

Pliki w systemie operacyjnym Linux

Jeżeli system uruchamia się w trybie tekstowym, to po jego załadowaniu na czarno-białym ekranie monitora pojawiają się zapytania o nazwę i hasło użytkownika. W nowo zainstalowanym systemie przedstawiamy się zwykle jako użytkownik uprzywilejowany o nazwie *root* (podczas instalacji nadaliśmy mu hasło *123*). Końcowe komunikaty mają taką oto postać:

```
Knoppix login: root
Password
Last login: .....
Welcome to Linux-Edu CD (Kernel 2.4.24-xfs)
root@Knoppix:~#
```

Symbol gotowości systemu do przyjmowania poleceń

W oknach terminala komunikacja z systemem odbywa się w trybie poleceń. System informuje o gotowości do interakcji z użytkownikiem wyświetlając na początku kolejnego wiersza ekranu *symbol gotowości do przyjmowania poleceń*, nazywany często *symbolem zachęty* lub *monitu* (ang. *prompt*). Jest to symbol, który może być skonfigurowany odrębnie dla każdego użytkownika.

Bezpośrednio po otwarciu sesji pracy przez użytkownika uprzywilejowanego jego standardowy symbol gotowości ma postać:

```
root@Knoppix:~#
```

Bezpośrednio po otwarciu sesji pracy przez użytkownika - dajmy na to – *piotr*, symbol ten ma postać:

```
piotr@Knoppix:~$
```

Składnia poleceń

Ogólna postać poleceń jest następująca:

```
nazwa [opcje] [ argumenty]
np.
# dir -l /bin
```

Opcje

Opcje poleceń mają z reguły postać pojedynczych liter poprzedzonych znakiem - (nazywanym - w zależności od kontekstu - minusem lub myślnikiem).

Standardowe wejście-wyjście poleceń

Standardowe wejście-wyjście wywoływanego polecenia może być zmienione za pomocą pewnych symboli, z których najczęściej stosowane to:

- > plik - skierowanie standardowego wyjścia do wyspecyfikowanego *pliku*

- >> plik - skierowanie standardowego wyjścia na koniec wyspecyfikowanego *pliku* (dopisywanie na jego końcu)
- < plik - związanie standardowego wejścia z wyspecyfikowanym *plikiem* (umożliwia wprowadzenie danych z pliku zamiast z klawiatury).

System plików Linuksa

Plik (ang. *File*) to pewna porcja danych, przechowywana w pamięci dyskowej przez system jako ciąg bajtów, gdzie bajt to 8 bitów, a bit to najmniejsza (zero-jedynkowa) jednostka informacji. Zorganizowany zgodnie z wymaganiami systemu zestaw plików przechowywanych w wydzielonym obszarze pamięci dyskowej nazywany jest *systemem plików*.

System plików Linuksa - podobnie jak w systemach Unix i DOS (Window's) - jest zorganizowany hierarchicznie, co powoduje, że jest zrozumiały dla użytkownika z uwagi na postać hierarchicznego drzewa, gdzie w każdym węźle tego drzewa mogą być opisane zarówno pliki danych, jak i katalogi niższych poziomów grupujące informacje o pewnych zestawach plików. W Linuksie i Uniksie - w odróżnieniu od systemu DOS (Windows) - istnieje tylko jeden katalog główny o jednoznakowej nazwie / (*slash*). Do jego wybranych podkatalogów można podłączać kolejne partycje tego samego lub innych dysków. Nie są one jednak widoczne jako kolejne urządzenia, lecz jako zawartość katalogów, w których je dołączono. Operacja *dołączania* systemu plików (popularnie *montowania*) jest wykonywana za pomocą polecenia *mount*. Wszystkie dostępne w danej chwili zasoby dyskowe tworzą, więc jedno drzewo plików. Tego typu mechanizmy Microsoft wprowadził dopiero w Windows 2000 (w systemie plików NTFS).

Termin *system plików* obejmuje również urządzenia zewnętrzne, reprezentowane w systemie przez *pliki specjalne*. Jest on uniwersalny, ponieważ nie zakłada niczego o wewnętrznej strukturze danych w pliku i posiada identyczne mechanizmy dostępu tak do plików dyskowych, jak i do urządzeń zewnętrznych reprezentowanych przez wymienione wyżej pliki specjalne. Ponadto zapewnia ochronę przed nieuprawnionym dostępem, pozwalając określić różne kategorie współużytkowników plików oraz różne tryby dostępu do danych.

Istnieją trzy rodzaje plików, do których dostęp jest z punktu widzenia użytkownika organizowany w analogiczny sposób:

- zwykłe pliki dyskowe
- katalogi,
- pliki specjalne.

Pliki specjalne mogą dotyczyć między innymi:

- urządzeń zewnętrznych,
- pamięci operacyjnej,
- kolejek FIFO służących do organizacji nazwanych łączy komunikacyjnych, używanych do przesyłania informacji pomiędzy procesami,
- gniazd komunikacyjnych (ang. *sockets*) służących do wymiany komunikatów pomiędzy procesami pracującymi w środowisku sieciowym

Nazwy plików

Pliki zwykłe, katalogi i pliki specjalne mogą mieć długie nazwy - do 256 znaków.

W nazwie pliku zwykłego i katalogu mogą wystąpić dowolne znaki ASCII. Wielkie i małe litery są rozróżniane. Ponieważ część znaków ma dla shella specjalne znaczenie, zaleca się używać w nazwach tylko znaków liter, cyfr, podkreślenia i kropki. Znaki o specjalnym znaczeniu dla shella (zwane *metaznakami*). * ? ; | & \ < > [] ' o ile miałyby być użyte w nazwach plików, w poleceniach odwołujących się do takich nazw muszą być "cytowane" z użyciem: znaku *backslash* (\), apostrofów (') lub cudzysłówów (,,). Jeżeli nazwa pliku zaczyna się znakiem kropki, to dla pewnej grupy poleceń plik taki jest traktowany jako *plik ukryty*. Np. polecenie `ls` normalnie pomija tego typu pliki (o ile nie użyto opcji `-a`).

Rozszerzenia nazw plików

Nazwy plików w Linuksie są po prostu ciągami znaków: system nie klasyfikuje nazw ze względu na ich postać. Natomiast w niektórych innych systemach operacyjnych pliki zawierające tekst, dane lub programy rozróżnia się dodaniem kilkuliterowego przyrostka (rozszerzenia) do bazowej nazwy pliku. Takie systemy, jak DOS i Windows, dla różnych klas plików wykonywalnych stosują rozszerzenia *.EXE*, *.COM* i *.BAT*. Jednakże to, że Linux nie rozpoznaje żadnych rozszerzeń nazw plików, nie oznacza, że nie rozpoznają ich inne dostępne w systemie programy lub polecenia ani, że użytkownicy nie mogą ich stosować. Zwyczajowo stosuje się je przy różnych okazjach, co pozwala z długiej listy różnych nazw plików wyodrębnić te, które posiadają nazwy zawierające pewne przyrostki i poddawać je selektywnie pewnym operacjom, na przykład kopiowania `cp *.c bak # skopiowanie plików *.c do podkatalogu bak`

W Linuksie spotyka się wiele przyrostków nazw plików rozpoznawanych przez powszechnie używane programy, na przykład:

- *.a* - pliki dla poleceń `ar` i `ld` (odpowiedniki DOS-owych plików **.LIB*).
- *.c* - pliki z programami źródłowymi w języku C.
- *.f* - pliki z programami źródłowymi w języku *Fortran*.
- *.h* - pliki z wierszami nagłówkowymi dla programów w języku C.
- *.o* - pliki modułów wynikowych (odpowiedniki DOS-owych plików **.OBJ*)
- *.p* - pliki z programami źródłowymi w języku *Pascal*.
- *.z* - pliki skompresowane za pomocą polecenia `packo`
- *.Z* - pliki skompresowane za pomocą polecenia `compress`.
- *.gz* - pliki skompresowane za pomocą polecenia `gzip` (*GNU zip*).
- *.tar* - pliki archiwów stworzone za pomocą polecenia `tar`.
- *.tgz* - pliki utworzone za pomocą polecenia `tar`, a następnie skompresowane programem `gzip`.

Pliki zwykłe

Pliki zwykłe służą do przechowywania danych i programów. Są to pliki tekstowe lub binarne, z którymi użytkownik ma najczęściej do czynienia. Zapisane są w pamięci dyskowej i zawierają informacje wpisane np. za pomocą edytora lub innych poleceń systemowych bądź programów użytkowych. Pliki zwykłe składają się z bajtów danych uporządkowanych w postaci jednowymiarowej tablicy. System nie nakłada na informację przechowywaną w pliku żadnych ograniczeń czy też wymagań strukturalnych (mogą to robić programy użytkowe).

Podział na pliki tekstowe i binarne wynika bardziej z ich wzrokowej czytelności dla użytkownika (oglądającego je na ekranie lub wydruku) niż z uwarunkowań systemowych.

Pliki tekstowe zawierają teksty różnych dokumentów, plików konfiguracyjnych lub teksty źródłowe programów. Pliki tego typu są wizualnie łatwo czytelne dla użytkownika i mogą być przez niego tworzone i modyfikowane między innymi za pomocą standardowych edytorów tekstowych. Czytelność plików tekstowych jest wynikiem podziału pliku na wiersze opatrzone znacznikami końca wiersza.

W systemach Linux i Unix *znacznikiem nowego wiersza* w plikach tekstowych jest pojedynczy znak *LF*. A nie *CR-LF* - jak to bywa w wielu innych systemach operacyjnych, na przykład w systemie MS DOS.

Pliki binarne nie zawierają standardowego tekstu, lecz przechowują programy wykonywalne lub pewne rodzaje danych. Na plik z programem binarnym (plik wykonywalny) składa się ciąg bajtów w takim porządku, w jakim program jest ładowany do pamięci w celu jego wykonania. Niektóre polecenia systemowe tworzą bardziej złożone struktury danych np. kompilator języka C generuje, a program konsolidujący (inaczej łączący - ang. *Linker*) przyjmuje do przetwarzania określony format pliku wynikowego. Podobnie bywa w przypadku systemów zarządzania bazami danych i wielu innych programów. Format używanych plików jest znany tylko tym programom. Wewnętrzną strukturę plików określają programy, a nie Linux. Z punktu widzenia systemu zwykły plik jest niestrukturalnym ciągiem bajtów o dostępie bezpośrednim.,

Katalogi

Katalog (inaczej: *kartoteka* - ang. *directory*) jest pojęciem wprowadzonym po to, aby uprościć uporządkowanie danych przechowywanych na dysku. Katalog widziany przez użytkownika grupuje wydzielony (tematycznie lub użytkowo) zestaw plików. W rzeczywistości katalogi zawierają nie pliki, lecz informacje niezbędne do ich odszukania. Katalog przypomina trochę teczkę, która zamiast dokumentów zawiera spis informacji o tym, gdzie te dokumenty można odnaleźć, w szczególnym przypadku mogą to być odesłania do innych teczek, czyli innych podkatalogów.

Katalog jest plikiem zawierającym listę nazw innych plików, przy czym z każdą nazwą jest związany (niewidoczny bezpośrednio dla użytkownika) odsyłacz do dokładnego opisu pliku nazywanego *i-węzłem* (ang. *i-node*). Każda z pozycji katalogu może być również katalogiem, czyli katalog, może zawierać podkatalogi niższego poziomu. Katalog może mieć nazwę budowaną według reguł ogólnie obowiązujących dla plików.

Struktura drzewa katalogów

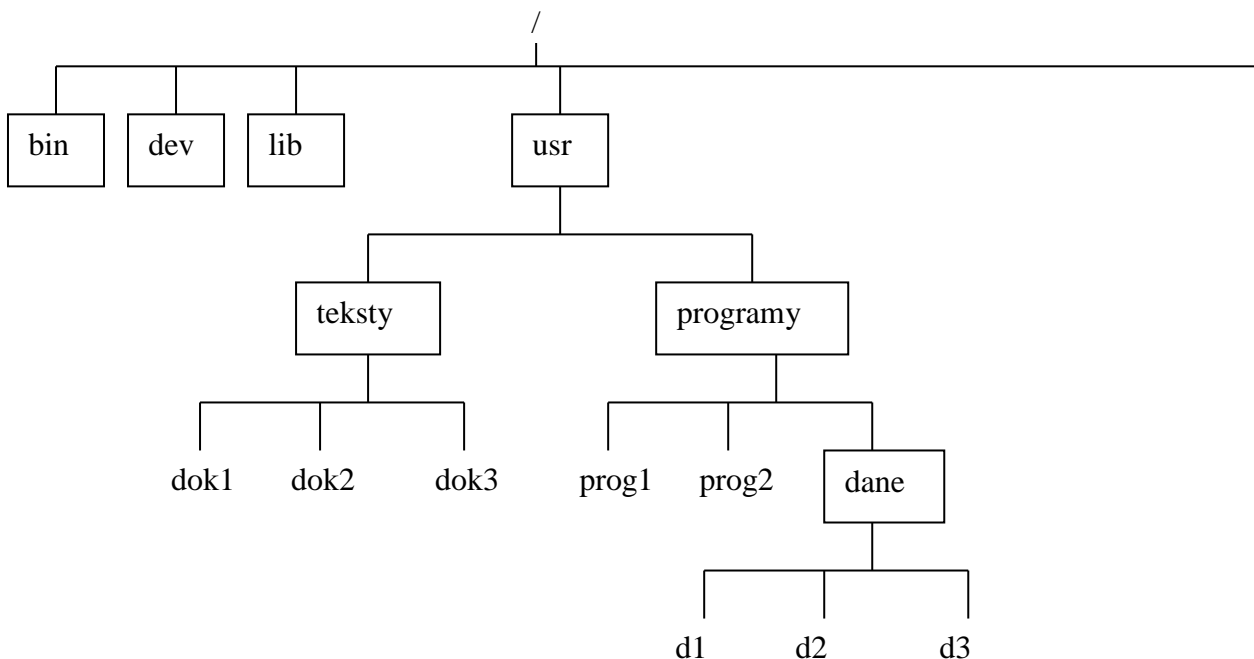
Informacje zapisywane na dysku w postaci plików i katalogów tworzą pewną strukturę hierarchiczną (drzewiastą). Takie określenie nie jest jednak w pełni ścisłe, ponieważ w Linuksie (podobnie jak w Uniksie) do fizycznie tego samego pliku w różnych katalogach mogą być utworzone *dowiązania* (ang. *links*). Ponieważ jednak wspomniany mechanizm jest w praktyce stosowany dość oszczędnie, w dalszym opisie stosowane będą terminy *drzewo plików* oraz *struktura drzewiasta* (*hierarchiczna*).

W każdym głównym systemie plików (zawierającym bazową część systemu operacyjnego) istnieje *katalog główny* o systemowej nazwie / (*slash*), w którym są zapisane zarówno informacje o plikach, jak i o podkatalogach niższego poziomu. Katalog główny systemu zawiera podkatalogi. Katalogi zapewniają łączność pomiędzy nazwami plików i samymi plikami, tworząc strukturę systemu plików. W odróżnieniu od zwykłych plików, użytkownik nieuprzywilejowany nie może stosować do nich zwykłych operacji zapisu i czytania ani modyfikować ich za pomocą edytorów. Do operacji na katalogach istnieje więc wyodrębniony zestaw poleceń. Pod wszystkimi innymi względami (z punktu widzenia systemu) katalog jest normalnym plikiem systemu plików.

Ścieżki dostępu do plików

W Uniksie stosowana jest uniwersalna konwencja identyfikowania plików, wykorzystująca pojęcie *ścieżki dostępu do pliku*, przy czym termin *ścieżka* odnosi się do drzewiastego obrazu systemu plików i obejmuje wszystkie nazwy węzłów tego drzewa (nazwy katalogów), przez które należy przejść, aby dotrzeć do pliku.

Ścieżka dostępu jest zapisywana jako ciąg nazw katalogów, oddzielonych znakiem /, np. */usr/programy/dane*



Rozróżniamy:

- *ścieżkę pełną* (inaczej: *bezwzględną*) - zaczynającą się od katalogu głównego; na przykład dla pokazanego na rysunku pliku *dok1* jest to ścieżka */usr/teksty/*
- *ścieżkę względną* - określoną względem aktualnego katalogu bieżącego; np., jeżeli katalogiem bieżącym jest katalog */usr/programy* to dla pliku *d1* ścieżką względną jest *./dane/* (kropka oznacza katalog bieżący).

Pełna nazwa pliku (określana również jako *ścieżkowa nazwa pliku*) składa się ze ścieżki pełnej i z *bazowej nazwy pliku*. na przykład dla pliku o bazowej nazwie *d3* jego pełną nazwą zgodnie z powyższym rysunkiem jest */usr/programy/dane/d3*.

Wszystkie katalogi, z wyjątkiem głównego, mają swój *katalog nadrzędny*, (w którym są umieszczone). Mogą zawierać również *katalogi podrzędne*, czyli *podkatalogi*.

Przy zakładaniu katalogu tworzone są w nim dwa standardowe zapisy o nazwach: *.(kropka)* oraz *..(dwie kropki)*.

- Nazwa *.* w każdym katalogu oznacza dany katalog. Tak, więc program może czytać dane z katalogu bieżącego, nie znając jego pełnej nazwy, odwołując się do niego przez nazwę *.* (kropka). Nazwą tą posługujemy się najczęściej przy kopiowaniu plików z innego katalogu do katalogu bieżącego, identyfikowanego przez nazwę *."* (znak kropka),
- Nazwa *..* identyfikuje katalog nadrzędny, to znaczy katalog, w którym utworzono katalog bieżący. Pozwala to w łatwy sposób "poruszać się" za pomocą polecenia *cd* po drzewie plików.

Katalog zawierający tylko nazwy *.*, *..* jest traktowany jako *katalog pusty*

Przegląd poleceń do obsługi katalogów

- *pwd* (od ang. *print working directory*) - wypisanie nazwy aktualnego katalogu roboczego
- *cd kat* (od ang. *change directory* - zmień katalog) - przejście do katalogu o nazwie *kat*, umieszczonego w katalogu bieżącym
- *mkdir kat* (od ang. *make directory* - utwórz katalog) - utworzenie w katalogu bieżącym katalogu o nazwie *kat*
- *rmdir kat* (od ang. *remove directory* - usuń katalog) - usunięcie pustego katalogu o nazwie *kat* z katalogu bieżącego
- *mv k1 k2* (od ang. *move* - przenieś) - zmiana nazwy katalogu *k1* na nazwę *k2*, o ile *k1* jest katalogiem
- *mv k1 p/k2* - przeniesienie całego katalogu *k1* w inne miejsce wyznaczone przez ścieżkę dostępu *p* i nadanie mu tam nazwy *k2* (przy założeniu, że *k1* jest nazwą katalogu, a katalog *k2* nie istniał)
- *ls* - wypisanie (żargonowo: *wylistowanie*) zwartej listy nazw plików katalogu bieżącego (bez nazw plików ukrytych)
- *ls -l* - wypisanie nazw i opisów plików katalogu bieżącego (bez nazw plików ukrytych) w formacie pełnym (*-l* od ang. *Long Format*)
- *ls -l /* - wypisanie nazw i opisów plików katalogu głównego
- *ls -a* - wypisanie nazw wszystkich plików z katalogu bieżącego, w tym nazw plików ukrytych (zaczynających się znakiem kropki).
- *ls -R kat* - wypisanie nazw plików katalogu *kat* i wszystkich jego podkatalogów

Wzorce nazw plików i ich rozwijanie

Wzorce nazw plików składają się z pożądaných fragmentów nazw tych plików oraz ze *znaków dowolności* pasujących do nazw plików dostępnych dla danego polecenia::

- *** - oznacza dowolny ciąg znaków pasujący do nazw plików.
- *?* - oznacza dowolny pojedynczy znak pasujący do nazw plików
- *[znak...]* - oznacza jeden ze znaków wymienionych w nawiasach

- `[z1 -z2]` – oznacza jeden ze znaków z leksykograficznego przedziału od `z1` do `z2`, - np. `[d -h]` – oznacza jeden ze znaków: d, e, f, g, h.

Dwa ostatnie zapisy mogą być łączone, przykładowo `[ac-f]` odpowiada jednemu ze znaków: a, c, d, e, f. Ponadto zapisy `[!...]` i `[^...]` oznaczają dowolny znak, z wyjątkiem wymienionych po znaku `!` lub `^`.

Polecenia do obsługi plików i katalogów

Wypisywanie informacji o plikach (ls)

Do wypisywania informacji o plikach oraz zawartości katalogów służą polecenia `ls`.

Składnia polecenia `ls` przedstawia się następująco:

`ls [opcje] [argument] ...`

Argumenty mogą być nazwami plików lub katalogów bądź ich wzorcami. Nazwy plików oraz ich wzorce mogą być poprzedzone ścieżką dostępu do tych plików. Jeżeli pominięto *argumenty* polecenie `ls` wyświetla zawartość bieżącego katalogu.

Opcje polecenia ls

- `-a`, `--all` - wypisanie nazw wszystkich plików, włącznie z plikami ukrytymi, których nazwy zaczynają się znakiem kropki
- `-l`, `--long` - wypisanie informacji w tzw. *formacie długim* (wielokolumnowym), pokazującym atrybuty plików; domyślnym *formatem jest format krótki* (tylko lista samych nazw)
- `-C` - w przypadku wypisywania w formacie krótkim (bez opcji `-l`) - porządkowanie nazw w dół wypisywanych kolumn nazw
- `--color` - stosowanie różnych kolorów do rozróżniania nazw plików według ich typów (kolory są dobierane automatycznie, w zależności od ustawionego tła ekranu)
- `-d`, `--directory` - jeżeli w generowanej liście nazw znajdują się katalogi - pokazywanie tylko ich nazw, bez ich zawartości (anulowanie opcji `-R`)
- `-F` - w przypadku wypisywania nazw w krótkim formacie - umieszcza po nazwie znacznik informujący o rodzaju pliku

Litera `d` na początku wypisanych wierszy informuje nas, że wypisany plik jest katalogiem. Nazwy zaczynające się od znaku kropki., są to *pliki ukryte*, które zostały pokazane dzięki opcji `-a` (od. ang. *all*)

Kopiowanie plików i katalogów (cp)

Do kopiowania plików i katalogów służy polecenie `cp` o następującej składni:

```
cp [opcje] plik1 plik2
cp [opcje] plik ...katalog
```

Opcje polecenia cp

- `-f`, `--force` - zastępowanie plików w katalogu docelowym

- -i, --interactive - pytanie o potwierdzenie zastąpienia plików w katalogu docelowym
- -l, --link -tworzenie dowiązań zwykłych zamiast kopii plików
- -p, --preserve - zachowywanie wszystkich atrybutów plików
- -r, -R, --recursive - kopiowanie również napotkanych podkatalogów (-R - włącznie z plikami specjalnymi)
- -s, --symbolic-link - tworzenie dowiązań symbolicznych zamiast kopii plików
- -S, --suffix *przyrostek* - określa przyrostek nazw dla zachowywanych plików
- -u, --update - uaktualnianie katalogów o nowsze wersje plików

Usuwanie plików (rm)

Do usuwania plików (i podkatalogów) służy polecenie

`rm [opcje} plik ...`

Opcje polecenia rm

- -d, --directory - usuwanie podkatalogów, nawet, jeżeli nie są puste
- -f, --force - usuwanie plików zabezpieczonych przed zapisem bez pytania o pozwolenie
- --help - wyświetlenie opisu polecenia
- -i, --interactive - pytanie, czy usunąć plik (odpowiedź: y- tak, n -nie)
- -r, -R, --recursive - jeśli plik jest katalogiem, usunięcie go wraz z całą zawartością

Wypisanie jednego lub kilku połączonych plików (cat)

Do wypisywania (zwykle wyświetlania) zawartości jednego lub kilku wyspecyfikowanych plików służy polecenie:

`cat [opcje} plik [plik...]`

Opcje polecenia cat

- -b - numerowanie wszystkich wypisanych wierszy zawierających tekst
- -n - numerowanie wszystkich wypisanych wierszy
- -S - zastępowanie sąsiednich pustych wierszy jednym pustym
- -E - umieszczanie na końcu wierszy znaku \$
- -O - pokazywanie znaków niedrukowalnych
- -T - pokazywanie znaku Tab jako ^I

Elektroniczny podręcznik (man)

W tekstowym środowisku systemów uniksowych typową pomocą jest *elektroniczny podręcznik – manual*, którego rozdziały można oglądać po wydaniu polecenia:

`man [opcje] [sekcja] [tytuł]`

Większość rozdziałów poświęcona jest poleceniom, stąd typowa forma wywołania tego polecenia to:

man polecenie

np.

man ls

Krótkie odpowiedzi poleceń (opcja `-help`)

Wiele poleceń Linuksa wywołanych z opcją `--help` lub `-h` wyświetla swoją składnię oraz krótki opis działania i opcji. Jest to cecha głównie poleceń posiadających długą listę opcji, ale spotykana również w poleceniach o kilku opcjach, np. `mkdir`.

```
# mkdir -help
```