

# **Wstęp do informatyki**

## **Shell podstawy**

# Podstawowe polecenia

*less plik(i)*

Przeglądaj interaktywnie zawartość plików.  
Można używać klawiszy kursora, PgDn, PgUp,  
etc. Koniec pracy (q), wyszukiwanie /

*"Less is more than more"*

# Podstawowe polecenia

*ls [opcje]*

Pokaż zawartość katalogu(\*)

Wylistuj pliki i katalogi

```
> ls  
WI01.odp  WI01.pdf  WI02.odp
```

Wylistuj wszystko (także ukryte tzn. zaczynające się od .)

```
> ls -a  
.  ..  .svn  WI01.odp  WI01.pdf  WI02.odp
```

(\*) Niebawem opcje zaawansowane

# Podstawowe polecenia

*ls [opcje]*

Pokaż zawartość katalogu

Wylistuj pliki i katalogi

```
> ls  
WI01.odp  WI01.pdf  WI02.odp
```

Wylistuj wszystko (także ukryte tzn. zaczynające się od .)

```
> ls -a  
.  ..  .svn  WI01.odp  WI01.pdf  WI02.odp
```

# Podstawowe polecenia

*mkdir [opcje] ścieżka*

Stwórz katalog

**Katalog tmp**

```
> mkdir tmp
```

**Katalog z podkatalogami**

```
> mkdir -p tmp/abc/xyz
```

# Podstawowe polecenia

*rmdir [opcje] ścieżka*

Usuń puste katalogi

**Katalog tmp:**

```
> rmdir tmp
```

**Katalog z podkatalogami**

```
> rm -p tmp/abc/xyz
```

# Podstawowe polecenia

*rm [opcje] ścieżka*

Usuń pliki i/lub katalogi

**Plik a.txt:**

```
> rm a.txt
```

**Katalog rekursywnie -r bez pytania -f:**

```
> rm -rf tmp
```

# Podstawowe polecenia

Inne użyteczne polecenia operujące na plikach i katalogach:

`mv` - przenieś

`cp` - kopiuje

Przydatne opcje: `-i` (interactive), `-r` (recursive), `-v` (verbose)

`pwd` - pokaż bieżący katalog

# Wzorce plików

*\* - wszystkie pliki*

*a\* - pliki o nazwie zaczynającej się od a*

*??? - trzyliterowe nazwy*

*[abx]\* - pierwsza litera to a, b lub x*

Używane we wszystkich poleceniach interpretowanych przez bash.

Uwaga: Bash **zastępuje dany wzorzec przez nazwy pasujących plików**, a następnie wykonuje tak zmodyfikowane polecenie.

Zagadka: `rm -rf *`

# Strumienie

Proces w Unixie podłączony jest do trzech strumieni:

- 0) wejściowy (standard input) – domyślne wyjście, wszędzie gdzie napisałem ``pokaż" powinno być ``przełącz na standardowe wyjście"
- 1) wyjściowy (standard output)
- 2) diagnostyczny (standard error)

# Strumienie

Przekierowywanie wyjścia do pliku:

```
> ls > lista.txt
```

Dopisanie wyjścia do pliku:

```
> ls >> lista.txt
```

Przekierowanie wyjścia diagnostycznego:

```
> ls 2> err.txt
```

Przekierowanie wejścia:

```
> cat < t.txt
```

# Filtry

Umożliwiają podłączenie wyjścia jednego procesu do innego (pipes). Można wielokrotnie używać. Jest to powszechnie stosowana metoda.

**Pokaż długą listę plików w less**

```
> ls | less
```

**Wkrótce:**

```
> ls -l | cut -f3 -d' ' | sort | uniq -c
```

# Polecenia cd.

*ls [opcje]*

- l – długa lista plików
- a – również pliki i katalogi ukryte
- t - sortuj wg daty
- R – pokaż rekurencyjnie
- S – sortuj wg rozmiarów

# ls - zaawansowane

## Pokaż listę plików w długim formacie

```
> ls -al
```

```
razem 532
```

```
drwxr-xr-x 3 gorecki gorecki 4096 2008-10-07 22:56 .
drwxr-xr-x 8 gorecki gorecki 4096 2008-09-30 22:58 ..
drwxr-xr-x 6 gorecki gorecki 4096 2008-10-02 00:39 .svn
-rw-r--r-- 1 gorecki gorecki 29174 2008-10-02 00:32 WI01.odp
-rw-r--r-- 1 gorecki gorecki 458929 2008-10-02 00:33 WI01.pdf
-rw-r--r-- 1 gorecki gorecki 31784 2008-10-08 00:39 WI02.odp
```

prawa dostępu (d/l/r/w/x)

liczba dowiązań

właściciel pliku

grupa

rozmiar

data

nazwa

# Dowiązania

## Pokaż listę plików w długim formacie

```
> ls -al
```

```
razem 532
```

```
drwxr-xr-x 3 gorecki gorecki 4096 2008-10-07 22:56 .
drwxr-xr-x 8 gorecki gorecki 4096 2008-09-30 22:58 ..
drwxr-xr-x 6 gorecki gorecki 4096 2008-10-02 00:39 .svn
-rw-r--r-- 1 gorecki gorecki 29174 2008-10-02 00:32 WI01.odp
```

liczba dowiązań dla katalogu:

- jedno własne (.) + dowiązanie z nadkatalogu (..) + każdy podkatalog

# Dowiązania – polecenie ln

Dowiązania symboliczne (przez ścieżkę) – nie zmieniają liczby dowiązań (twardych) do pliku:

```
> ln -s WI01.odt WI01link.odt
> ls -al WI01*
```

```
lrwxrwxrwx 1 gorecki gorecki      8 ... WI01link.odp -> WI01.odp
-rw-r--r-- 1 gorecki gorecki 29174 ... WI01.odp
```

Dowiązania twarde zmieniają liczbę dowiązań do pliku, dwie nowe nazwy są nierozróżnialne:

```
> ln WI01.odt WI01twardy.odt
> ls -al WI01*
```

```
lrwxrwxrwx 1 gorecki gorecki      8 ... WI01link.odp -> WI01.odp
-rw-r--r-- 2 gorecki gorecki 29174 ... WI01linktwardy.odp
-rw-r--r-- 2 gorecki gorecki 29174 ... WI01.odp
```

# Prawa dostępu

## Prawa:

r – czytanie pliku/zawartości katalogu (np. z ls)

w – pisanie

x – wykonywanie dla pliku/przeszukiwanie dla katalogu (np. cd)

## Użytkownicy:

u – użytkownik

g – grupa

a - wszyscy

o - inni

# Zmiana praw dostępu

*chmod [opcje] pliki*

## Zmień prawa dostępu

**Ustaw prawa czytania**

```
> chmod u-r p.txt
```

```
> cat p.txt
```

```
cat: p.txt: Permission denied
```

**Ustaw prawa wykonywania (skrypty!)**

```
> chmod ugo+x mojskrypt.sh
```

```
> ./mojskrypt.sh
```

```
> ...
```

# Prawo x dla katalogu

```
> mkdir kat # nowy katalog
> ls -l
drwxr-xr-x ..... kat
> echo "Ala ma kota" > kat/p.txt # nowy plik
> chmod u-x kat # drw-r-xr-x
> cd kat
Permission denied # cd zabroniony
> echo kat/* # ale można podejrzec kat
p.txt
> less kat/p.txt # to się nie uda
kat/p1.txt: Permission denied
```

# Prawo x dla katalogu

**Prawo x dla katalogu oznacza:**

- można wejść do niego (cd)
- obejrzeć jego pliki (o ile prawa plików na to pozwolą)

# Prawo r dla katalogu

```
> chmod u+x,u-r kat # d-wxr-xr-x
> ls kat
Permission denied
> cat kat/p.txt
Ala ma kota
> ls -l kat/p.txt
-rw-r--r-- 1 .... kat/p.txt
> cd kat # to OK
> ls
??? Zagadka
```

W katalogu bez tego prawa można obejrzeć tylko te pliki, których znamy nazwę.

# Zmiana właściciela i grupy

*chown [opcje] pliki*

*chgrp [opcje] pliki*

Użytkownicy mogą być członkami grup użytkowników. Właściciel pliku może zmienić grupę pliku na inną grupę o ile do niej należy.

## Zmień grupę

```
> chgrp users p1.txt; ls -l p1.txt  
-rw-r-r-- 1 gorecki users ... p1.txt
```

## Zmień właściciela (zwykle tylko root może to zrobić).

```
> chown root:root p1.txt
```

# Edytory

*vi* – posiada go każdy UNIX, niewygodny w obsłudze

*joe, jed* – dość popularne edytory

*pico* – bardzo prosty edytor

*gedit* – prosty edytor pod X-ami (odpowiednik notepad)

*emacs* – uniwersalne narzędzie o bogatej funkcjonalności;

*Ctrl-h t* – tutorial

*Ctrl-h i* – info

*Ctrl-x f* – otwórz plik

*Ctrl-x s* – zapisz plik

*Ctrl-x c* – zakończ

*Ctrl-s* - inkrementalny search

*Alt-x command* (lub *Esc-x command*) – uruchom

*polecenie command* (np. *regexp-search, compile, etc*)

*xemacs* – wersja w pełni X-owa *emacsa*

# Procesy

Proces to abstrakcja działającego programu.

Procesy mogą komunikować się ze sobą.  
Proces nie może zakłócić działania innego procesu.

Proces może uruchomić inny proces: proces potomny (child process)

# Procesy

Proces potomny, dziedziczy kod, otwarte pliki oraz dane po rodzicu.

Rodzic może czekać (lub nie) na zakończenie procesu dziecka.

Każdy proces, który kończy działanie pozostawia w tablicy procesów systemu kod zakończenia procesu, który powinien być odebrany przez rodzica.

Jeśli rodzic tego nie zrobi, pozostaje proces zombie (jest to tylko miejsce w tablicy procesów, które przechowuje nieodebrany kod).

# Lista procesów

*ps [opcje]*

Pokaż listę procesów

> ps

PID	TTY	TIME	CMD
6430	pts/0	00:00:00	bash
6513	pts/0	00:01:39	soffice.bin
7861	pts/0	00:00:00	gv
7864	pts/0	00:00:03	gs
9045	pts/0	00:00:00	ps

# Lista procesów

```
> ps u
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAN
gorecki	6430	0.0	0.1	5892	3352	pts/0	Ss	10:22	0:00	bash
gorecki	7861	0.0	0.1	4540	2620	pts/0	S	10:38	0:00	gv x.ps
gorecki	7864	0.3	0.5	22948	12036	pts/0	S	10:38	0:04	ps
gorecki	9633	0.0	0.0	2644	1004	pts/0	R+	11:01	0:00	ps u

PID – id procesu

CPU – użycie CPU

MEM – użycie pamięci

VSZ – całkowita pamięć wirtualna

RSS – pamięć rezydentna (nie swap)

TTY – terminal kontrolujący

STAT – kod stanu procesu (R – wykonywany/w kolejce do wykonania;

S – przerywalnie uspiiony; T – zatrzymany przez sygnał lub śledzony; Z – zombie;  
patrz man ps)

START – godzina rozpoczęcia

COMMAND – polecenie z argumentami, które uruchomiło proces