C2110 *Operační systém UNIX a základy programování*

7. lekce

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

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

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

CZ.1.07/2.2.00/15.0233

Obsah

- **Skripty**
  - Skriptování v Bashi
  - Cyklus pomocí for a for in
- **ImageMagic**
  - Konverze obrázků
  - Příkazy convert, display
- **Příkazy pro práci se jmény a cestami**
  - dirname, basename

Skripty

- Jak psát skripty
- Cyklus pomocí for

Skript v Bashi

```
#!/bin/bash
# toto je komentar
echo 'Toto je skript v interpretu Bash!'
echo "Obsah adresare `pwd` je:"
ls     # vypise obsah adresare
A=6    # nastaveni hodnoty promenne A
echo "Hodnota promenne A je $A"
echo "jeden prikaz"; echo "druhy prikaz"
./mujprikaz prvni_argument druhy_argument \
            treti_argument
```

ihned následuje nový řádek

- prázdné řádky se ignorují
- text uvozený znakem **#** se ignoruje (používá se ke komentovaní funkčnosti skriptu)
- na jeden řádek lze uvést více příkazů, příkazy se oddělují středníkem **;**
- jeden příkaz lze napsat na více řádků pomocí zpětného lomítka ****

Spouštění skriptů v Bashi

1) Nepřímé spouštění

Spouštíme interpreter jazyka a jako argument uvádíme jméno skriptu.

```
$ bash muj_skript_v_bashi
```

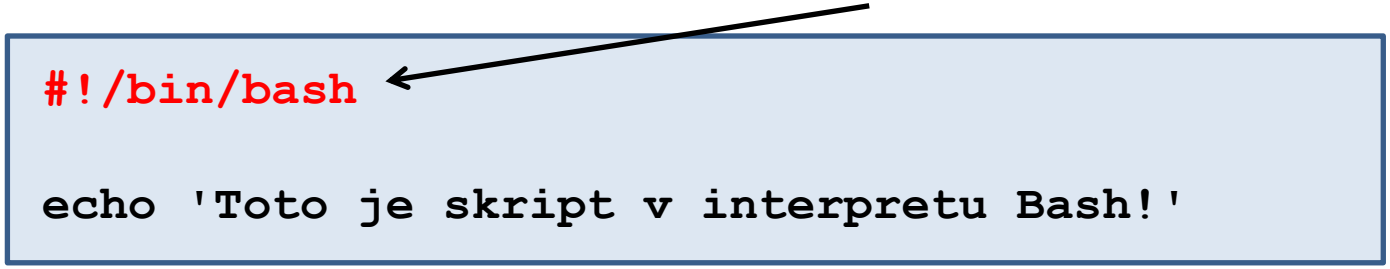
Skripty **nemusí** mít nastaven příznak x (executable).

2) Přímé spouštění

Spouštíme přímo skript (shell automaticky spustí interpreter).

```
$ chmod u+x muj_skript_v_bashi
$./muj_skript_v_bashi
```

Skripty **musí** mít nastaven příznak **x** (executable) a **interpreter** (součást skriptu).



Cyklus pomocí for

Cyklus (smyčka) je řídicí struktura, která opakovaně provádí posloupnost příkazů. Opakování i ukončení cyklu je řízeno podmínkou.

provede se před spuštěním cyklu (inicializace počítadla)

pokud je podmínka splněna, vykonají se příkazy prikaz1 a další

```
for((inicializace;podminka;zmena))
      do
            prikaz1
            ...
      done
```

aktualizace počítadla po vykonání příkazů

Kompaktní zápis:

```
for((inicializace;podminka;zmena)); do
      prikaz1
      ...
done
```

Cyklus pomocí for, použití

Vypíše čísla 1 až 10

```
for((I=1;I <= 10;I++)); do
      echo $I
done
```

Vypíše čísla 10 až 1

```
for((I=10;I >= 1;I--)); do
      echo $I
done
```

Proměnná `I` má roli **počítadla**.

Inicializace proměnné se řídí pravidly pro nastavovaní proměnných v Bashi.

Změna:
Pokud lze proměnnou interpretovat jako celé číslo, lze použít následující aritmetické operátory:

- `++` hodnotu proměnné zvýší o jedničku
- `--` hodnotu proměnné sníží o jedničku

další ....

Podmínka:
Pokud lze proměnnou interpretovat jako celé číslo, lze použít následující porovnávací operátory:

`!=`	nerovná se
`==`	rovná se
`<`	menší
`<=`	menší nebo rovno
`>`	větší
`>=`	větší nebo rovno

Cyklus pomocí for, změna počítadla

Pokud lze proměnnou interpretovat jako celé číslo, lze použít následující aritmetické operátory:

`++` hodnotu proměnné zvýší o jedničku

```
A++
```

`--` hodnotu proměnné sníží o jedničku

```
A--
```

`+` sečte dvě hodnoty

```
A=5 + 6
A=A + 1
```

`-` odečte dvě hodnoty

```
A=5 - 6
A=A - 1
```

`*` vynásobí dvě hodnoty

```
A=5 * 6
A=A * 1
```

`/` vydělí dvě hodnoty (celočíselné dělení)

```
A=5 / 6
A=A / 1
```

```
A=A+3
```

`+=` k proměnné přičte hodnotu

```
A+=3
A+=B
```

`-=` od proměnné odečte hodnotu

```
A-=3
A-=B
```

`*=` proměnnou vynásobí hodnotou

```
A*=3
A*=B
```

`/=` proměnnou podělí hodnotou

```
A/=3
A/=B
```

Cyklus a proměnné

povolené zápisy

Iniciace počítadla:

```
A=1
A = 1
```

Podmínka:

```
A <= 10
$A <= 10
```

Změna počítadla:

```
A=$A + 1
A = $A + 1
A=A + 1
A = A + 1
```

Je možné použít mezery mezi znakem =, jménem proměnné a její hodnotou.

Je možné použít operátor **$** pro přístup k hodnotě proměnné, ale není to zapotřebí.

benevolentnější pravidla platí pouze ve specifikaci cyklu

```
for((I=1;I <= 10;I++)); do
        echo $I
done
```

zde již musí být operátor pro přístup k hodnotě proměnné

Cvičení

1. Napište skript, který vypíše cestu k aktuálnímu adresáři a jeho obsah.
2. Vypište deset znaků **A**, každý na jeden řádek.
3. Vypište deset znaků **A** vedle sebe na jeden řádek (**echo –n** a manuálové stránky).
4. Napište skript, který vypíše sudá čísla od **2** do **100**.
5. Napište skript, který vypíše mocniny 2^n pro **n** od **0** do **32**.

Doporučení:

Každou úlohu řešte v samostatném adresáři. Adresáře si očíslujte, např.:
uloha01
uloha02
atd.

Vnořování cyklů

Řídící skupiny cyklů lze do sebe libovolně vnořovat.

```
for((I=1;I <= 10;I++)); do
      for((J=1;J <= 10;J++)); do
            echo "$I $J"
      done
done
```

vnější cyklus

vnitřní cyklus

```
for((I=1;I <= 10;I++)); do
      for((J=1;J <= I;J++)); do
            echo "$I $J"
      done
done
```

počítadlo vnějšího cyklu může ovlivňovat chování vnitřního cyklu

Cvičení

1. Prakticky porovnejte funkci vnořených cyklů uvedených na předchozí straně.
2. Napište skript, který vypíše desetkrát deset znaků (dle vaší volby) na řádek.
3. Napište skript, který na první řádek vypíše jeden znak, na druhý řádek dva znaky, atd. a to celkem provede desetkrát.
4. Upravte první skript tak, aby vypsal patnáctkrát šest znaků na řádek.
5. Uveďte důvody proč je vhodné pro řídící hodnoty používat proměnné.

Řídící hodnoty je vhodné předávat pomocí proměnných:

```
for((I=1;I <= 10;I++)); do
        echo $I
done
```

méně vhodné

```
POCET=10
for((I=1;I <= POCET;I++));
do
        echo $I
done
```

Image Magic

- **Konverze obrázků**

http://www.imagemagick.org

(dokumentace, tutoriály, zdrojové kódy)

Příkazy

Přehled:

animate, compare, composite, conjure, **convert**, **display**, identify, import, mogrify, montage, stream

Detailní popis je dostupný v manuálových stránkách příkazů nebo na webové stránce Image Magic.

Nejdůležitější příkazy:

display	zobrazí obrázek nebo sekvenci obrázků na obrazovce
convert	provede konverzi mezi různými formáty včetně různých typů operací jako je změna velikosti, ořezání, rozostření, apod.

Příklady:

```
$ convert input.eps output.png
```

zkonvertuje obrázek ve formátu postscript do formátu PNG

Vysoká kvalita pro publikační účely:

```
$ convert -density 300x300 input.eps -units PixelsPerInch \
          -density 300 -background white -flatten output.png
```

Skripty

- **Cyklus pomocí for in**

Cyklus pomocí for ... in ...

Příkazy v bloku **do/done** (**prikaz1, …**) se vykonají pro každý prvek v seznamu **LIST**. V daném běhu cyklu obsahuje proměnná **VAR** aktuální prvek ze seznamu **LIST**.

```
for VAR in LIST
      do
            prikaz1 $VAR
            ...
      done
```

Kompaktní zápis:

```
for VAR in LIST; do
      prikaz1 $VAR
      ...
done
```

Cyklus pomocí for ... in ..., seznamy

```
for A in a b c; do
      echo $A
done
```

Cyklus proběhne třikrát, během toho postupně vytiskne znaky **a**, **b**, **c**.

Seznamy položek je vhodné vytvářet programově (pomocí příkazů uvedených v obrácených apostrofech).

```
for A in `ls *.eps`; do
      ./process_file $A
done
```

Příkaz **process_file** se vykoná pro každý soubor s příponou **.eps**, který se nachází v aktuálním adresáři.

Příkazy pro práci se jmény a cestami

- dirname
- basename

Příkazy pro práci s jmény a cestami

dirname	vyextrahuje jméno adresáře z úplného jména souboru
basename	vyextrahuje jméno souboru z úplného jména souboru

Příklady:

```
$ basename /home/kulhanek/pokus.txt
pokus.txt
$ basename pokus.txt
pokus.txt
$ basename /home/kulhanek/pokus.txt .txt
pokus
$ dirname /home/kulhanek/pokus.txt
/home/kulhanek
$ dirname pokus.txt
.
```

Příkazy **dirname** and **basename** zpracovávají zadané informace bez ohledu na to, zda-li daný soubor či adresář existuje.

Cvičení

1. Vytvořte adresář **obrazky**
2. Do adresáře **obrazky** nakopírujte soubory z adresáře **/home/kulhanek/Data/Snapshots/** , které mají zakončení **.eps**
3. Napište skript, který na obrazovku vypíše názvy souborů, které obsahuje adresář **obrazky** v následujícím formátu:

```
Adresar obrazky obsahuje soubor:  soubor1.eps
Adresar obrazky obsahuje soubor:  soubor2.eps
```

4. Napište skript, který převede soubory ve formátu **eps** v adresáři **obrazky** do formátu **png**.
5. Ověřte příkazem **display**, že konverze proběhla v pořádku.