A7B38UOS Úvod do operačních systémů

9. Cvičení

Základy skriptů, složené příkazy, funkce

Obsah cvičení

Základy skriptů

Složené příkazy

Funkce

Jak vypadá skript?

Struktura skriptu

```
#!/bin/bash
#Příkazy skriptu
.
.
.
```

Spuštění

```
$ chmod 755 skript.sh
$ ./skript.sh
```

Příkaz if

Syntaxe

```
if seznam_příkazů_1
   then seznam_příkazů_2
fi
```

Popis

Pokud **`seznam_příkazů_1`** vrátí nulový návratový kód, je proveden **`seznam_příkazů_2`**

Příkaz if

Syntaxe

```
if seznam_příkazů_1
   then seznam_příkazů_2
   else seznam_příkazů_3
fi
```

Popis

Pokud **`seznam_příkazů_1`** vrátí nulový návratový kód, je proveden **`seznam_příkazů_2`** jinak je proveden **`seznam_příkazů_3`**

Příkaz if

Syntaxe

```
if seznam_příkazů_1
   then seznam_příkazů_2
[      elif seznam_příkazů_3
         then seznam_příkazů_4 ]
...
[        else seznam_příkazů_5 ]
fi
```

Popis

Pokud **`seznam_příkazů_1`** vrátí nulový návratový kód, je proveden **`seznam_příkazů_2`** jinak, pokud **`seznam_příkazů_3`** vrátí nulový návratový kód, je proveden **`seznam_příkazů_4`** jinak ...

Příkaz if - příklady

Zjištění existence souboru:

```
if [ -f /etc/fstab ] ; then echo "Je to dobry." ; fi
```

Zjištění existence účtu roota

```
$ if grep root /etc/passwd > /dev/null
> then
> echo "ucet root existuje"
> else
> echo "ucet root neexistuje"
> fi
```

Kontrola počtu argumentů:

```
if [ $# -ne 2 ] ; then echo "Usage:..." ; fi
```

Varianty příkazu if

`seznam_příkazů_1 && seznam_příkazů_2`

je variantou:

```
if seznam_příkazů_1
   then seznam_příkazů_2
fi
```

`seznam_příkazů_1 || seznam_příkazů_2`

je variantou:

```
if seznam_příkazů_1; then :
   else seznam_příkazů_2
fi
```

Příklady:

```
$ grep XYZ soubor && lp soubor
$ grep XYZ soubor || echo "XYZ nenalezeno"
```

Příkaz case

Syntaxe

```
case hodnota in
vzor_1) seznam_příkazů_1 ;;
...
vzor_n) seznam_příkazů_n ;;
esac
```

Popis

Hodnota je postupně porovnávána se vzory (podle pravidel náhrad jmen souborů, **nikoliv regulárních výrazů**).

Pokud dojde ke shodě, je proveden seznam příkazů uvedených za vzorem a tím příkaz **case** končí.

Vzor může být tvořen více variantami oddělenými znakem |.

Bývá zvykem uvádět jako poslední vzor *) a za něj příkazy provedené v případě selhání ostatních vzorů.

Příkaz case - příklady

Zpracování argumentu:

```
#!/bin/sh
case "$1" in
'start')
  [ -x /usr/lib/lpsched ] && /usr/lib/lpsched ;;
'stop')
  [ -x /usr/lib/lpshut ] && /usr/lib/lpshut ;;
*)
  echo "Usage: $0 { start | stop }"
  exit 1
  ;;
esac
```

Příkaz case - příklady

Zpracování přepínačů a argumentů skriptu:

```
#!/bin/sh
case "$1" in
-[tT])
  echo "Parametr -t nebo -T" ;;
yes|no)
  echo "Parametr yes nebo no" ;;
*)
  echo "Neznamy argument."
  exit 1 ;;
esac
```

Příkaz case - příklady

Určení akcí pro různé dny v týdnu:

```
DAY=`date +%w`
case $DAY in
 0) echo "Full backup";;
 6) echo "No backup";;
 *) echo "Partial backup";;
esac
```

Cyklus while

Syntaxe

```
while seznam_příkazů_1
do
 seznam_příkazů_2
done
```

Popis

Dokud `seznam_příkazů_1` vrací nulový návratový kód, je prováděn `seznam_příkazů_2`.

Cyklus je možno předčasně ukončit příkazem `break`.

Předčasný návrat na začátek příkazu lze příkazem `continue`.

Cyklus while

```
#!/bin/sh
MAX=5
I=1
while [ $I -le $MAX ]
do
  echo "Hodnota I je $I"
  I=`expr $I + 1`
done
```

Cyklus while

```
#!/bin/sh
while :
do
  /bin/echo -n "Zadej cele cislo [0-99][k=konec]: "
  read C
  case $C in
   k)
    break
   ;;
   [0-9]|[0-9][0-9] )
    echo "Druha mocnina cisla $C je `expr $C \* $C`."
   ;;
   *) echo "Neplatny argument."
  esac
done
```

Cyklus until

Syntaxe

```
until seznam_příkazů_1
do
 seznam_příkazů_2
done
```

Popis

Obdoba příkazu **`while`** s obrácenou podmínkou.

Dokud **`seznam_příkazů_1`** vrací nenulový návratový kód, je prováděn **`seznam_příkazů_2`**.

Příkazy **`break`** a **`continue`** lze použít jako u cyklu **`while`**.

Cyklus for

Syntaxe

```
for proměnná in seznam_argumentů
do
   seznam_příkazů
done
```

Popis

Cyklus **`for`** proběhne tolikrát, kolik má **`seznam_argumentů`** elementů.

V každém cyklu nabývá proměnná hodnotu jednoho elementu ze seznam_argumentů.

Příkazy **`break`** a **`continue`** lze použít jako u cyklu **`while`**.

Cyklus for - příklady

Zpracování seznamu:

```
#!/bin/sh
I=1
for E in Petr Jana Jiri Karel
do
  echo "Element $I je $E."
  I=`expr $I + 1`
done
```

Cyklus for - počítadlo

Jiný zápis for cyklu:

```
for ((i=0; i<100; i++))
do
  printf "%02d " $i
done
printf "\n"
```

Funkce

Syntaxe

```
function jméno_funkce
{
  seznam_příkazů
}
```

Popis

Funkce se volá jménem a volitelnými parametry

Počet a hodnoty parametrů jsou uvnitř funkce dostupné pomocí $# a pozičních parametrů $1, $2, ...

Funkce může volat jinou funkci případně sebe rekurzivně.

Návrat z funkce proběhne po provedení všech příkazů s

návratovým kódem posledního provedeného příkazu

nebo předčasně příkazem return návratový_kód.

Funkce

```
maximum()
{
 if [ $1 -gt $2 ] ; then
  echo "$1"
 else
  echo "$2"
 fi
}

if [ $# -ne 2 ] ; then
 echo "Pouziti: $0 num1 num2"
 exit 1
fi
echo "Maximun je `maximum $1 $2` "
```