



Úvod do Operačních Systémů

9. cvičení

Proměnné, složené příkazy, funkce, numerické výpočty.





- Numerické výpočty
 - v shellu
 - externí příkazy
- Proměnné
 - práce s proměnnými
 - práce s pozičními parametry
- Bloky příkazů
- Podmíněné vykonávání příkazů
- Složené příkazy
 - podmínky
 - cykly
- Funkce



- Inkrementace hodnoty proměnné

```
max=`ypcat passwd | cut -d: -f3 | sort -n | tail -1`
```

```
max=`expr $max + 1`
```

```
max=$((max+1))
```

```
((max=max+1))
```

```
((max++))
```

```
let max=max+1
```

```
let max++
```

```
max=`awk "BEGIN {print max=$max+1}" <&-`
```



- Průměrná velikost obyč. souborů v aktuálním adresáři

```
ls -l | awk '/^-/ {sum+=$5;c++} END {print sum/c}'  
ls -l | awk '/^-/ {sum+=$5;c++}  
END {printf("%f\n", sum/c)}'
```

- Průměrný počet systémových volání za sekundu

```
calls=`vmstat -s | grep calls | tr -dc '[0-9]`  
uptime=`expr 60 \* \  
$(/home/courses/Y36U0S/common/cv09/uptime_min)`  
cps=$((calls/uptime))  
  
echo $calls/$uptime | bc -l
```



- Výpočty v různých soustavách

```
echo $((012))
```

```
echo $((10#012))
```

- Ternární operátor podmínky

```
x=5
```

```
echo $((x%2==0 ? x : x+1))
```

- Práce s řetězci a RE

```
RE=' /.*\/'
```

```
expr $PWD : "$RE"
```

```
RE=' /.*\/\ (.*\ )\/'
```

```
expr $PWD : "$RE"
```



- Zjištění limitů programu bc

```
grep BC_ /usr/include/limits.h
```

```
let MAX_SCALE=`sed -n '/BC_SCALE_MAX/s/.* \([0-9]*\)\/\1/p' /usr/include/limits.h`-1
```

- Vytvoření konstanty pi s maximální přesností programu bc

```
typeset -r pi=`echo "scale=$MAX_SCALE;4*a(1)" \  
| bc -l | tr -dc '[0-9].'`
```

```
echo $pi
```

```
pi=3.14
```

```
echo $pi
```



1:10am up 35 day(s), 4:02, 2 users, load average: 0.05, ...

RE='.* \([0-9]*\) day[^0-9]* \([0-9]*\) :0\{0,1\} \([0-9]*\) .*'

- Výpočet doby běhu serveru v minutách (shell)

```
((uptime=`uptime | sed "s/$RE/10#\1*24*60 + 10#\2*60 + 10#\3/"`))
```

- Výpočet doby běhu serveru v minutách (awk)

```
uptime=`uptime | sed "s/$RE/\1 \2 \3/" \
| awk '{print $1*24*60 + $2*60 + $3}'`
```

- Výpočet doby běhu serveru v minutách (eval)

```
eval `uptime | sed "s/$RE/d=\1 h=\2 m=\3/"`
((uptime=10#d*24*60 + 10#h*60 + 10#m))
```



- Přiřazení hodnoty do proměnné

```
prefix=mylog
```

```
date=`date +%y%m%d`
```

```
file=$PWD/${prefix}_${date}
```

- Výpis obsahu proměnné

```
echo "$file"
```

- Nastavení implicitní hodnoty

```
echo "${data_file:=./data}"
```

```
echo "data_file=$data_file" > my.conf
```

- Nastavení hodnoty proměnné ze souboru

```
df=`sed -n '/^data file=/s/.*=//p' my.conf` [^=]
```




- Délka hodnoty proměnné

```
file_len=${#file} ; echo $file_len
```

- Část obsahu proměnné (bash)

```
s=`expr $file_len - 6`  
echo "${file:s}"
```

- Nejkratší oříznutí obsahu zprava vzorem

```
image=screenshot.jpg  
cp /home/courses/Y36U0S/common/cv09/$image .  
convert "$image" "${image%.jpg}.tif"  
file "${image:0:10}"*
```



- Nejdelší oříznutí obsahu zleva vzorem

```
item="${file##*/}_1"
```

```
echo $item
```

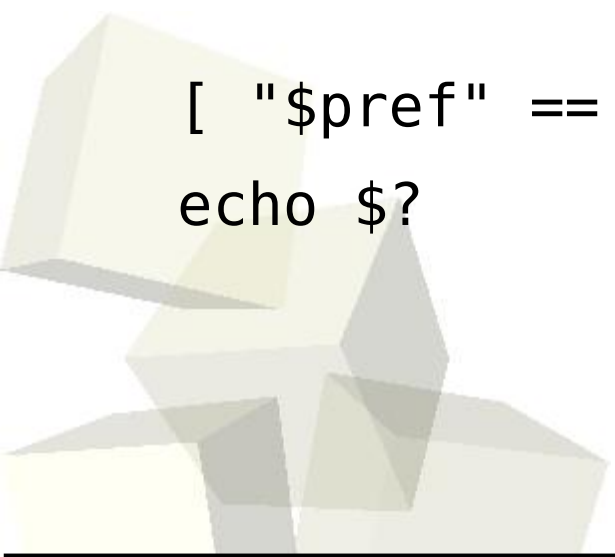
- Nejdelší oříznutí obsahu zprava vzorem

```
pref="${item%%_*}"
```

```
echo "$pref"
```

```
[ "$pref" == "$prefix" ]
```

```
echo $?
```





- Nastavení pozičních parametrů

```
IFS=":" ; set `grep root /etc/passwd`
```

- Název programu

```
echo $0
```

- Výpis pozičních parametrů

```
echo $1, ${5}:$7
```

- Počet pozičních parametrů

```
[ $# -eq 7 ] ; echo $?
```

- Všechny poziční parametry

```
echo $* : @$
```



- Deklarace pole

```
files[1]=s1
```

```
files[2]=s2
```

```
files[3]=s3
```

- Použití všech položek pole

```
touch "${files[@]}"
```

- Délka pole (počet položek)

```
echo ${#files[*]}
```

- Zrušení pole

```
unset files
```



- Substituce v hodnotě proměnné

```
echo "$file"
```

```
echo "${file/mylog/$USER-log}"
```

```
echo "${file/[0-9]/-}" "${file//[0-9]/-}"
```

- Nepřímá indexace poziční proměnné

```
IFS=":" ; set `grep root /etc/passwd`
```

```
for ((i=1; i<=7; i++)); do echo $i: ${!i}; done
```

- Nastavení typu proměnné

```
typeset -i int_x=5
```

```
int_x=abc
```



- Vykonání skupiny příkazů v aktuálním shellu

```
ls ; date ; who > s1  
{ ls ; date ; who; } > s1  
{ sleep 60 ; w > users; } &
```

- Vykonání skupiny příkazů v subshellu

```
cd /tmp ; pwd  
cd  
( cd /tmp; pwd ) ; pwd  
x=5; ( x=7; echo $x ) ; echo $x
```



Podmíněné vykonání příkazů

- Podmíněné vykonání po úspěchu předešlého příkazu
`tar xvf archiv.tar && rm archiv.tar`
- Podmíněné vykonání po neúspěchu předešlého příkazu
`[-d adr] || mkdir adr`
- Použití obou podmínek
`ping dray2 &>/dev/null \
&& ssh dray2 uptime \
|| echo dray2 je nedostupny`
- Podmíněné vykonání bloku příkazů
`[-d /adr] && { cd /; tar cvf ~/adr.tar adr; }`

! && a || není logické AND a OR !



Složené příkazy – podmínky

- Kontrola počtu argumentů

```
USAGE="Pouziti: $0 arg1 arg2"
```

```
if [ $# -ne 2 ]; then echo "$USAGE"; else echo $*; fi
```

- 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
```




Složené příkazy – cykly

- Postupné zpracování argumentů (while)

```
set bash ksh sh
```

```
while [ $# -gt 0 ]
```

```
do
```

```
    man "$1" > man\ $1
```

```
    shift
```

```
done
```

- Postupné zpracování argumentů (for)

```
set man\ *sh
```

```
for manfile in "$@"; do wc -l "$manfile"; done
```



Složené příkazy – cykly !

- Rozdíl mezi \$* a \$@

```
set man\ *sh
```

```
for manfile in $*; do wc -l "$manfile"; done
```

```
set man\ *sh
```

```
for manfile in "$*"; do wc -l "$manfile"; done
```

- Zpracování výsledku náhrady příkazu a názvů souborů

```
touch 'my long filename1' 'my long filename2'
```

```
for file in my*; do echo ".$file."; done
```

```
for file in `ls my*`; do echo ".$file."; done
```

```
for file in "`ls my*`"; do echo ".$file."; done
```

```
IFS="
```

```
" ; for file in `ls my*`; do echo ".$file."; done
```



Složené příkazy – cykly +

- Jiný zápis *for* cyklu

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

- Cyklus *select*

```
select manfile in man\ *sh
do
    wc -l "$manfile"
done
^D
```



```
function max() {  
    [ $# -eq 0 ] && return 2  
    local max=$1; shift  
    while [ $# -gt 0 ]  
    do  
        if ! echo "$1" | grep '[^0-9]' >/dev/null; then  
            [ "$1" -gt "$max" ] && max=$1; fi  
        shift  
    done  
    echo "$max"  
}
```

max 7 1 6 5 9 4 2 3



Příprava na příští cvičení I

- Upravte funkci max tak, aby reagovala na špatné argumenty návratovým kódem 1.
- Napište skript, který vypíše zadané argumenty skriptu (poziční parametry) v náhodném pořadí na výstup.
- Upravte předešlý skript tak, aby jediným argumentem byl vzor. Jména obyčejných souborů v aktuálním adresáři odpovídající zadanému vzoru budou vypisovány v náhodném pořadí na výstup.
- Napište skript, který najde v zadaném adresáři a jeho podadresářích největší obyčejný soubor a nejstarší obyčejný soubor daného uživatele. Skript otestuje, zda zadaný adresář existuje a zda má uživatel dostatek práv.



Příprava na příští cvičení II

- V souboru `~/.bashrc` nadefinujte funkci `log` (a zavolejte), která do zvoleného log souboru vloží při startu shellu řádku oznamující start shellu.
Do souboru `~/.bash_logout` nadefinujte funkci `log` (a zavolejte), která do log souboru vloží řádku oznamující ukončení shellu. Funkce `log` by se měla zavolat i při ukončení shellu na základě příchodu signálu.
Vytvořte z obsahu log souboru tabulku obsahující informace o jménu počítače a délce běhu shellu.
Log soubor by měl mít následující podobu:

```
Nov 19 19:07:20 dray3 bash (4533): starting (version 2.14.0), pid 4607, user barinkl
datum a čas hostname PPID verze PID username
```

- Vytvořte z obsahu log souboru tabulku obsahující informace o datu startu jménu počítače a délce běhu shellu.
- Vytvořte na základě předešlé tabulky statistiku používání jednotlivých počítačů.
- Vytvořte přehled použitých počítačů a verzí shellů.