

C2110 Operační systém UNIX a základy programování

9. lekce / modul 2

PS/2020 Distanční forma výuky: Rev3

Petr Kulhánek

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

Cyklus 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.

```
for A in `seq 1 0.25 10`; do
    printf "%8.3f\n" $A
done
```

Vypíše reálná čísla v intervalu od 1 do 10 s přírůstkem 0,25. Čísla budou uvedeny s přesností tří desetinných míst a zarovnány doprava v poli o délce 8 znaků.

Dokumentace: man seq

Cvičení 1

1. Napište skript, který vypíše reálná čísla v intervalu od -10 do 10 s přírůstkem 0,5. Čísla budou uvedena včetně znaménka, zarovnány doprava v poli 10 znaků a uvedeny s přesností na jedno desetinné místo.
2. Napište skript, který pro každý soubor *.cpp v adresáři /home/kulhanek/Documents/C2110/Lesson09/data1 vypíše název souboru a počet řádků, které obsahuje. V řešení použijte for ... in a pro výpis souborů použijte příkaz ls.
3. Upravte předchozí script tak, aby pro každý soubor *.f90 v adresáři /home/kulhanek/Documents/C2110/Lesson09/data2 vypsal název souboru a počet řádků, které obsahuje. V řešení použijte for ... in a pro výpis souborů použijte příkaz find.

Cykly a přesměrování

Přesměrování a roury - vstup

Pomocí roury nebo přesměrování je možné do cyklu předat vstupní data. Typické použití je načítání souboru/vstupního proudu po řádcích:

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

roura

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

přesměrování

Přesměrování a roury - vstup

Vstup může být "**konzumován**" příkazy v bloku cyklu. Tomu je možné zabránit přesměrování standardního vstupu z /dev/null pro blok příkazů uvedených do {}.


```
cat soubor.txt | while read A; do
{
    prikaz2
    ...
} < /dev/null
done
```

```
while read A; do
{
    prikaz2
    ...
} < /dev/null
done < soubor.txt
```


Přesměrování a roury - výstup

Přesměrování výstupu do souboru:


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



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

Roura:

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



Výstup všech příkazů v cyklu je spojen se standardním vstupem příkazu **tee**. Ten jej vytiskne na standardní výstup a zároveň uloží do souboru **soubor.txt**.

Přesměrování a roury - příklady

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

stejná funkcionality

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

```
for((I=1;I <= 10;I++)); do  
    echo $I  
    printf "N=%10d\n" $I  
done > soubor.txt
```

rozdílná funkcionality

```
rm -f soubor.txt  
for((I=1;I <= 10;I++)); do  
    echo $I >> soubor.txt  
    printf "N=%10d\n" $I  
done
```

Cvičení 2

1. Napište skript, který pro každý soubor *.f90 v adresáři /home/kulhanek/Documents/C2110/Lesson09/data2 vypsal název souboru a počet řádků, které obsahuje. V řešení použijte rouru s while read a pro výpis souborů použijte příkaz find.
2. Upravte skripty vykreslující obrazce (L08.M01.C01), tak aby se rozměr obrazce načítal ze standardního vstupu a výsledný obrazec se tisknul do souboru, jehož jméno zadá uživatel opět ze standardního vstupu.


Domácí úkoly



Domácí úkol I


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

