C2110

Operační systém UNIX a základy programování

5. lekce

Petr Kulhánek

kulhanek@chemi.muni.cz

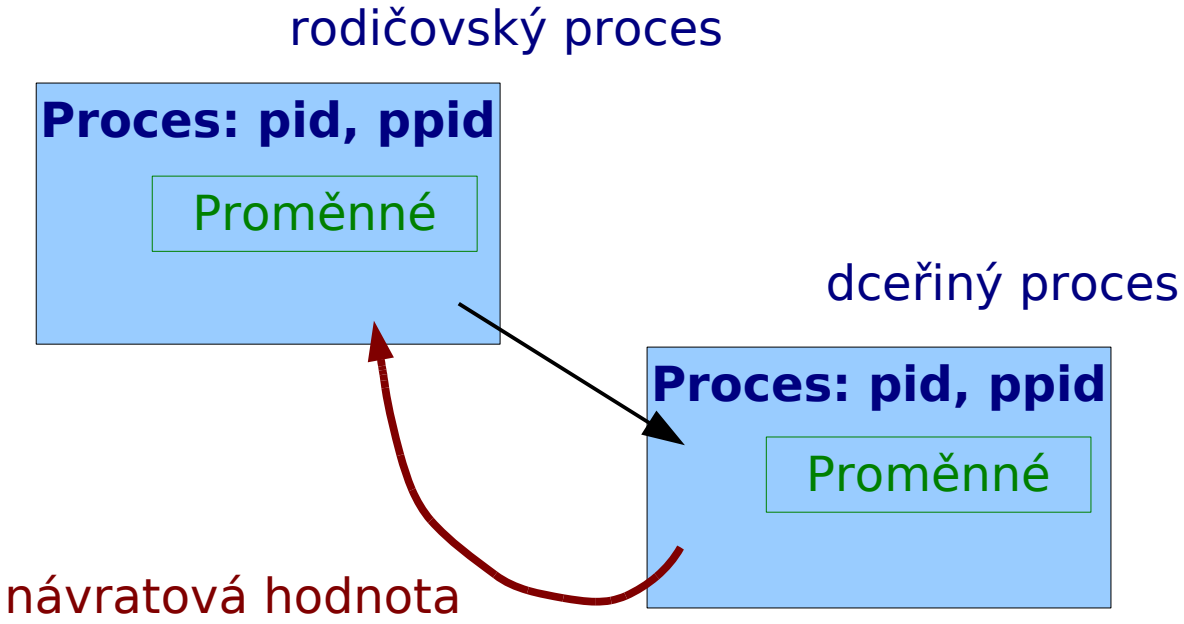
Národní centrum pro výzkum biomolekul, Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

Skriptování v jazyce bash

- spouštění příkazů – návratová hodnota
- cyklus pomocí *while*, příkaz test, aritmetické operace
- cyklus pomocí *for*
- cyklus pomocí *for in*
- podmínky
- příkazy *expr*, *read*, *printf*
- přesměrování a roury
- speciální proměnné
- předávaní argumentů

Spouštění příkazů

Končící proces může rodičovskému procesu sdělit informaci o svém průběhu pomocí **návratové hodnoty**. Návratová hodnota je celé číslo nabývající hodnot 0-255.



Návratová hodnota:

0 = vše proběhlo úspešně

> 0 = došlo k chybě, vrácená hodnota pak zpravidla identifikuje chybu

Návratová hodnota

Návratovou hodnotu posledně provedeného příkazu lze zjistit pomocí proměnné **?**.

```
$ mkdir test
$ echo $?
0

$ mkdir test
mkdir: cannot create directory `test1': File exists
$ echo $?
1
```

Cyklus pomocí while

Syntaxe:

```
while prikaz1
    do
        prikaz2
        ...
    done
```

Příkaz **prikaz1** a všechny příkazy v bloku **do/done** (**prikaz2, ...**) se vykonávají ve smyčce, **dokud** (while) **příkaz1** končí návratovou hodnotou 0.

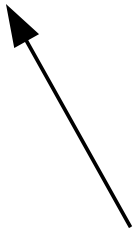
Kompaktní zápis:

```
while prikaz1; do
    prikaz2
    ...
done
```

Cyklus pomocí while ...

```
inicializace_pocitadla
while test_pocitadla; do
    prikaz2
    ...
    zvyseni_hodnoty_pocitadla
done
```

```
I=1
while test $I -le 10; do
    prikaz2
    ...
    I=$(($I+1))
done
```



Vykoná **prikaz2** desetkrát.

Příkaz test

Příkaz **test** slouží k porovnávání hodnot a testovaní typů souborů a adresářů.

Porovnávání celých čísel:

test cislo1 operand cislo2

Operand:

-eq	rovná se
-ne	nerovná se
-lt	menší než
-le	nenší než nebo rovno
-gt	větší než
-ge	větší než nebo rovno

V případě, že je podmínka splněna, je návratová hodnota 0.

Další informace: man bash, man test

Aritmetické operace

Aritmetické operace s celými čísly lze vykonat v bloku **((...))**.

Možné zápisy:

```
I=$(( $I + 1 ))
I=$(( I + 1 ))
(( I = I + 1 ))
(( I++ ))
```

Zvýšení hodnoty proměnné **I** o jedna.

Operátory:

`+`	sčítání
`-`	odčítání
`*`	násobení
`/`	dělení
`%`	zbytek po dělení
`++`	inkrementace (zvýšení hodnoty o 1)
`--`	dekrementace (snížení hodnoty o 1)

Další informace: man bash

Cyklus pomocí for

Syntaxe:

```
for((inicializace;test;zmena))
    do
        prikaz2
        ...
    done
```

```
inicializace
while test; do
    prikaz2
    ...
    zmena
done
```

Kompaktní zápis:

```
for((inicializace;test;zmena)); do
    prikaz2
    ...
done
```

Příklad:

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

Cyklus pomocí for in

Syntaxe:

```
for VAR in LIST
    do
        prikaz2 $VAR
        ...
    done
```

Příkazy v bloku **do/done** (**prikaz2, ...**) se vykonají pro každý prvek v seznamu **LIST**. V daném běhu cyklu obsahuje proměnná **VAR** aktuální hodnotu prvku.

Kompaktní zápis:

```
for VAR in LIST; do
    prikaz2 $VAR
    ...
done
```

Příklad:

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

Podmínky

Syntaxe:

```
if prikaz1
   then
      prikaz2
      ...
   fi
```

```
if prikaz1
   then
      prikaz2
      ...
   else
      prikaz3
      ...
   fi
```

Kompaktní zápis:

```
if prikaz1; then
   prikaz2
   ...
fi
```

```
if prikaz1; then
   prikaz2
   ...
else
   prikaz3
   ...
fi
```

Pokud **prikaz1** skončí s návratovou hodnotou **0**, vykoná se **prikaz2**. V opačném případě se vykoná **prikaz3**.

Příkaz expr

Příkaz **expr** vyhodnocuje matematické výrazy, výsledky se tisknou do standarního výstupu.

Příklady:

```
$ expr 1 + 2
3

$ expr 2 \* 3
6
```

```
I=1
while test $I -le 10; do
    prikaz2
    ...
    I=`expr $I + 1`
done
```

`prikaz` spustí prikaz a text, který jde do standarního výstupu se umístí do místa uvozovek

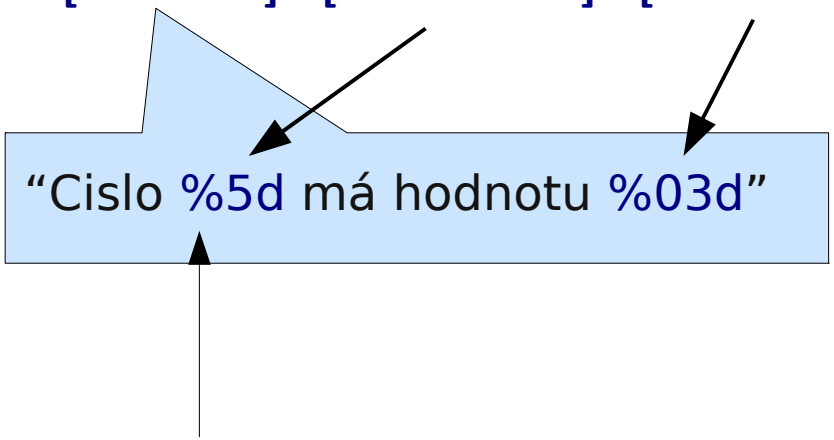
Další informace: man expr

Příkaz printf

Příkaz **printf** slouží k vypisování formátovaných textů.

Syntaxe:

printf `[format] [hodnota1] [hodnota2] ...`

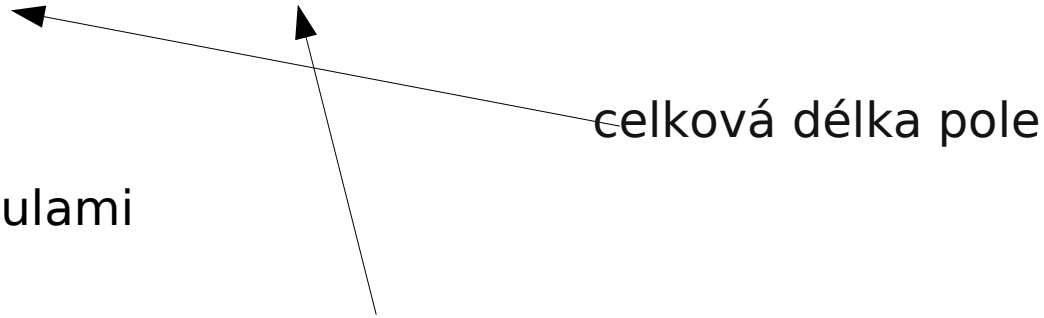


Další informace: man bash, man printf

Příkaz printf ...

Formát:

%[priznak][delka][.presnost]typ



Příznak:

-	zarovnat doleva
0	prázdné místo zaplnit nulami
+	vždy uvést znaménko

Typ:

d	celé číslo
s	řetezec (text)
f	reálné číslo

Speciální znaky:

\n	- konec řádku
\r	- vrať se na začátek řádku
%%	- znak %

Další informace: man bash, man printf

Příkaz read

Příkaz **read** slouží k čtení textu ze standardního vstupu a jeho uložení do proměnných. Příkaz načte vždy celý řádek.

Syntaxe:

```
read A          # celý řádek se uloží do proměnné A
read A B        # první slovo se uloží do proměnné A
                # zbytek řádku do proměnné B
```

Příklad:

```
echo "Zadej dve cisla oddelena mezerou:"
read A B
echo "Soucet je : $(($A + $B))"
```

Pozor: nepoužívejte přikaz read ve spojení s rourami

```
echo "text" | read A
echo $A
```

Nebude obsahovat hodnotu "text".

Další informace: man bash

Přesměrování a roury

Čtení souboru po řádcích:

```
cat soubor.txt | while read A; do
    prikaz2
    ...
done
```

roura

```
while read A; do
    prikaz2
    ...
done < soubor.txt
```

přesměrování

Přesměrování do souboru:

```
for((I=1;I <= 10;I++)); do
    echo $I
done > soubor.txt
```

Výstup všech příkazů v cyklu je přesměrován do **soubor.txt**.

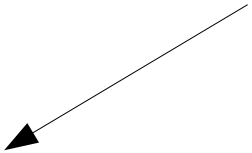
Speciální proměnné

Proměnné:

?	návratová hodnota posledního příkazu
$	číslo procesu (PID)
#	počet argumentů
0	název spuštěného skritpu
1 … 9	hodnoty argumentů 1 až 9
*****	všechny argumenty

Argumenty skriptu

$ ./muj_skript 10 druha 11



```
$0      ./muj_skript
$#      3
$1      10
$2      druha
$3      11
$*      10 druha 11
```

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

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

Cvičení

Cvičení I

A) Vypište deset písmen A vedle sebe na jeden řádek.

B) Vypište deset písmen A, každé na jeden řádek.

C) Upravte řešení A tak, že počet písmen zadá uživatel z klávesnice po spuštění skriptu.

D) Upravte řešení A tak, že se počet písmen zadá jako první argument skriptu.

Cvičení II

A) Vykreslete plný obdélník z písmen X. Rozměry obdélníku zadá uživatel pomocí argumentů skriptu.

B) Upravte řešení A tak, že vykreslíte pouze obrys obdélníku.

Cvičení III

A) Vykreslete kružnici nebo kruh z písmen X. Poloměr a to zda se má vykreslit kružnice či kruh zadá uživatel z klávesnice po spuštění skriptu.

Cvičení IV

A) Napište skript implementující jednoduchou interaktivní kalkulačku. Skript se bude ptát na dvě čísla a operaci, která se s nimi má provést. Po zadání dat skript znázorní výsledek a zeptá se uživatele, zda-li chce pokračovat nebo skript ukončit.

Cvičení V

Vysvětlete rozdílné chování následujících skriptů. Soubor data.txt obsahuje pět řádků.

```
#!/bin/bash
I=0
cat data.txt | while read A; do
    I=$(($I+1))
done
echo $I
```

vypíše číslo 0

```
#!/bin/bash
I=0
while read A; do
    I=$(($I+1))
done < data.txt
echo $I
```

vypíše číslo 5

Cvičení VI

A) Soubor rst.out (wolfn:/home/kulhanek/Data/rst.out) obsahuje výsledky z molekulové dynamiky. Úkolem je ze souboru vyextrahovat závislost teploty simulovaného systému na čase.

```
...
NSTEP =     2000   TIME(PS) =       2.000  TEMP(K) =   292.99  PRESS =     0.0
...
```

čas

teplota

Průběh teploty znázorněte v programu gnuplot.

POZOR: skript nesmí obsahovat příkazy grep, awk a ani jejich varianty

Cvičení VII

A) 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 (http://personal.cscs.ch/~mvalle/mencoder/mencoder.html). Video přehrajte pomocí příkazu mplayer.