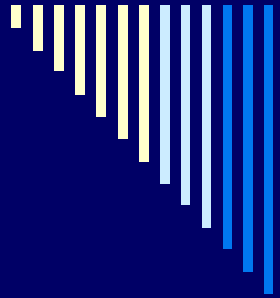


Cvičení 3.

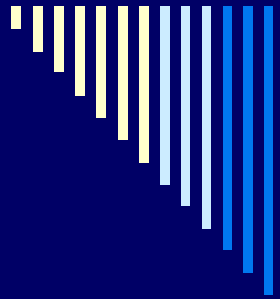
ZOS 2005, L. Pešička





Plán

- procesy, jobs
 - Find
 - Wildcards
 - Příklad – uživatelé
-



Procesy

```
eryx4> ps x
```

PID	TTY	STAT	TIME	COMMAND
15055	pts/6	S	0:00	-tcsh
15256	pts/6	R	0:00	ps x

```
eryx4> top &
```

```
[1] 15260
```

```
eryx4>
```

```
[1] + Suspended (tty output)    top
```



Procesy 2.

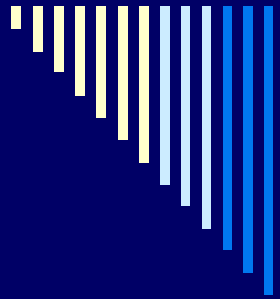
eryx4> ps x

PID	TTY	STAT	TIME	COMMAND
15055	pts/6	S	0:00	-tcsh
15260	pts/6	T	0:00	top .. ma Tcko
15261	pts/6	R	0:00	ps x

PS – man ps

PROCESS STATE CODES

- D uninterruptible sleep (usually IO)
 - R runnable (on run queue)
 - S sleeping
 - T traced or stopped
 - Z a defunct ("zombie") process
-



Procesy 3.

```
eryx4> jobs
```

```
[1] + Suspended (tty output) top
```

```
eryx4> jobs -l
```

```
[1] + 15260 Suspended (tty output) top
```

fg .. proces na popředí, fg %cislo, fg cislo

kill cislo_procesu

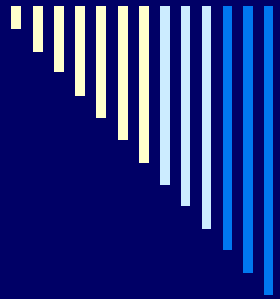
kill -9 cislo_procesu

pstree



Číslování řádků v souboru

- `cat /etc/passwd | nl | head`
 - vypíše prvních 10 řádků
 - `cat /etc/passwd | nl | head -15`
 - vypíše prvních 15 řádků
 - `cat /etc/passwd | nl | tail`
 - vypíše posledních 10 řádků
 - `cat /etc/passwd | nl | tail -3` KVIZ
 - `cat /etc/passwd | nl | tail +3 | more` KVIZ
-



Hledáme s find

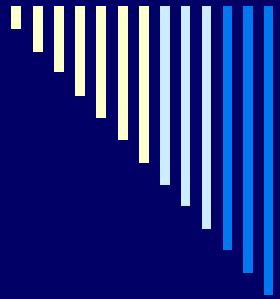
```
find /kupka/sena -name jehla -print
```

- kde hledáme
- co hledáme
- akce
- `find /bin -name date -print`
- `find /dev -type c -print`
- `find /dev -type b -print`



Wildcards * ? a další

- ❑ `cd /bin`
- ❑ `ls m*`
- ❑ `ls mk*`
- ❑ `ls m?`
- ❑ `ls m[a-u]`
- ❑ `ls *a[b-e]*`
- ❑ `ls m[k,o]*`
- ❑ `ls "*"c` .. neinterpretuje obsah uvozovek

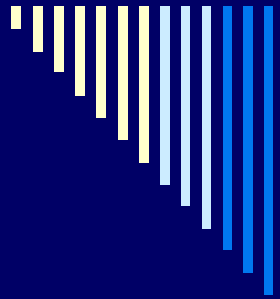


Příklad 1 – uživatelé

□ who

```
pesicka pts/1 Oct 5 23:43 (84.242.95.197)
student6 ttyp0 Oct 6 00:21 (hyperochus.zcu.cz)
fhoudek pts/3 Oct 6 00:09 (koleje-zcu.souepl.cz)
maskova pts/6 Oct 5 20:25 (b1.sab.plz.sloane.cz)
student6 ttyp1 Sep 27 14:32 (hyperochus.zcu.cz)
```

□ chceme seznam uživatelů, abecedně seříděný a bez duplikací

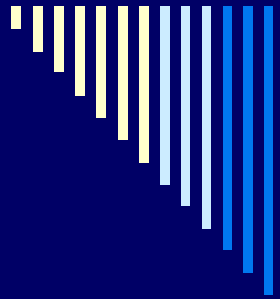


Příklad 1 – uživatelé

□ `who | cut -c1-8`

```
pesicka  
student6  
fhoudek  
maskova  
student6
```

□ **vybere jen znaky 1 az 8**

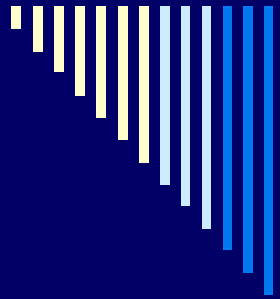


Příklad 1 – uživatelé

□ `who | cut -c1-8 | sort`

```
fhoudek  
maskova  
pesicka  
student6  
student6
```

□ **setřídí podle abecedy**

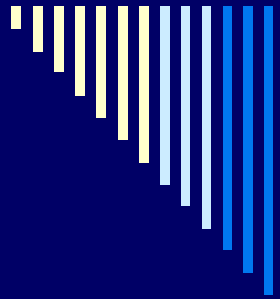


Příklad 1 – uživatelé

□ `who | cut -c1-8 | sort | uniq`

```
fhoudek  
maskova  
pesicka  
student6
```

□ odstraní duplicity seřazených řádků



Příklad 1 – uživatelé

- vytvoříme dávku pro spuštění
- `cd ; cat > vypis_prihlasene`
- `who | cut -c1-8 | sort | uniq`
- stiskneme Ctrl+D (^D)
- `chmod +x vypis_prihlasene`
- vyzkoušime: `./vypis_prihlasene`

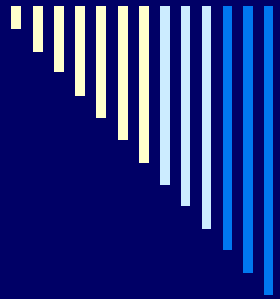


Nahrazování slov

□ **echo** Ahoj NAZDAR BAF buF | tr '[A-Z]' '[a-z]'

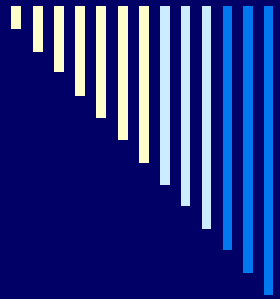
□ vypíše:

□ ahoj nazdar baf buf



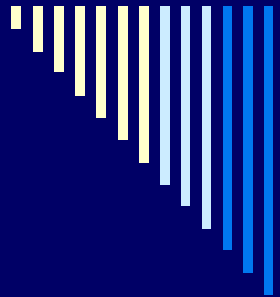
Více adresářů najednou

- `mkdir ahoj/{jedna,dve,tri}`
- adresář `ahoj` už existuje
- vytvoří všechny tři



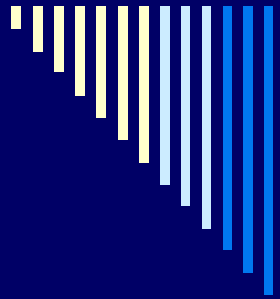
Příklad – četnost slov

- Napřed ve vstupním souboru unifikuji velikost znaků
- `tr '[A-Z]' '[a-z]' < textik.txt`



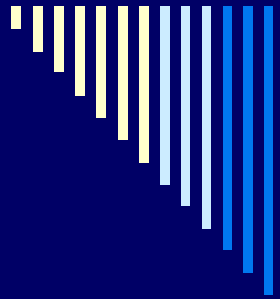
Příklad – četnost slov

- Odstraníme interpunkci, slova oddělená mezerami
- ```
tr '[A-Z]' '[a-z]' < textik.txt |
tr -cd '[A-Za-z0-9_ \012]'
```



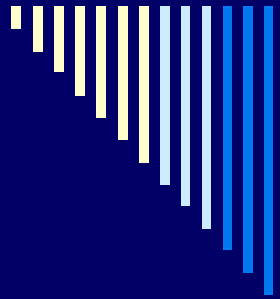
## Příklad – četnost slov

- každé slovo na novém řádku
- `tr '[A-Z]' '[a-z]' < textik.txt | tr -cd '[A-Za-z0-9_ \012]' |`
- `tr -s '[' '\012'`



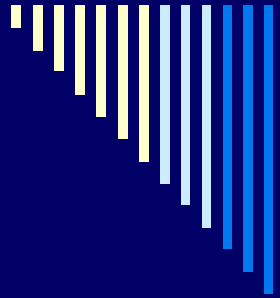
## Příklad – četnost slov

- Výstup seříděný podle slov, odstraněné duplicity
- `tr '[A-Z]' '[a-z]' < textik.txt | tr -cd '[A-Za-z0-9_ \012]' |`
- `> tr -s '[' '\012' | sort | uniq -c | ...`



## Příklad – četnost slov

- seřadit podle četnosti výskytu - finále
- `tr '[A-Z]' '[a-z]' < textik.txt | tr -cd '[A-Za-z0-9_ \012]' | tr -s '[' '\012' | sort | uniq -c | sort -nr`



---

# Dodatky

- Práva číselně, umask
  - Filtr tee
  - Komprese, tar, ...
-



# Nastavení přístupových práv

- `chmod ug+rwx soubor`
- Práva lze zadat i číselně
  - r,w,x .. Tři trojice
  - Binárně 000 až 111 .. 0 až 7 .. osmičková soustava
- Příklady
  - `chmod 777 ahoj.txt`                   - `rwx rwx rwx`
  - `chmod 700 ahoj.txt`                   - `rwx --- ---`
  - `chmod 644 ahoj.txt`                   - `rw- r-- r--`



# Přístupová práva - pokus

- Pokusy vždy v nějakém pracovním adresáři, např. zos04
- touch ahoj.txt ; ls -l; chmod 777 ahoj.txt ; ls -l

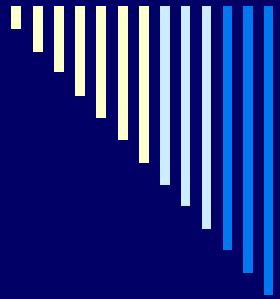
- -rw----- 1 pesicka users 0 Oct 12 10:15 ahoj.txt
- -rwxrwxrwx 1 pesicka users 0 Oct 12 10:15 ahoj.txt



## umask

- Maska přístupových práv při vytváření souborů
- Obsahuje práva, která „vypne“ (doplněk)
- Samotný **umask** – vypíše aktuální nastavení
  - Např. `umask` -> vypíše `77`
  - Tj. `group`, `other` nebudou mít žádná práva při vytvoření nového souboru

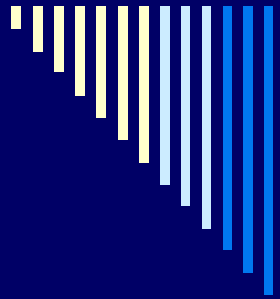




## Umask - příklad

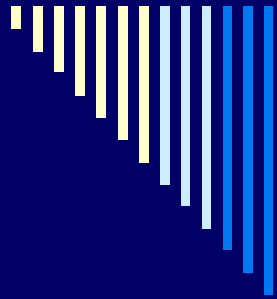
- umask 007 ; touch nazdar1 ; ls -l
- umask 000 ; touch nazdar2 ; ls -l
- umask 077 ; touch nazdar3 ; ls -l

- -rw-rw---- 1 pesicka users 0 Oct 12 10:42 nazdar1
- -rw-rw-rw- 1 pesicka users 0 Oct 12 10:42 nazdar2
- -rw----- 1 pesicka users 0 Oct 12 10:43 nazdar3



## Filtr **tee**

- Kopíruje std. vstup na std. výstup
- A současně zapisuje do uvedeného souboru
- Např. kopírování mezivýsledků
- Analogie – pipe (potrubí),  
tee (odbočka vodovodního řadu, téčko)
  
- `ls -l | tee soubor.txt | grep ahoj`



# Kompresa souborů

## □ gzip

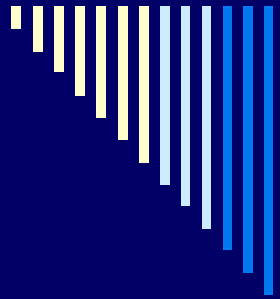
- Komprimuje zadaný soubor
- Původní soubor přestane existovat, je nahrazen komprimovaným

## □ gzip soubor.txt ; ls -l

□ -rw----- 1 pesicka users 59 Oct 12 10:59 soubor.txt.gz

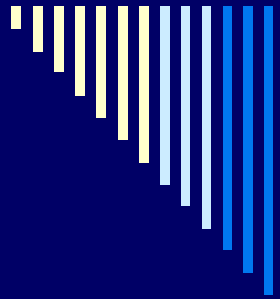
□ Gunzip soubor.txt.gz

□ Gzip -d soubor.txt.gz



# Kompresce souborů – bzip2

- Jiný komprimační algoritmus
- Výsledný soubor menší x více zatěžuje CPU
  
- **bzip2 soubor.txt**
- `-rw----- 1 pesicka users 72 Oct 12 10:59 soubor.txt.bz2`
- **bzip2 -d soubor.txt.bz2**
- Stejně tak lze použít bunzip2



## tar - archivace

- pro archivaci na pásku
- Z několika souborů a adresářů – 1 velký soubor
- Distribuce sw v balíčku – tarball – tar.gz
- Zabalení a následná komprimace
- tar.gz, tar.Z, tar.bz2
- Parametr -z před rozbalení použije gunzip



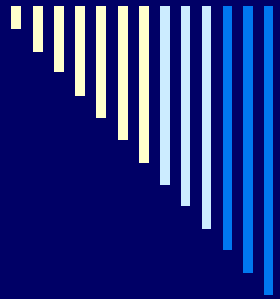
# Rozbalujeme tarball

□ `tar -xvzf soubor.tar.gz`

□ Parametry

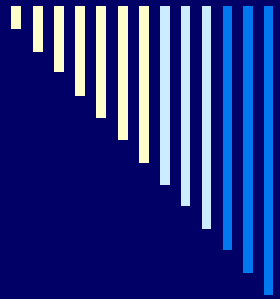
- `-x`      rozbalit (extract)
- `-v`      upovídáný (verbose)
- `-z`      nejprve použije gunzip  
(pro bzip2 by bylo `-j`)
- `-f`      následuje jméno souboru

□ Vybalené soubory ukládá do aktuálního adresáře



# Vytváření archivu

- `tar -cvzf soubor.tar.gz .`
  - Volba `-c` vytváří archiv
  
- Vytvořený archiv můžeme prohlédnout přes `mc`



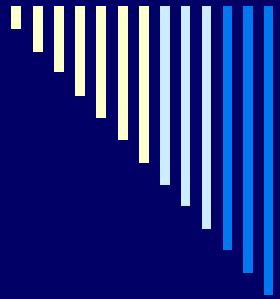
# Zip, unzip

- zip archiv \*

  - Vytvoří archiv.zip

- unzip archiv.zip





# Studium materiálů

- <http://www.kiv.zcu.cz/~simekm/skoleni/index.html>
  - kapitoly 1 až 4
  
- <http://aldebaran.feld.cvut.cz/~xmyslik/www/theidiot.html>
  - povídání o Linuxu – tabulky příkazů
  - Všechny části článku zajímavé čtení