# C2110
# *Operační systém UNIX a základy programování*

## 10. lekce

Petr Kulhánek

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

# Obsah

- **Spuštění příkazů III**
  **proměnná PATH**
- **Hybridní skripty**
  **přesměrování v rámci skriptu**
- **Nové příkazy**
  **type, hash, tr, memcoder, mplayer**

# Bash

# Spouštění příkazů a aplikací, III

Aby mohl shell zadaný příkaz spustit, potřebuje **znát úplnou cestu** k souboru, který obsahuje binární program nebo skript.

1. Cesta k příkazu se nejdříve hledá v tabulce s již použitými příkazy:

```
$ hash
hits    command
1    /bin/rm
3    /bin/ls
```

Tabulku lze smazat příkazem:
`$ hash -r`

2. Pokud není příkaz nalezen, hledá se v adresářích uvedených v systémové proměnné **PATH**

**$ echo $PATH** — pořadí prohledávání

```
.../usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Adresáře se oddělují znakem **:** (dvojtečka)

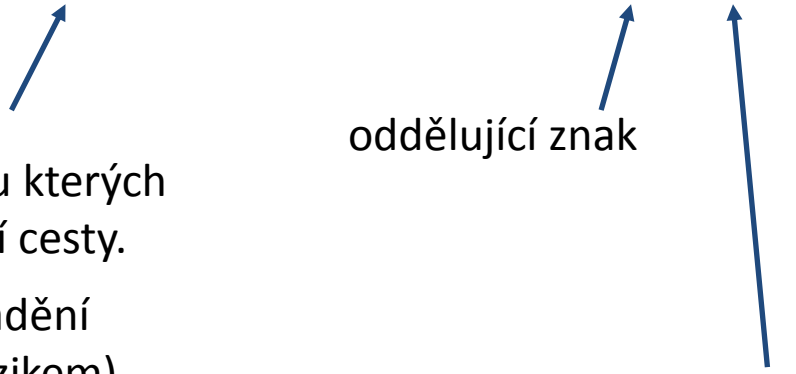
3. Cestu k příkazu, pokud existuje, lze zjistit příkazem **type**

```
$ type ls
/bin/ls
```

# Úprava proměnné PATH

**Manuální změna proměnné PATH**

```
$ export PATH=/moje/cesta/k/mym/prikazum:$PATH
```



Cesta k adresáři obsahující příkazy, u kterých chci, aby byly přístupné bez uvádění cesty.

**Cesta se vždy uvádí absolutně!** (uvádění relativních cest je bezpečnostním rizikem)

oddělující znak

Původní hodnota proměnné **PATH**
(nutné pro nalezení systémových příkazů)

**Automatizovaná změna proměnné PATH**

Automatizovanou změnu proměnné PATH (a případně jiných systémových proměnných) provádí příkaz **module.**

```
$ module add vmd
```

# Příkaz tr

Příkaz **tr** slouží k transformaci nebo mazání znaků ze standardního vstupu. Výsledek je zasílán do standardního výstupu.

**Příklady:**

```
$ cat soubor.txt | tr --delete "qwe"
```

z obsahu souboru **soubor.txt** odstraní znaky "q", "w" a "e"

```
$ cat soubor.txt | tr --delete "[:space:]"
```

z obsahu souboru **soubor.txt** odstraní všechny bílé znaky

```
$ echo $PATH | tr ":" "\n"
```

v textu zaslaného příkazem echo budou nahrazeny znaky ":" znakem nového řádku "\n"

# Cvičení

1. Vypište hodnotu proměnné PATH.
2. Vypište adresáře obsažené v proměnné PATH, každý na jeden řádek.
3. V kterém adresáři se vyskytuje program **kwrite**?
4. Jaké je obsah tabulky použitých příkazů?
5. Jakým způsobem změní hodnotu proměnné **PATH** příkaz **module add vmd**?
6. V jakém adresáři se vyskytuje příkaz **vmd**?

# MPlayer

http://www.mplayerhq.hu

# mplayer

**mplayer** slouží k přehrávání videa. Stručný popis ovládání lze získat spuštěním příkazu bez žádného argumentu.

**Příklad:**

```
$ mplayer movie.avi
                    přehraje video movie.avi
```

**Zajímavé volby:**

| | |
|---|---|
| -loop N | přehraje video N-krát |
| -fs | video přehraje v celoobrazovkovém režimu |

# mencoder

**mencoder** slouží ke kódování videa. Lze jej využít pro konverzi jednoho formátu do druhého, změně kodeku, nebo sestavení videa ze série obrázků.

**Sestavení videa z obrázků:**

```
$ mencoder "mf://*.png" –mf fps=25 -ovc lavc -o output.avi
```

**Vstupní data.** Použije všechny obrázky s příponou png. Obrázky musí mít vhodné jméno, které, pokud je použito pro setřídění, poskytne správnou sekvenci.

**Výstupní encoder.**

**Název vytvořeného videa.**

**Počet snímků za sekundu** (FPS – frames per second).

Přehled: http://mariovalle.name/mencoder/mencoder.html

# Alternativy

**http://ffmpeg.org/**

FFmpeg is a complete, cross-platform solution to record, convert and stream audio and video. It includes libavcodec - the leading audio/video codec library.

**http://gstreamer.freedesktop.org/**

GStreamer is a library for constructing graphs of media-handling components. The applications it supports range from simple Ogg/Vorbis playback, audio/video streaming to complex audio (mixing) and video (non-linear editing) processing.

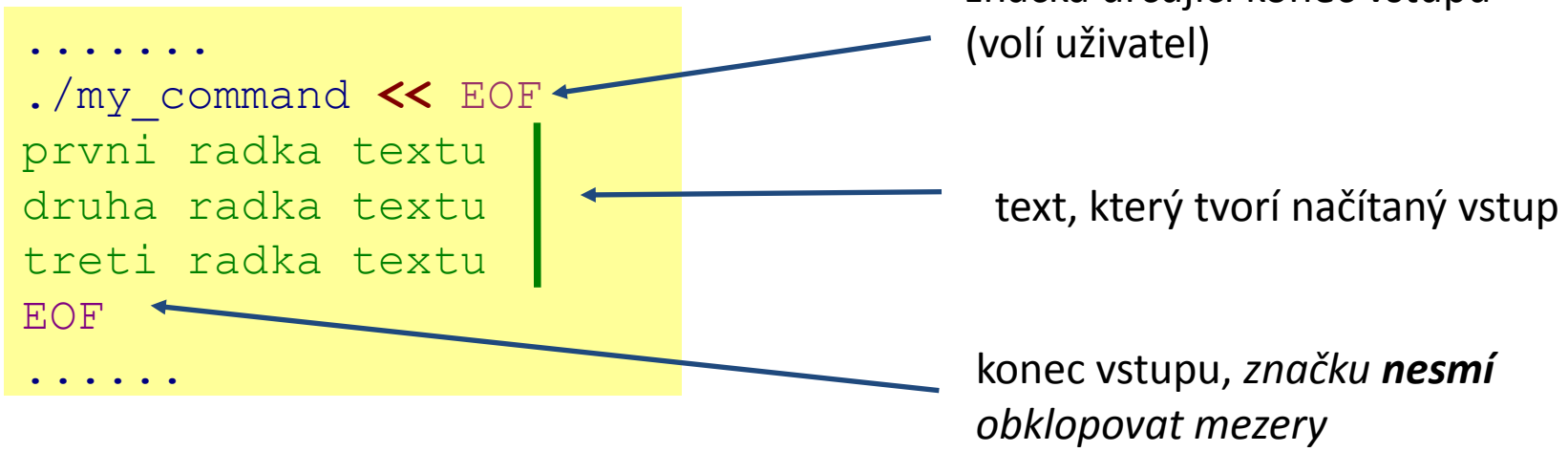
# Cvičení

1. V adresáři **/home/kulhanek/Data/Video** jsou dva soubory s příponou **avi**. Oba soubory si překpírujte do adresáře **moje_video**, který vytvoříte ve vašem domovském adresáři.
2. Obě videa přehrajte v programu **mplayer**. Naučte se základní ovládání programu: pozastavení videa, přesouvání ve videu, přepnutí do celoobrazovkého režimu.
3. V adresáři **/home/kulhanek/Data/MovieImages** jsou obrázky ve formátu png. Vytvořte si adresář **/scratch/vas_login/mimages** , do kterého obrázky překopírujte.
4. Jaké rozměry (šířku, výšku a bitovou hloubku) má obrázek **e_0010.png** ?
5. Z obrázků sestavte dvě videa o FPS=10 a FPS=50.
6. Vytvořená videa přehrajte.

# Hybridní skripty

# Přesměrování v rámci skriptu

**Přesměrování standardního vstupu** programu my_command ze souboru skriptu.




Tento způsob přesměrování je obzvláště výhodné používat ve skriptech, nicméně funguje i v příkazové řádce. Výhodou je expanze proměnných v načítaném textu.

# Ukázky

```
#!/bin/bash

for((I=1;$I<=10;I++)); do
      NAME=`printf "%02d.txt" $I`
      cat << EOF > $NAME
Toto je soubor cislo: $I
EOF
done
```

vyznačený text je poslán do **standardního vstupu** příkazu cat
před odesláním textu jsou **expandovány** proměnné a příkazy uvozené `prikaz`
příkaz cat jej pak uloží do souboru $NAME

```
#!/bin/bash

gnuplot << EOF
plot sin(x)
EOF
```

Uvedeným způsobem lze programově vytvářet skripty pro gnuplot.

# Cvičení

1. Vytvořte skript, který vytvoří deset souborů. Jméno souboru bude ve formátu XX.txt, kde XX je číslo souboru. Pokud je číslo souboru menší než deset, tak jako první cifru v názvu použijte znak 0. Každý soubor bude obsahovat následující text (X je číslo souboru):

```
Automaticky vytvoreny textovy soubor
Cislo souboru je:  X
```

2. Napište skript(y), který vytvoří sérii obrázků zobrazující vlnění (funkce sin, nebo cos v 2D nebo 3D, dle vašeho uvážení). Z obrázků sestavte video pomocí příkazu **mencoder**. Video přehrajte pomocí příkazu **mplayer**.