Systemy Operacyjne I: System plików

Andrzej Stroiński

Politechnika Poznańska

18 marca 2014

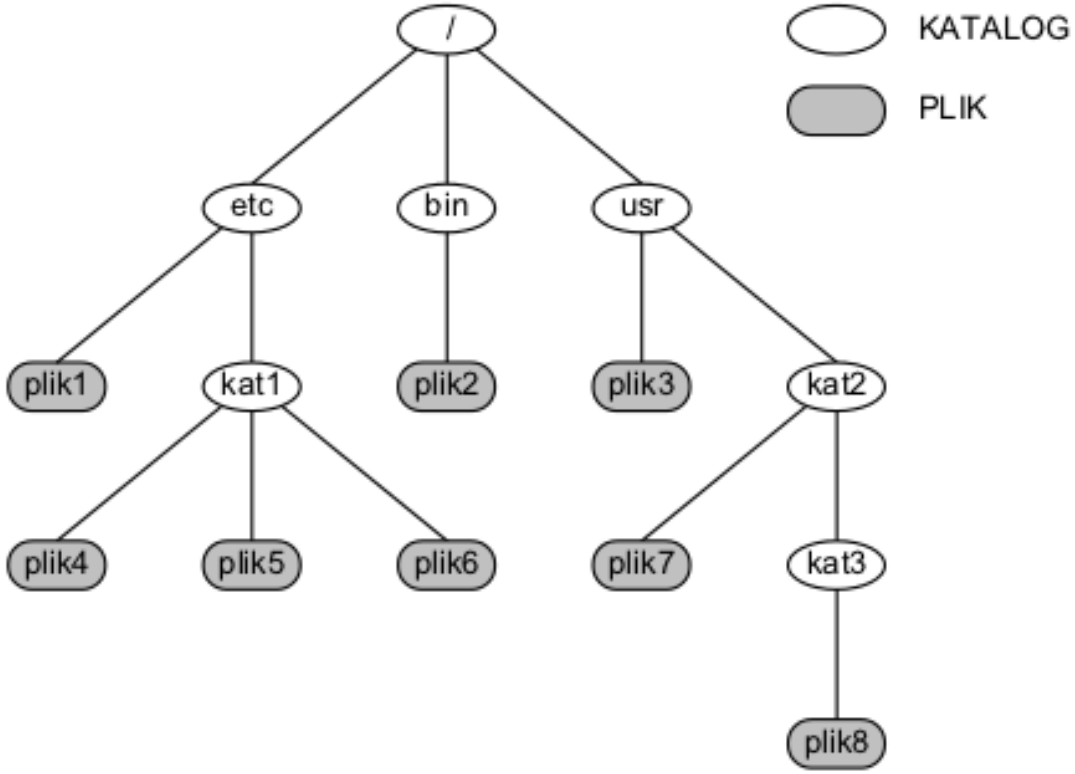
Materiały

- Prezentacja oraz inne materiały zostały przygotowane na podstawie:
  - Użytkowanie systemu operacyjnego UNIX - *dr D.Wawrzyniak*
  - Systemy operacyjne - skrypt - *dr C.Sobaniec*
  - Strona przedmiotu
  - Strona *Dr A.Kobusińskiej*
  - Strona *K.Sieka*
  - pomoc systemowa `man`
  - inne...

- W systemie `UNIX` miejsce trwałego przechowywania danych nazywane jest systemem plików. Wyróżnia się następujące rodzaje plików:
  - plik zwykły (*ang. normal file*) - elementarny zbiór danych identyfikowany przez system po swojej nazwie i położeniu w systemie plików
  - katalog (*ang. directory*) - zawiera pliki (w tym również kolejne katalogi)
  - urządzenie (*ang. device*)
  - plik specjalny (link symboliczny, kolejka FIFO itp.)

- Powszechnie stosowana hierarchia katalogów w systemach operacyjnych z rodziny `UNIX`.
  - `/` — root
  - `/bin, /sbin` — binaries, system binaries
  - `/boot` — boot loader
  - `/dev` — devices
  - `/etc` — local configuration
  - `/home` — personal files
  - `/lib` — libraries for `/bin, /sbin`
  - `/media, /mnt` — mount points: removable devices, temporary
  - `/opt` — optional packages
  - `/proc` — kernel, process status
  - `/tmp` — temporary files (may be erased at reboot)
  - `/var` — variable files: log, spool, www, tmp
  - `/usr` — read-only user files: bin, lib, share, src

- Przykład struktury hierarchicznej w systemie `UNIX`.



- Nazwy plików są pojedynczymi wyrazami i mogą składać się z dowolnych liter, cyfr oraz kilku dozwolonych znaków, takich jak np. kropka "." lub znak łącznika "_".
- *krótka nazwa pliku* — na samym dole w hierarchii np. `plik8`.
- System `UNIX` rozróżnia wielkie i małe litery. `plik8 != Plik8`
- *ścieżka dostępu do pliku* — podająca wszystkie katalogi, jakie należy przejść aby znaleźć dany plik. `usr/kat2/kat3/`
- *pełna nazwa pliku* — jest połączeniem ścieżki dostępu i nazwy krótkiej. `/usr/kat2/kat3/plik8`

Systemy operacyjne

- Ścieżki stosowane w systemach operacyjnych z rodziny `UNIX`.
  - `/` — root, korzeń drzewa katalogów
  - `~` — katalog domowy (`$HOME`)
  - `.` — katalog bieżący (`$PWD`)
  - `..` — rodzic (katalog wyżej)

- Wyrażenia uogólniające.
  - * — dopasowanie dowolnej liczby dowolnych znaków np.
    `ls /home/*`
    listuje pliki w katalogu `/home/` *i wszystkie podkatalogi*
  - ? — dopasowanie do jednego znaku np.
    `$ls /home/????`
    `/home/aaaa /home/bbbb`
  - {} — zbiór słów np.
    `$mkdir /home/{asia,basia,kasia}`
    tworzy katalogi `/home/asia/ /home/basia /home/kasia`
  - [] — dowolny znak ze zbioru znaków np.
    `$ls plik[132]`
    `plik1 plik2 plik3`
    `$ls plik[1-3]`
    `plik1 plik2 plik3`
  - łączenie wzorcy
    `$ ls pl*`
    `plik1 plik2 plik-test`

Nawigacja po systemie plików

`pwd`

Polecenie wypisujące na standardowe wyjście pełną nazwę katalogu w, którym się znajduje zalogowany użytkownik systemu.

Opcje:

`--help` — Wyświetla informację o stosowaniu programu i dostępnych opcjach, kończy pracę.

`--version` — Wyświetla numer wersji programu i kończy pracę.

Przykład: `$ pwd`

Nawigacja po systemie plików

cd

Polecenie pozwalające na zmianę aktualnego katalogu roboczego.
Argumenty:
`[nazwa_katalogu]` — Pełna nazwa katalogu do którego użytkownik chce przejść lub ścieżka względna (względem aktualnego katalogu roboczego).
`..` — Cofnięcie się do poprzedniego katalogu.
Przykład:

```
$ cd /home/inf76543/test/
$ cd ..
```

Nawigacja po systemie plików

`tree`

Wyświetla katalogi, podkatalogi oraz pliki znajdujące się w aktualnym katalogu roboczym lub argumencie polecenia.

Argumenty:

`[nazwa_katalogu]` — Pełna nazwa katalogu lub ścieżka względna (względem aktualnego katalogu roboczego).

Przykład:

```
$ tree /home/inf76543/test/
$ tree
```

Nawigacja po systemie plików

`ls`

Program wywołany bez parametrów wyświetli tylko katalogi i pliki (w obecnym katalogu lub podanym jako argument).

Opcje:

`-l` — Wyświetla dokładniejsze informacje na temat listowanego katalogu (typ pliku, prawa dostępu, liczba powiązań, właściciel, grupa przypisana do pliku, rozmiar, data modyfikacji i nazwa elementu).

`-a` — Parametr umożliwia wyświetlenie także ukrytych plików (poprzedzonych kropką) oraz symbole ”.” lub ”. .”, które oznaczają odpowiednio katalog aktualny i nadrzędny.

Nawigacja po systemie plików

`ls (2)`

Argumenty:

`[nazwa_katalogu]` — Pełna nazwa katalogu lub ścieżka względna (względem aktualnego katalogu roboczego).

Przykład:

```
$ ls /home/inf76543/test/
$ ls ..
$ ls -a .
$ ls -a .  ..
```

`mkdir` — (ang. make directory)

Polecenie to bez żadnych dodatkowych opcji tworzy katalog podany jako argument. Jeśli argumentów będzie więcej, to odpowiednia liczba katalogów zostanie utworzona.

Składnia: `mkdir [opcje] [katalog(i)]*`

Opcje:

`-m` — Pozwala na ustawienie praw dostępu do tworzonego katalogu. Po opcji podajemy wartość oktalną praw.

`-p, -parents` — utwórz brakujące nad katalogi

`-v, -verbose` — Pisz co robisz.

Argumenty:

`[nazwa_katalogu]` — Nazwa tworzonego katalogu.

Przykład: `$ mkdir -m 700 test`

Zarządzanie systemem plików

`rmdir` — (ang. remove directory)

Usuń katalog.

`-p, -parents` — utwórz brakujące nad katalogi

`-v, -verbose` — Pisz co robisz.

`[nazwa_katalogu]` — Nazwa katalogu do usunięcia.

Przykład:

```
$ rmdir test/
```

Zarządzanie systemem plików

cp — (ang. copy)

Kopiowanie plików lub katalogów.
`-r, -R` — Rekurencyjnie, czyli skopiuj również zawartość (pliki i podkatalogi).
`-u, -update` — Skopiuj tylko nowsze wersje.
`-i, -interactive` — Zapytaj o zgodę przed nadpisaniem.
`-v, -verbose` — Pisz co robisz.
`-f, -force` — Nie pytaj o zgodę przed nadpisaniem.
Przykład:

```
$ cp plik1 plik2
```

Zarządzanie systemem plików

`mv` — (ang. move)

Przenoszenie plików, lub zmiana ich nazwy.

`-u, -update` — Skopiuj tylko nowsze wersje.

`-f, -force` — Nie pytaj o zgodę przed nadpisaniem.

Przykład:

```
$ mv plik1 plik2
```

`rm` — (ang. remove)

Usuwanie plików.

`-r, -R, -recursive` — Usuwanie katalogów z zawartością.

`-i, -interactive` — Zapytaj o zgodę przed skasowaniem.

`-f, -force` — Nie pytaj o zgodę przed usunięciem.

Przykład:

```
$ rm plik1
```

Zarządzanie systemem plików

- Dowiązania twarde (*ang. hard links*) — Dowiązanie twarde to utworzenie nowej nazwy dla istniejącego pliku. Jeśli wyobrazimy sobie plik jako jego nazwę i dane na które ta nazwa wskazuje to można powiedzieć, że dowiązanie twarde to utworzenie nowej nazwy wskazującej na te same dane. Polecenie to jest podobne do kopiowania pliku z tym, że przy kopiowaniu tworzona jest niezależna kopia pliku a przy dowiązaniu twardym tworzona jest tylko nowa nazwa, która wskazuje na ten sam istniejący już plik. Do tworzenia dowiązań twardych służy polecenie `ln`.

Zarządzanie systemem plików

- Dowiązania symboliczne (*ang. symbolic links*) — Link symboliczny to dowiązanie do pliku wskazujące na jego nazwę. O ile dowiązanie twarde to była nowa nazwa wskazująca na plik tak samo jak stara to link symboliczny wskazuje na nazwę pliku, która dopiero wskazuje na plik. Linki symboliczne tworzy się analogicznie jak dowiązania twarde tylko dodając do polecenia `ln` parametr `-s`. Na przykład `ln -s plik1 plik2` utowrzy `plik2` jako dowiązanie symboliczne do `plik1`.

Zarządzanie systemem plików

`ln` — (ang. link)

Tworzy dowiązanie do pliku.

Składnia: `ln [nazwa_pliku] [nazwa_dowiązania]`

`-s, -symbolic` — link symboliczny (do etykiety pliku)

`-P, -physical` — link fizyczny (do zawartości pliku)

Przykład:

```
$ ln plik1 link1
```

`file`

Podaje typ pliku.

```
$ file plik1
```

Zarządzanie systemem plików

stat

Podaje informacje o pliku pobrane z inode.

Przykład:

```
$ stat plik1
```

touch

Tworzy pusty plik lub modyfikuje jego czas dostępu jeśli już istniał.

```
$ touch plik1
```

cat

Wypisywanie zawartości pliku na konsolę.

```
$ cat plik1
```

Prawa dostępu

chmod

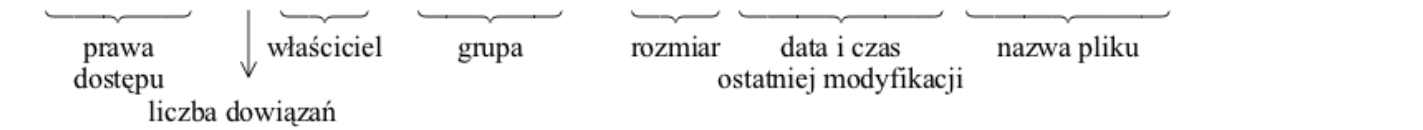
Zmienia uprawnienia do pliku. Do sprawdzenia używaj `$ ls -l`.
`-R` — Rekurencyjnie (podkatalogi i pliki).
`[ugoa] [+-=] [rwx]` — Ustal przywileje do zapisu, odczytu, wykonywania dla właściciela, grupy, innych użytkowników lub wszystkich użytkowników.
`[0-7] [0-7] [0-7]` — Ustal przywileje w zapisie oktalnym.
Przykład:

```
$ chmod a+rx plik1.
$ chmod 755 plik1
```

Prawa dostępu

- Pełna informacja o pliku: Polecenie `ls` posiada kilka opcji podawanych po znaku -. Opcja `-l` pozwala poznać pełną informację o plikach w katalogu bieżącym.

```
% ls -l
total 56
-rw-r--r--   1 darek    student      136 Apr 10 19:16 plik1
-rw-r--r--   1 darek    student      136 Apr 10 19:19 plik2
-rw-r--r--   1 darek    student      136 Apr 10 19:20 plik3
-rw-r--r--   1 darek    student       18 Apr 10 19:25 plik_inny
drwxr-sr-x   2 darek    student      512 Apr 10 19:29 podkatalog1
drwxr-sr-x   2 darek    student      512 Apr 10 19:30 podkatalog2
-rw-r--r--   1 darek    student       13 Apr 10 19:26 skrypt
%
```



Prawa dostępu

- Prawa (trzy kolejne litery) podawane są dla właściciela pliku, użytkowników grupy i pozostałych użytkowników.
- Prawa dostępu do plików zwykłych:
  - `r` — prawo do odczytu zawartości (umożliwia również kopiowanie)
  - `w` — prawo do zapisu (zmiany zawartości pliku lub usunięcia zawartości)
  - `x` — prawo do uruchomienia (dotyczy plików zawierających programy binarne lub skrypty)
- Prawa dostępu do katalogów:
  - `r` — prawo do przeglądania (np. umożliwia wykonanie polecenia `ls`)
  - `w` — prawo do tworzenia, usuwania i zmiany nazw plików
  - `x` — prawo dostępu do plików

Prawa dostępu

- Prawa dostępu do pliku mogą być zmienione tylko przez właściciela i użytkownika uprzywilejowanego o nazwie `root`. Do zmiany praw służy polecenie `chmod`. Przykłady użycia polecenia `chmod`:
  - `chmod u+x` — plik dodanie prawa wykonywania właścicielowi
  - `chmod g-w` — plik zabranie prawa zapisu grupie
  - `chmod a+r` — plik dodanie prawa odczytu wszystkim użytkownikom
  - `chmod o=x` — plik zmiana praw pozostałych użytkowników na prawo wykonywania
  - `chmod o=` — plik wyzerowanie praw pozostałych użytkowników
  - `chmod 777` — plik nadanie wszystkich praw wszystkim użytkownikom (prawa w postaci ósemkowej)

Prawa dostępu

chown

Zmienia właściciela pliku podanego jako argument.
`-R` — Rekurencyjnie (podkatalogi i pliki).
`[username]` — Nazwa użytkownika, który stanie się nowym właścicielem pliku lub katalogu.
`[username:group]` — Nazwa grupy i użytkownika z niej, który stanie się nowym właścicielem pliku lub katalogu.
Przykład:

```
$ chown inf70001 plik
$ chmod chown inf 70001:users plik
```

chgrp

Zmienia grupę użytkowników pliku podanego jako argument.
`-R` — Rekurencyjnie (podkatalogi i pliki).

Pliki ukryte

- Utworzenie pliku ukrytego polega na nadaniu nazwy zaczynającej się od znaku ".".
- Jakiekolwiek operacje na plikach z wzorcem uogólnionym "*" (kopiowanie `cp` , usuwanie `rm` itp.) nie dotyczą plików ukrytych.
- Wzorzec obejmujący pliki ukryte składa się z gwiazdki poprzedzonej kropką ".*."
- W celu wyświetlenia plików ukrytych należy użyć opcji `-a` w poleceniu `ls`, np.:
  ls -a; ls -al

Wyszukiwanie plików

`locate`

Polecenie `locate` służy do bardzo szybkiego wyszukiwania plików, nie szuka ich w katalogach lecz wykorzystuje specjalną bazę danych nazw plików oraz ich lokalizacji. Aby polecenie działało prawidłowo należy najpierw stworzyć taką bazę danych. Do tego celu służy programem `updatedb`, który uruchamiamy jako użytkownik `root`. Pewną wadą takiej bazy jest to, że nie odzwierciedla ona natychmiast zmian w systemie plikowymi staje się nieaktualna, kiedy doda się lub usunie pliki. Polecenie `locate` jest łatwe do użycia np. jeżeli chcemy znaleźć wszystkie pliki w formacie PostScript w naszym komputerze.

Przykład:

```
$ locate ''*.ps'' | more
$ locate ''/etc/*.conf''
```

Wyszukiwanie plików

more

Polecenie more służy do przeglądania długich plików.

Przykład:

```
$ more plik.txt
```

Wyszukiwanie plików

find

Polecenie `find` służy do wyszukiwania plików w systemie.

Składnia: `find [katalog_początkowy] [warunki] [akcje]`

Warunki:

`-name` — nazwa pliku (nazwy uogólnione również)

`-iname` — j/w, case-insensitive

`-type` — typ pliku (`f` - zwykły plik, `d` - katalog)

`-empty` — pusty plik

Akcje:

`-exec` — wykonaj polecenie

`-ok` — zapytaj użytkownika, a następnie wykonaj polecenie

`-quit` — zakończ działanie

Przykład:

```
$ find .  -name ''*.tex'' -ok cat ''{}'' \;
```

Zadania

- A teraz przechodzimy do praktycznej części zajęć.