

## Tvorba skriptů I



# Skripty & Programování

- **Interpretovaný jazyk**

- Nutnost mít zdrojový kód->typicky txt soubor
- Nutnost mít **interpeter**
- Obecně pomalejší
  - Octave, Matlab
  - BASH
  - PHP,Perl, ....

- **Kompilovaný jazyk**

- Zdrojový kód je přeložen pomocí kompilery do strojového kódu
- Výhodou je rychlost

- **Kombinace obojího**

- Nejprve kompilace do **byte-kódu**
- Potom interpretace, nebo kompilace do do strojového kódu
  - C# (.NET jazyky)
  - Java



# Skripty v Linuxu

- Nutnost specifikace cesty k interpretu
- Použití `> which bash --> /usr/bin/bash`

`#!/usr/bin/octave`

```
A=[1,2,3;4,5,6;7,8,9]
dA=det(A);
printf("Determinant je %f\n",dA)
b=[4;5;6]
x=A^-1*b
```

`#!/usr/bin/gnuplot`

```
faktorial(n)=(n>1) ? n*faktorial(n-1):1;
set grid
set title "Porovnani faktorial a
exponenciala"
plot [1:6] faktorial(x) , exp(x)
pause(-1)
```



# Skripty v BASHI

- **Důležité proměnné**

- \$0 – název skriptu
- \$# -počet předepsaných parametrů
- \$\$ - PID procesu skriptu
- \$1,\$2 ... \$9 poziční parametry
  - Posun pomocí **shift**
- \$@ - seznam parametrů
- \$? - návratová hodnota posledního spuštěného příkazu( skriptu)
  - Použití u **exit**



# Proměnné ukázka

## **./promenne.sh a b c d e f**

```
#!/usr/bin/bash
#prvni skript
echo "Bezi skript $0"
echo "Seznam pozicnich parametru $@"
echo "Pocet pozicnich parametru $#"
```

echo "Nekolik parametru \$1 \$2 \$3 ... \$9"

echo "Zadej posun"

read posun

shift \$posun #posun

echo "A opet nekolik parametru \$1 \$2 \$3 ... \$9"

echo ">\$posun"

echo '>>\$posun'

exit 0



# Podmínky I

- Příkaz **test** , [ podmínka ]
  - Mezery jsou důležité
- **Testování řetězců**
  - Shoda [ \$s1 = \$s2 ], lepší [ "\$s1" = "\$s2" ]
  - Neshoda [ "\$s1" != "\$s2" ]
  - Nenulový (-n), prázdný (-z),
    - [ -n "\$s1" ]
  - Numerické testy
    - negace !
    - [ V1 -eq V2 ] (V1=V2)
    - [ V1 -ne V2 ] (V1!=V2)
    - [ V1 -gt V2 ] (V1>V2), -ge <=
    - [ V1 -lt V2 ] ( V1<V2), -le <=



# Podmínky II

- **Testy souborů**

- -d pravda pro adresář
- -f pravda pro normální soubor
- -r,-w,-x pravda má li daná práva

- **Konstrukce if,elif, &&, ||**

```
#!/bin/bash
if [ -n "$1" ] && [ -n "$2" ]; then
    echo "$1 zdravi $2"
else
    echo "Spatne pouziti"
    exit 1
fi
exit 0
```

```
#!/bin/bash
if grep -E "^[[:digit:]]+$" <<< "$1" 2>&1 > /dev/null; then
    echo "Vstupem je cislo $1"
elif grep -o -E "[[:digit:]]+" <<< "$1" 2>&1 >/dev/null
;then
    echo "Cast je cislo"
else
    echo "KO"; exit 1
fi
exit 0
```



# Cykly I

- **Cyklus for: podobnost s foreach v C#**

–

```
for n in {a..b}; do  
    echo $n  
done
```

```
#!/bin/bash  
for soubor in $( ls -l | tr -s " " | grep -E "^\-.*" | sort -t " " -n -k 5,5 | cut -d " " -f 8 )  
do  
    echo $soubor  
done
```

```
#!/bin/bash  
for((i=1; "$i" <= "10" ;i=$((i+1))))  
do  
    echo $i  
done
```





# Cykly II

- Cyklus **while**, **until**

```
#!/bin/bash
tmp=1
while [ "$tmp" -le "20" ]
do
    echo $tmp
    sleep "$tmp"s
    tmp=$((tmp+1))
done
```

```
#!/bin/bash
until who | grep "$1" > /dev/null
do
    sleep 30s
done
echo "Uzivatel $1 se prihlasil"
exit 0
```

```
#!/bin/bash
find $HOME -type f -name "*.jpg" -o -name "*.png" 2> /dev/null > soubory.txt
while read soubor
do
    echo $soubor
done < soubory.txt
```

- Příkazy **break**, **continue**

