pracujemy, to *linux*,
- katalog, w którym się znajdujemy, to / (katalog główny).

Wyświetlanie zawartości katalogu

Wyświetlanie katalogów, zwane inaczej *listowaniem ich zawartości*, można w systemie Linux wykonać za pomocą kilku poleceń. System posiada bardzo rozbudowany program służący do wykonywania tego typu zadań — jest nim polecenie `ls`.

Polecenie dir

Polecenie to może przypominać jedno z poleceń systemu operacyjnego MS-DOS i wszyscy użytkownicy, którzy znają to środowisko, zapewne poczuli się jak w domu. Jednak wynik działania tego polecenia w Linuksie różni się od tego z systemu DOS. W Linuksie wyświetlana po wykonaniu tego polecenia lista plików i katalogów jest prezentowana w postaci linii, a nie w kolumnie. Trzeba przyznać, że taki zapis utrudnia nieco odczytanie drzewa katalogów i orientację w nim, szczególnie w przypadku, gdy użytkownik przyzwyczajony jest do prezentowania go w postaci znanej z MS-DOS. Oto przykład wykonania polecenia `dir` w systemie Linux:

```
[lukasz@linux /]$ dir
bin  boot  dev  etc  home  initrd  lib  lost+found
misc  mnt  opt  proc  root  sbin  tmp  usr  var
```

Polecenie vdir

Wykonanie polecenia `vdir` powoduje wyświetlenie bardziej szczegółowych informacji o zawartości bieżącego katalogu; oprócz nazw plików i katalogów podawane są informacje o typie elementu, prawach dostępu do niego, jego właścicielu oraz kilka innych, które omówię na przykładzie polecenia `ls`. Oto przykład wykonania polecenia `vdir`:

```
[lukasz@linux /]$ vdir
```

```
drwxr-xr-x    2 root     root         4096 lis 28
17:47 bin
drwxr-xr-x    3 root     root         4096 lip  4
2003 boot
drwxr-xr-x   20 root     root       118784 lut 14
17:03 dev
drwxr-xr-x   62 root     root         4096 lut 14
17:03 etc
drwxr-xr-x    5 root     root         1024 wrz 13
21:07 home
drwxr-xr-x    2 root     root         4096 sty 25
2003 initrd
drwxr-xr-x    9 root     root         4096 lis 28
18:06 lib
drwx------    2 root     root        16384 lip  4
2003 lost+found
drwxr-xr-x    2 root     root         4096 sty 28
2003 misc
drwxr-xr-x    4 root     root         4096 lip  4
2003 mnt
drwxr-xr-x    2 root     root         4096 sty 25
2003 opt
dr-xr-xr-x   76 root     root            0 lut 14
2004 proc
drwxr-x---   22 root     root         4096 sty  5
21:01 root
drwxr-xr-x    2 root     root         8192 lip  4
2003 sbin
drwxrwxrwt   21 root     root         4096 lut 14
17:05 tmp
drwxr-xr-x   15 root     root         4096 lip  4
2003 usr
drwxr-xr-x   19 root     root         4096 lip  4
2003 var
```

Polecenie ls

Program ten jest bardzo rozbudowany i posiada wiele parametrów, które pokrótce postaram się omówić. Wykonanie samego polecenia `ls` (bez dodatkowych parametrów) da efekt identyczny

z tym, jaki powodowało wykonanie omówionego poprzednio polecenia `dir`. Aby uzyskać więcej informacji na temat zawartości katalogu, powinniśmy zastosować polecenie `ls` wraz z parametrem `-l`. Wykonanie tego polecenia w takiej postaci daje wynik identyczny z wynikiem działania polecenia `vdir`.

Jako parametr można podać ścieżkę dostępu dla katalogu, którego zawartość chcemy wyświetlić. Jeżeli jej nie podamy, zawsze wyświetlona zostanie zawartość katalogu bieżącego — tego, w którym się obecnie znajdujemy.

W zaprezentowanym tu przykładzie polecenie `ls` wywołane z parametrem `/etc` spowoduje wyświetlenie zawartości katalogu *etc*; użyte w drugiej linii przykładu polecenie `ls` wywołane bez parametru wyświetli natomiast zawartość katalogu bieżącego, którym jest w tym przypadku katalog główny.

```
[lukasz@linux /]$ ls /etc

[lukasz@linux /]$ ls
```

Najbardziej przydatnym parametrem polecenia `ls` jest `-l`; dzięki jego wykonaniu otrzymamy kompletny zestaw informacji na temat zawartości katalogu i typach zawartych w nim elementów. Zanim przedstawię resztę najważniejszych parametrów tego polecenia, wyjaśnię, jak interpretować zdobyte w ten sposób informacje. Oto przykład wykonania polecenia `ls` z parametrem `-l`:

```
[lukasz@linux /]$ ls -l
drwxr-xr-x    2 root     root         4096 lis 28
17:47 bin
drwxr-xr-x    3 root     root         4096 lip  4
2003 boot
drwxr-xr-x   20 root     root       118784 lut 14
17:03 dev
drwxr-xr-x   62 root     root         4096 lut 14
17:03 etc
drwxr-xr-x    5 root     root         1024 wrz 13
21:07 home
```

```
drwxr-xr-x    2 root     root         4096 sty 25
2003 initrd
drwxr-xr-x    9 root     root         4096 lis 28
18:06 lib
drwx------    2 root     root        16384 lip  4
2003 lost+found
drwxr-xr-x    2 root     root         4096 sty 28
2003 misc
drwxr-xr-x    4 root     root         4096 lip  4
2003 mnt
drwxr-xr-x    2 root     root         4096 sty 25
2003 opt
dr-xr-xr-x   76 root     root            0 lut 14
2004 proc
drwxr-x---   22 root     root         4096 sty  5
21:01 root
drwxr-xr-x    2 root     root         8192 lip  4
2003 sbin
drwxrwxrwt   21 root     root         4096 lut 14
17:05 tmp
drwxr-xr-x   15 root     root         4096 lip  4
2003 usr
drwxr-xr-x   19 root     root         4096 lip  4
2003 var
```

Wyświetlone w tym przykładzie informacje, zdobyte wskutek wykonania polecenia z parametrem `-l`, zinterpretujemy na podstawie pierwszej linii wyniku:

```
drwxr-xr-x    2 root     root         4096 lis 28
17:47 bin
```

- kolumna 1: typ elementu i prawa dostępu do niego (`drwxr-xr-x`);
- kolumna 2: liczba powiązań do tego elementu (`2`);
- kolumna 3: właściciel pliku (`root`);
- kolumna 4: grupa, która została przypisana do tego pliku (`root`);

- kolumna 5: rozmiar elementu (`4096`);
- kolumna 6: data modyfikacji (`lis 28 17:47`);
- kolumna 7: nazwa elementu (`bin`).

Rozpoznanie typu elementu

Zapis `drwxr-xr-x` z kolumny pierwszej składa się z czterech zasadniczych elementów. Pierwsza litera zawsze określa typ elementu.

Oto symbole oznaczające typy elementów:

- `-` — zwykły plik;
- `b` — specjalny plik blokowy;
- `c` — specjalny plik znakowy;
- `d` — katalog;
- `l` — dowiązanie symboliczne;
- `p` — nazwany potok;
- `s` — gniazdo.

A zatem, jak można wnioskować po zapisie `drwxr-xr-x`, rozpatrywany obiekt jest katalogiem.

Interpretacja praw dostępu

Niech przykładem, przy pomocy którego wyjaśnię, na czym polega system praw dostępu w systemie Linux, będzie ten wiersz przykładowego wyniku wykonania polecenia `ls`, który dotyczy katalogu *var*:

```
drwxr-xr-x   19 root     root         4096 lip  4
2003 var
```

Prawa dostępu określane są tu poprzez litery `r`, `w` i `x`, następujące po definiującej typ elementu literze `d` (katalog). Każda litera na odpowiedniej pozycji informuje o tym, kto i jakie prawa posiada dla tego pliku lub katalogu.

Zwróć uwagę na to, że w naszym przykładzie litery `x` oraz `r` występują trzykrotnie. Taki zapis określa uprawnienia według schematu: „użytkownik-grupa-inni". Litery oznaczające uprawnienia mają różne znaczenia w zależności od tego, czy stosują się do plików, czy do katalogów.

W przypadku katalogów oznaczają następujące prawa:

- `r` — do przeszukania zawartości;
- `w` — do zmiany zawartości;
- `x` — do wejścia do katalogu.

Jakie zatem uprawnienia przypisane są do katalogu *var* z naszego przykładu? Określa je następujący zapis:

```
rwx r-x r-x
```

Oznacza to, że właściciel katalogu ma prawo na jego przeszukanie, zmianę jego zawartości i wejście do katalogu, zgodnie z zapisem `rwx`.

Grupa, która została przypisana do tego elementu, ma prawa do wejścia do katalogu i przeszukania go, zgodnie z zapisem `r-x`.

Także wszyscy inni użytkownicy mają prawa do wejścia do katalogu i przeszukania go, zgodnie z zapisem `r-x`.

Jak już wspomniałem, w przypadku plików prawa dostępu określane są przez te same symbole, jednak różna jest ich interpretacja. Tym razem litery `r`, `w` i `x` oznaczają następujące prawa:

- `r` — do odczytania pliku;
- `w` — do modyfikacji pliku;
- `x` — do uruchomienia pliku.

Rozważmy przykład z następującymi prawami dostępu do pliku:

```
rw- rw- r-
```

Ten zapis informuje o tym, iż właściciel pliku ma prawo do jego odczytywania oraz do zmiany jego zawartości, zgodnie z zapisem `rw-`.

Także grupa, która została przypisana do pliku, ma prawo do jego odczytywania i zmiany jego zawartości, zgodnie z zapisem `rw-`.

Wszyscy inni użytkownicy mają prawa jedynie do odczytania zawartości pliku, zgodnie z zapisem `r--`.

Przejdźmy teraz do omówienia kolejnych parametrów, których można użyć z poleceniem `ls`.

`-a`

Polecenie `ls` wykonane z parametrem `-a` wyświetli wszystkie pliki i katalogi w danej lokalizacji. Pokazane zostaną także pliki ukryte, które w przypadku wywołania `ls` bez tego parametru nie są widoczne. Oto przykład wykonania ls z parametrem `-a`; widzimy w nim także dwa symbole „`.`” (kropka) i „`..`” (dwie kropki), które oznaczają odpowiednio katalog bieżący i nadrzędny:

```
[lukasz@linux linux]$ ls -a
.  ..  katalog  plik1  plik2  plik_kopii~
.ukryty_plik
```

`-A`

Parametr ten pozwoli zobaczyć wszystkie elementy w podanej lokalizacji, wraz z plikami ukrytymi, jednak w tym

przypadku w wyniku nie będą widoczne symbole „.” i „..”, które oznaczają katalog bieżący i nadrzędny:

```
[lukasz@linux linux]$ ls -A
katalog  plik1  plik2  plik_kopii~
.ukryty_plik
```

`-B`

Użycie parametru `-B` spowoduje ukrycie plików kopii zapasowych, które znajdują się w danym katalogu. Pliki te można rozpoznać po znaku tyldy (~) występującym na końcu nazwy. Jak widać, plik *plik_kopii~* nie został tutaj pokazany:

```
[lukasz@linux linux]$ ls -B
katalog  plik1  plik2
```

`-d`

Jeżeli w danym katalogu zawarte są pliki i katalogi podrzędne, polecenie `ls` wykonane bez parametru `-d` spowoduje wyświetlenie ich wszystkich. Jeżeli jednak zastosujemy parametr `-d`, zostaną wypisane tylko elementy rozpoznane jako katalogi:

```
[lukasz@linux linux]$ ls -d
katalog
```

`-I wzorzec, --ignore=wzorzec`

Dzięki temu parametrowi możemy nie pokazywać plików, których nazwy zawierają zdefiniowany przez nas wzorzec. Podobnie jak w systemie MS-DOS, także i tu możemy używać znaków ogólnych, takich jak gwiazdka (`*`), która zastępuje dowolny ciąg znaków (również pusty), oraz znak zapytania (`?`), który zastępuje dowolny pojedynczy znak.

W zaprezentowanym tu przykładzie wyświetlona została zawartość katalogu, z wyłączeniem tych elementów, których nazwy zaczynają się na literę „p”; pominięte zostały zatem *plik1*, *plik2* i *plik_kopii~*:

```
[lukasz@linux linux]$ ls --ignore='p*'
katalog
```

`wzorzec`

Dzięki podaniu wzorca nazwy elementu można wyświetlić tylko te znajdujące się w danym katalogu elementy, które pasują do tego wzorca. Wzorce tworzy się w taki sam sposób jak w systemie MS-DOS.

W tym przypadku chcemy, aby wyświetlane były elementy, których nazwa rozpoczyna się od litery „p":

```
[lukasz@linux linux]$ ls p*
plik1  plik2  plik_kopii~
```

W kolejnym przykładzie wykorzystałem także znaki zapytania, aby lepiej zobrazować ich działanie. Taki zapis pozwala na wyświetlenie wszystkich elementów, których nazwa zaczyna się od litery „p" i składa się z pięciu znaków:

```
[lukasz@linux linux]$ ls p????
plik1  plik2
```

`-R, --recursive`

Parametry te powodują rekurencyjne wyświetlenie zawartości katalogu i jego podkatalogów. W bieżącym katalogu z naszego przykładu mamy także podkatalog o nazwie *katalog*. Jak widać, po wydaniu polecenia `ls` z parametrem `-R` wyświetlona została także jego zawartość:

```
[lukasz@linux linux]$ ls -R
.:
katalog  plik1  plik2  plik_kopii~

./katalog:
plik1_w_katalogu
```

`-r, --reverse`

Użycie tych parametrów powoduje odwrócenie kolejności wyświetlania w wyniku zawartości katalogu. Najbardziej

opcja ta przydaje się w przypadku sortowania, podczas którego możemy odwrócić domyślną kolejność wypisywanych elementów.

Dla większej jasności porównajmy zawartość przykładowego katalogu w porządku oryginalnym (bez opcji `-r`) i odwróconym (z użyciem tej opcji):

```
[lukasz@linux linux]$ ls
katalog  plik1  plik2  plik_kopii~

[lukasz@linux linux]$ ls -r
plik_kopii~  plik2  plik1  katalog
```

`-S, --sort=size`

Użycie tych parametrów powoduje posortowanie wyniku według wielkości plików. Największe pliki są wyświetlane jako pierwsze na liście, zaraz za katalogami:

```
[lukasz@linux linux]$ ls -S
katalog  duzy_plik  plik1  plik1.1  plik_kopii~
```

`-t, --sort=time`

Ten parametr sortuje wyniki według czasu ich modyfikacji. Najnowsze pliki są wyświetlane jako pierwsze:

```
[lukasz@linux linux]$ ls -t
duzy_plik  katalog  plik_kopii~  plik2  plik1
```

`-u, --time=atime, --time=access, --time=use`

W tym przypadku wyniki sortowane są według czasu ostatniego dostępu do pliku. Pliki, do których odwołano się ostatnio, są wyświetlane jako pierwsze.

```
[lukasz@linux linux]$ ls -u
duzy_plik  plik1  plik2  katalog  plik_kopii~
```

`-U, --sort=none`

Użycie tego parametru zapewni, że zawartość katalogu nie będzie sortowana. Elementy wyświetlone na liście wystą-

pią na niej dokładnie w takiej kolejności, w jakiej znajdują się w katalogu.

```
[lukasz@linux linux]$ ls -U
plik1  plik2  katalog  duzy_plik  plik_kopii~
```

`-X, --sort=extension`

Użycie tego parametru powoduje posortowanie plików według ich rozszerzeń. Pliki, które takich nie posiadają, zawsze są wypisywane jako pierwsze.

```
[lukasz@linux linux]$ ls -x
duzy_plik  katalog  plik1  plik2  plik.aaa
plik.bbb
```