

# Wstęp do obsługi Linux'a

Katarzyna Grzelak

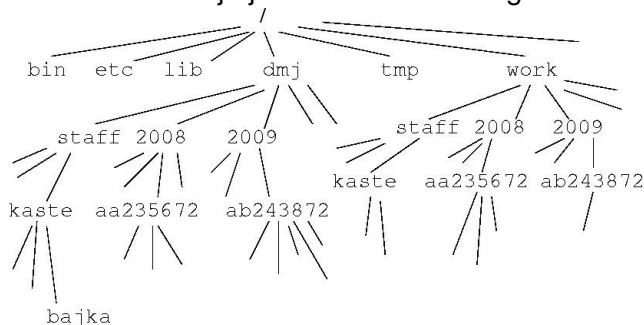
październik 2018

- Programy służące do zarządzania wykonywaniem innych programów: **systemy operacyjne** (ang. operating system)
- Komputer w każdej chwili wykonuje jakiś program:  
albo program użytkowy (**aplikację**)  
albo fragment systemu operacyjnego
- W OKWF zainstalowany jest system operacyjny Linux w wersji Fedora

- Tylko jeden super-użytkownik administrujący systemem
- Zwykli użytkownicy mają możliwość modyfikacji tylko własnych plików i katalogów

# Drzewo katalogowe

- Użytkownik uzyskuje dostęp do danych za pośrednictwem **drzewa katalogowego**.
- W MS Windows drzewa katalogowe oznaczane są literami alfabetu : **C:, D:, F: ...**
- W Linux'ie istnieje jedno drzewo katalogowe



# Katalog domowy i roboczy

- Po wpisaniu nazwy użytkownika i hasła, użytkownik podłącza się do swojego domowego katalogu
- W katalogu domowym znajdują się zwykłe pliki i katalogi oraz **pliki i katalogi ukryte**, których nazwy zaczynają się od kropki (na ogół są to pliki konfiguracyjne, tworzone automatycznie przez używane programy)
- Każdy student, np. `aa235672` ma dostęp do dwóch ograniczonych obszarów dysku: `/dmj/2018/aa235672` (katalog domowy, robione kopie zapasowe) i `/work/2018/aa235672` (tzw. katalog roboczy, większa ilość miejsca)

- Użytkownik wpływa na działanie systemu operacyjnego za pomocą **graficznego lub tekstowego interfejsu użytkownika** (ang. user interface)
- **Interfejs użytkownika = aplikacja (zbiór aplikacji) pozwalająca na wydawanie poleceń systemowi operacyjnemu**
- Obecnie w OKWF zainstalowany jest graficzny interfejs `Gnome`
- Przełączanie się pomiędzy trybem graficznym a tekstowym:  
`np. Ctrl-Alt-F4. Powrót Ctrl-Alt-F1.`

- Powłoka (*ang. shell*) to program, który pozwala użytkownikowi na wpisywanie poleceń, a następnie przekazuje je do wykonania przez system operacyjny
- **Przykład działania powłoki**, w okienku Terminal wpisujemy dowolne polecenie, np. `pwd`:
  - poszukiwanie programu (plik z uprawnieniami do wykonywania (x)) o nazwie `pwd`  
(= przeglądanie katalogów zdefiniowanych w zmiennej `$PATH`)
  - uruchomienie programu
  - wynik działania programu w oknie terminala
- Polecenia Linux'a można też zapisywać jedno po drugim w plikach tekstowych, tzw. skryptach i uruchamiać je poprzez uruchomienie skryptu

- **W poleceniach rozróżniane są wielkie i małe litery**
- **Zwyczajowe** nazwy katalogów: np. w katalogu `bin` przechowywane są wykonywalne wersje programów, w katalogu `src` wersje źródłowe, w katalogu `home` katalogi domowe użytkowników ...
- **Kropka w nazwie pliku nie jest znakiem wyróżnionym**
- **Umowne** rozszerzenia plików: `.cxx`, `.cpp` → programy w języku C++, `.pdf` → pliki w formacie pdf, `.tex` → pliki w formacie  $\text{\LaTeX}$ , `.txt` → pliki w formacie tekstowym ...
- **W nazwach plików i katalogów nie powinno być polskich liter ani spacji**



- Przykłady poleceń:

Zawartość katalogu :	<code>ls</code>
Zawartość katalogu (długi wydruk):	<code>ls -l</code>
Zawartość katalogu (długi wydruk + pliki ukryte ):	<code>ls -a -l</code> lub <code>ls -al</code>
Przejdźcie do katalogu <i>katalog</i> :	<code>cd <i>katalog</i></code>
Powrót do katalogu domowego :	<code>cd</code>
Przejdźcie do nadrzędnego katalogu:	<code>cd ..</code>
Powrót do ostatniego odwiedzanego katalogu :	<code>cd -</code>
Sprawdzanie nazwy aktualnego katalogu:	<code>pwd</code>

- Pełna składnia wywołania np. programu `ls` to `/bin/ls`
- Nazwy katalogu w którym znajduje się program nie trzeba podawać, jeśli jest on umieszczony na specjalnej liście (zmienna środowiskowa `PATH`, `echo $PATH`).
- **Pomoc:** `man nazwa_polecenia`

# Użytkownicy i prawa dostępu w Linux'ie

- Po wywołaniu polecenia `ls -l`, z lewej strony wypisują się kolumny liter `r`, `w`, `x`, które mówią o prawach dostępu do plików
- Właściciela pliku/katalogu może zmienić tylko super użytkownik (ang. superuser)
- Prawa dostępu:
  - **r** - prawo do odczytu pliku,  
dla katalogu prawo do wyświetlenia jego zawarości
  - **w** - prawo do modyfikacji pliku,  
dla katalogu prawo do tworzenia w nim plików/katalogów
  - **x** - dla pliku prawo do wykonania kodu programu/skryptu,  
dla katalogu prawo do wejścia do niego
- Prawa dostępu może zmieniać właściciel pliku/katalogu lub super użytkownik
- Super użytkownik ma identyfikator **0**

- .
  - ..
  - ~
  - ~kaste
  - \*
- katalog bieżący  
katalog nadrzędny (bliżej korzenia)  
własny katalog domowy  
katalog domowy użytkownika `kaste`  
dowolna kombinacja znaków

Oglądanie zawartości pliku tekstowego

```
more nazwa_pliku
```

Oglądanie zawartości pliku tekstowego

```
less nazwa_pliku
```

Kopiowanie:

```
cp skąd dokąd
```

Kopiowanie pliku b.txt do katalogu ~/txt

```
cp b.txt ~/txt
```

Kopiowanie plików o rozszerzeniu .txt do katalogu ~/txt

```
cp *.txt ~/txt
```

Kopiowanie do katalogu bieżącego

```
cp -r /etc/init.d .
```

Zmiana nazwy

```
mv stara_nazwa nowa_nazwa
```

Tworzenie nowego katalogu

```
mkdir nazwa_katalogu
```

Usuwanie pliku

```
rm nazwa_pliku
```

Usuwanie katalogu

```
rm -r nazwa_katalogu
```

Oglądanie struktury katalogu

```
tree nazwa_katalogu
```

# Zmiana praw dostępu w Linux'ie

- u = użytkownik (user)
- g = grupa (group)
- o = reszta (other)

Odejmujemy sobie prawo do pisania

```
chmod u-w nazwa_pliku
```

Odejmujemy innym wszystkie prawa

```
chmod go-rwx nazwa_pliku
```

Dodajemy prawa rekurencyjnie

```
chmod -R go+r nazwa_katalogu
```

# Pakowanie i kompresowanie

Pakowanie katalogu <code>nowy</code>	<code>tar cvf plik.tar nowy</code>
Pakowanie katalogu <code>nowy</code> z kompresją	<code>tar cvzf plik.tar.gz nowy</code>
Oglądanie zawartości pliku <code>plik.tar</code>	<code>tar tvf plik.tar</code>
Odpakowywanie pliku <code>plik.tar</code>	<code>tar xvf plik.tar</code>
Odpakowywanie pliku <code>plik.tar.gz</code>	<code>tar xvzf plik.tar.gz</code>
Kompresja pliku <code>ba.txt</code>	<code>gzip ba.txt</code>
Odkompresowywanie pliku <code>ba.txt.gz</code>	<code>gunzip ba.txt.gz</code>
Rozszerzenie pliku <code>.tgz</code> równoważne z rozszerzeniem <code>.tar.gz</code>	

Polecenie `alias` służy do definiowania nowych poleceń i przedefiniowywania starych.

Jakie nowe polecenia już istnieją ? `alias`

Przykład (tcsh): `alias dir 'ls -la'`

Przykład (bash): `alias dir='ls -la'`

Przykład (tcsh): `alias rm 'rm -i'`

Przykład (bash): `alias rm='rm -i'`

## Przykłady uruchamiania edytora tekstu:

- `gedit plik1.txt &`
- `emacs plik3.txt &`
- `atom plik2.txt`

## lub

- `gedit plik1.txt`
- `Ctrl-Z`
- `bg`



- `cd; ls -l`
- `ls -l | wc`
- `ls -l > lista.txt`
- `ls -l ~/_work_ » lista.txt`

- Przeglądanie zawartości pliku tekstowego  
`cat nazwa_pliku`
- Łączenie dwóch plików tekstowych w jeden  
`cat nazwa_pliku1 nazwa_pliku2 > suma.txt`
- Tworzenie pliku tekstowego (`.tcshrc`) bez użycia edytora tekstu  
`cat > .tcshrc`  
`echo Witam`  
`Ctrl-D`
- Dopisanie nowej linii do pliku tekstowego (`.tcshrc`), bez użycia edytora tekstu. Ciąg liter `KONIEC` można zastąpić dowolnym innym.  
`cat « KONIEC >> .tcshrc`  
`echo Witam`  
`KONIEC`

Za ciągiem kończącym plik (tutaj `KONIEC`) nie może być spacji !

# Linux - inne ważne polecenia

Łączenie się (logowanie) na oddalony komputer `ssh -l użytkownik komputer`

Przesyłanie pliku między komputerami `scp plik użytkownik@komputer:katalog`

Przykładowa nazwa komputera: `primus.okwf.fuw.edu.pl`

Komputer, na który można zalogować się spoza wydziału:

`tempac.okwf.fuw.edu.pl`

# Szukanie plików - przykłady

- Szukanie pliku o nazwie `README` w całym drzewie katalogowym  
`find / -name README`
- Szukanie tylko zwykłych plików we fragmencie drzewa katalogów, poczynając od katalogu bieżącego  
`find . -type f`
- Szukanie plików mających w nazwie ciąg liter `conf`, poczynając od katalogu `/etc`  
`find /etc -name '*conf*'`
- Szukanie w pliku/plikach w katalogu bieżącym wierszy zawierających ciąg liter `abc`  
`grep abc *`
- Szukanie wierszy zawierających ciąg liter `abc`, bez zwracania uwagi na małe i duże litery  
`grep -i abc *`
- Szukanie wierszy zawierających ciąg liter `abc` w plikach z katalogu o danej nazwie i wszystkich jego podkatalogach  
`grep -r abc nazwa katalogu`

- ... do znajdowania samych zwykłych plików
- do znajdowania samych katalogów
- do znajdowania plików ostatnio modyfikowanych
- ... (stosowne opcje do znalezienia za pomocą polecenia `man find`)

- **Dowiązanie symboliczne (link) :**  
`ln -s /work/2012/aa235672 _work_`
- Polecenie `quota -v` służy do sprawdzania dostępnego i aktualnie zajmowanego miejsca na dysku

- Polecenie `printenv` pokazuje wszystkie zmienne środowiskowe
- Tworzenie nowej zmiennej np.  
`setenv PRACA /dmj/2012/ab545632`  
(**tsh**)
- Tworzenie nowej zmiennej np.  
`export PRACA=/work/2012/ab545632`  
(**bash**)
- `echo $PRACA`
- `cd $PRACA`

# Różne typy apostrofów

- Polecenie `echo pwd` powoduje wypisanie na ekran słowa `pwd`
- Polecenie `echo `pwd`` powoduje uruchomienie polecenia `pwd` i wypisanie wyniku tego polecenia na ekran
- Polecenie `echo "$PATH"` (równoważne `echo $PATH`) wypisuje ścieżkę dostępu (`$` jest traktowany jako znak specjalny)
- Polecenie `echo '$PATH'` wypisuje `$PATH` (`$` jest traktowany dosłownie)



- Typ powłoki definiuje administrator systemu w pliku z hasłami
- Obecnie każdy użytkownik w OKWF ma standardowo ustawioną powłokę `bash` (zmienna `$SHELL`)
- Standardowo uruchamiane skrypty:
  - przy logowaniu: `/etc/profile` i `~/.bash_profile`
  - start powłoki: `~/.bashrc`