



Přednáška 3

Filtry a některé užitečné příkazy OS Unix.



`tee [-ai] [soubory]`

- Duplikuje standardní vstup, jednu kopii posílá na standardní výstup a další do souboru/souborů.

- a připojení k souboru
- i ignorování všech přerušení

• **Příklady:**

```
finger | tee a.txt b.txt  
spell text.txt | tee -a typos
```

n1 [přepínače] [soubor]

- Očísluje řádky souboru.
- s řetězec' oddělí text od čísla řádky daným řetězcem
- bp' vzor' očísluje pouze řádky obsahující daný vzor

• Příklady:

```
ls -l / | n1
```

```
ls -l / | n1 -s' ) '
```

```
ls -l / | n1 -bp'^-'
```

head [-n] [soubor]

- Vytiskne prvních n řádek ze souborů (implicitně tiskne prvních 10 řádek).

Příklady:

ls -lct / | nl | head -4

`tail [přepínače] [soubor]`

- Vytiskne posledních 10 řádek ze souborů.
 - n vytiskne posledních n řádek
 - +n tiskne od řádky n až do konce
 - f neukončí se na konci souboru (CTRL-C)
 - r vytiskne v obráceném pořadí

• **Příklady:**

```
ls -lct / | nl | tail
```

```
ls -lct / | nl | tail -5
```

```
ls -lct / | nl | tail +5
```

```
tail -f soubor.txt
```

`wc` [přepínače] [soubory]

- Vypíše kolik znaků, slov, řádek obsahují soubory.

`-c` pouze počet znaků

`-w` pouze počet slov

`-l` pouze počet řádek

- **Příklady:**

```
ls -l / | tail +2 | wc -l
```

```
wc -l /etc/passwd
```

cut přepínače [soubory]

- Vypiše specifikované sloupec ze souborů.
 - cseznam vypiše znaky na příslušných pozicích definovaných seznamem (např. 2-10,15,45-)
 - dznak definice oddělovače
 - fseznam vypiše sloupce určené seznamem (oddělovač může být pouze jeden znak)

• Příklady:

```
ls -l | cut -c2-10,15-23,54-
```

```
ypcat passwd | cut -d: -f1,3,5
```

```
who | cut -d' ' -f1
```

paste [přepínače] soubory

- Sloučí odpovídající řádky z jednoho nebo více souborů do jednoho souboru.
-dznaky definice oddělovačů

- **Příklady:**

```
ypcat passwd | cut -d: -f1 > jmeno.txt
ypcat passwd | cut -d: -f3 > uid.txt
ypcat passwd | cut -d: -f6 > adresar.txt
ypcat passwd | cut -d: -f7 > shell.txt
paste -d'+-' uid.txt jmeno.txt shell.txt
```


`split [přepínače] soubor jméno`

- Rozdělí soubor do menších souborů o příslušné velikosti a pojmenuje je: `jménoaa, jménoab, jménoac,...`
 - b *velikost* definice velikosti výstupních souborů v bytech
 - l *velikost* definice velikosti výstupních souborů v počtu řádků
 - a *velikost_přípony*

- **Příklady:**

```
split -b 10k /bin/ls ls
```

```
cat ls?? > ls
```

```
man ls | nl > ls.txt
```

```
split -l 10 -a 5 ls.txt ls
```

```
cat ls????? > ls.txt
```

uniq [přepínače] [soubor]

- Odstraní duplikované sousední řádky.
 - c a vypíše informaci kolikrát se daná řádka vyskytovala
 - d vypíše pouze duplikované řádky jednou (jedinečné nevypíše vůbec)
 - u vypíše pouze jedinečné řádky (duplikované nevypíše vůbec)

• Příklady:

```
who | cut -d" " -f1 | sort > jmena.txt
```

```
uniq -c jmena.txt
```

```
uniq -d jmena.txt
```

```
uniq -u jmena.txt
```

tr [přepínače] řetězec1 řetězec2

- Kopíruje standardní vstup na standardní výstup a provádí náhradu znaků z **řetězec1** na znaky z **řetězec2** nebo maže znaky z **řetězec1**.
- d maže znaky z **řetězec1** z výstupu
- s opakuje se znaky nahradí pouze jedním znakem

• Příklady:

```
cat file.txt | tr 'ABC' 'xyz'
```

```
cat file.txt | tr '[a-z]' '[A-Z]'
```

```
ls -l | tr -s ' ' | cut -d' ' -f2,9-
```

```
ls -l | tr -s ' ' '\012'
```

sort [přepínače] [soubor]

- Setřídí řádky souboru (implicitně v abecedním pořadí).
 - f ignoruje malá/velká písmena
 - n setřídí numericky
 - M porovnává první tři písmena jako měsíc
 - r setřídí v opačném pořadí
 - u identické řádky vypíše pouze jednou
 - znak definuje oddělovač sloupečků
 - kpočáteční-sloupec[typ],[koncový-sloupec[typ]]

definujeme sloupce podle, kterých se bude třídit

Příklady:

```
ls -l / | sort
```

```
ls -l / | sort -k3
```

```
ls -l / | sort -k3,3
```

```
ls -l / | sort -k2,2n
```

```
ls -l / | sort -k2,2nr
```

```
ls -l / | sort -k6,6M
```

```
ls -l / | sort -k6,6M -k7,7n
```

```
ls -l / | sort -k6,6M | sort -k7,7n
```

```
ls -l / | sort -k6,6M -k7,7n -k8.2,8.3n -k8.5,8.6n
```

```
ypcat passwd | sort -t' : ' -k3,3n
```



```
cmp [přepínače] soubor1 soubor2
```

- Porovná dva soubory a vypíše informaci o tom, kde je první rozdílnost.
- s nevypíše nic, pouze vrátí návratový kód

- **Příklady:**

```
$ cmp s1.txt s3.txt
```

```
s1.txt s3.txt differ: char 15, line 2
```

comm [-123] soubor1 soubor2

- Porovná řádky dvou setříděných souborů a vypíše tři sloupce:
 1. sloupec obsahuje řádky obsažené pouze v souboru1
 2. sloupec obsahuje řádky obsažené pouze v souboru2
 3. sloupec obsahuje řádky obsažené v obou souborech

Příklady:

comm s1.txt s3.txt

comm -12 s1.txt s3.txt

diff [přepínače] soubor1 soubor2

- Hledá rozdílné a shodné části dvou souborů.
- Rozdíly vypisuje v různých tvarech, implicitně:
 - XaY za řádek X prvního souboru jsou přidány řádky Y ve druhém souboru
 - XcZ řádky X prvního souboru jsou nahrazeny řádky Y druhého souboru
 - XdY řádky X z prvního souboru chybí ve druhém souboru za řádkem Y
- Přepínačem -u docílíme jiný tvar výstupu (+, -).
 - + označuje řádky, které jsou ve druhém souboru navíc
 - označuje řádky, které ve druhém souboru chybí
- Přepínačem -c docílíme opět jiný tvar výstupu (+,-,!).

patch [přepínače] [soubor]

- Podle výstupu z **diff** a orig. souboru vytvoří soubor nový.
 - b vytvoří zálohu původního souboru (soubor.orig)
 - i patchfile rozdílové informace čte ze souboru patchfile
 - R návrat k původní verzi

Příklady:

```
diff -u f1 f2 > patch.diff
```

```
patch -b -i patch.diff f1
```

```
patch -b -R -i patch.diff f1
```

find adresář [výraz]

- V daném adresáři hledá soubory podle zadaného výrazu.
- Výraz se vyhodnocuje zleva doprava a vyhodnocení končí v okamžiku, když je jasné, že výraz není splněný.
- Implicitně jsou jednotlivé části výrazu spojeny logickým součinem.
- Jednotlivé části můžeme spojit logickým součtem **-o**.
- Jednotlivé části výrazu můžeme seskupovat pomocí závorek **(a \)**.
- Také můžeme využít operátor negace **!**.

- Ve výrazu můžeme použít následující příkazy:
 - print výpis souborů, které splňují výraz
 - ok příkaz { } \; interaktivní provedení příkazu pro soubory, které splňují výraz
 - exec příkaz { } \; neinteraktivní provedení příkazu pro soubory, které splňují výraz

- Ve výrazu můžeme použít následující podmínky pro:

-type [d,f,l,b,c]	typ souboru
-inum n	číslo i-uzlu
-name 'vzor'	jméno (lze použít znaky *,?,[,], ,...)
-perm nnn	přístupové práva
-mtime [n -n +n]	čas modifikace
-atime [n -n +n]	čas přístupu
-ctime [n -n +n]	čas modifikace

Příklady:

```
find $HOME
```

```
find $HOME -type f
```

```
find /bin -type f -name at
```

```
find /usr/bin -type f -name `*grep`
```

```
find . -atime -1
```

```
find . -name `[A-Z]*`
```

```
find . \! -name `[A-Z]*`
```

Příklady:

```
find . -name 'f*' -print
find . -name 's*' -print
find . \( -name 'f*' -o -name 's*' \) -print
find . \! \( -name 'f*' -o -name 's*' \) -print

find . -print -name 'f*' -print
find . -name core -ok rm { } \;
find . -name core -exec rm { } \;
```