

C2110

Operační systém UNIX a základy programování

7. lekce

Petr Kulhánek

kulhanek@chemi.muni.cz

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

Obsah

- **Skriptování v Bashi**
skripty, cyklus pomocí for a for in, speciální proměnné
- **ImageMagic**
konverze obrázků
- **Nové příkazy**
convert, display, dirname, basename

Skripty

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
```

↓
pořadí vykonávání příkazů

↙
ihned následuje
nový řádek

- prázdné řádky se ignorují
- text uvozený znakem # se ignoruje (používá se ke komentování 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).

```
#!/bin/bash ←
```

```
echo 'Toto je skript v interpretu Bash!'
```

Cyklus pomoci 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 pomoci for, použití

Vypíše čísla 1 až 10

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

Proměnná **I** má roli **počítadla**.

Inicializace proměnné se řídí pravidly pro nastavování 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

Vypíše čísla 10 až 1

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

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 pomoci 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

Cvičení

1. Napište skript, který vypíše cestu k aktuálnímu adresáři a jeho obsah.
2. Vypište deset znaků **A**, každé 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 očísľujte, např.:

uloha01

uloha02

atd.

Image Magic

<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 pomoci 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.

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

Kompaktní zápis:

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

Cyklus pomoci 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.

Nové příkazy

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.

Speciální proměnné

Argumenty skriptu

```
$ bash muj_skript_v_bashi arg1 arg2 arg3
```

```
$ ./muj_skript_v_bashi arg1 arg2 arg3
```

```
#!/bin/bash
```

```
echo "Pocet zadanych argumentu: $#"
```

```
echo "Prvni argument je: $1"
```

```
echo "Druhy argument je: $2"
```

```
echo "Vsechny zadane argumenty jsou: $*"
```

```
echo "Nazev spusteneho skriptu: $0"
```

3

arg1

arg2

arg1 arg2 arg3

./muj_skript_v_bashi

Použití a význam argumentů si určuje autor skriptu.

Přehled

Argumenty skriptu:

- # počet argumentů, se kterými byl skript spuštěn
- 0 název spuštěného skriptu
- 1 ... 9 hodnoty argumentů 1 až 9, se kterými byl skript spuštěn
- * všechny argumenty, se kterými byl skript spuštěn

Procesy:

- ? návratová hodnota posledně vykonaného příkazu (procesu)
- \$ číslo procesu (PID)

Pokročilá práce s argumenty:

Pokud potřebujeme předat více jak devět argumentů, je nutné použít příkaz **shift**. Příkaz odstraní první argument ze seznamu argumentů.

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

Vypíše postupně zadané argumenty skriptu.

Cvičení

1. Napište skript, který vypíše počet argumentů, které jste zadali při jeho spuštění.
2. Napište skript, který vypíše znaky **A** vedle sebe. Počet vytisknutých znaků uživatel zadá jako první argument skriptu.
3. Napište skript, který vypíše **PID** procesu, ve kterém je skript interpretován. Za výpis vložte příkaz **sleep**, kterým vykonávání skriptu pozastavíte na 5 minut. Použijte příkaz **kill** v jiném terminálu k předčasnému ukončení vašeho skriptu.