

## F4270 UNIX, počítačové sítě (jaro 2016), Cvičení 7

Příkazy a řídicí struktury: `printf`, `read`, `for`, `while`, `if`, `case`, `printf`, `expr`, `test`, `dirname`, `basename`.  
Proměnné a expanze:  `$#`,  `$0`,  `$n`,  `${n}`,  `shift`,  `$`,  `set -`,  `$?`,  `$$`,  `$!`,  `$()`.  
Defaultní hodnota  `${parameter:-word}`, subshell pomocí  `()`, skipina  `{}`.

- A1** Napište skript, který očekává dva parametry, název souboru a číslo. Ze zadaného souboru vymaže řádku se zadaným číslem (např. pomocí `sedu`).

```
#!/bin/sh
sed -e "$2d" $1
```

- A2** Vytvořte skript, který dostane `n+1` parametrů, první obsahuje číslo `x` mezi 1 a `n`, skript vypíše parametr s číslem (`x+1`). Tj. například, skript `2 a b c` vypíše `b`, tedy ten třetí parametr (ale druhý, když budete počítat od `a`). Použijte `shift` s parametrem.

```
#!/bin/sh
shift $1
echo $1
```

- A3** Napište skript, který očekává jeden až dva číselné parametry. První číselný parametr pak zarovná na počet míst zadaných druhým číselným parametrem. Pokud druhý parametr není zadaný, implicitně se rozumí 5. Zkuste také zarovnání nulama. Na konci výpisu odřádkujte.

```
#!/bin/sh
printf "%${2:-5}d\n" $1
printf "%0${2:-5}d\n" $1
```

- A4** Zkuste co vypíše následující příkazy:

```
[guy@nix ~]$ a="ahoj"; { echo $a ; a="cau" ; echo $a ; } ; echo $a
[guy@nix ~]$ b="ahoj"; ( echo $b ; b="cau" ; echo $b ; ) ; echo $b
[guy@nix ~]$ x="ahoj"; x="cau" sh -c 'echo $x'; echo $x
[guy@nix ~]$ y="ahoj"; y="cau"; sh -c 'echo $y'; echo $y
[guy@nix ~]$ z="ahoj"; sh -c 'echo $z'
[guy@nix ~]$ export u="ahoj"; sh -c 'echo $u'
```

- A5** Rozmyslete si, co dělají následující příkazy:

```
[guy@nix ~]$ a=1;b=2; { a=LEVY; echo $a >&2; } | \
{ b=PRAVY; echo $b; tr 'A-Z' 'a-z'; }; echo "A=$a,B=$b"
[guy@nix ~]$ a=1;b=2; { a=LEVY; echo $a; } | \
{ b=PRAVY; echo $b; tr 'A-Z' 'a-z'; }; echo "A=$a,B=$b"
```

- A6** Spočítá následující skript správně počet řádků souboru `/etc/passwd`?

```
pocet=0;
cat /etc/passwd | while read x
do
    pocet=`expr $pocet + 1`
    echo $x, $pocet
done
echo $pocet
```

A co tento?

```
pocet=0;
cat /etc/passwd | { while read x
do
    pocet=`expr $pocet + 1`
    echo $x, $pocet
done
echo $pocet; }
```

A co tento?

```
pocet=0
while read x
do
    pocet=`expr $pocet + 1`
done </etc/passwd
echo $pocet
```

A co tento?

```
pocet=0
while read x </etc/passwd
do
    pocet=`expr $pocet + 1`
done
echo $pocet
```

Jaký je mezi těmito skripty rozdíl?

- A7** Napište skript, který dostane v parametrech seznam souborů zadaný cestami, které mohou být i relativní. Ke každému souboru vypíše řádek s absolutní cestou.

```
#!/bin/sh
# Je-li část in ... u for cyklu vynechaná, doplní se in "$@"
for x in $@
do
    adresar=$(dirname "$x")
    adresar=$(cd $adresar && pwd)
    echo $adresar/${basename $x}
done
```

- A8** Napište skript, který přejmenuje všechny soubory v aktuálním adresáři na soubory se týmž jménem, kde jsou velká písmena změněná na malá. (Například soubor ZALOHA.ZIP by přejmenoval na zaloha.zip)

```
#!/bin/sh
for x in *[A-Z]*
do
    mv "$x" "`echo $x | tr '[:upper:]' '[:lower:]'`"
done
```

- A9** Napište skript, který přejmenuje všechny soubory v aktuálním adresáři s příponou .jpeg na soubory s týmž jménem, ale příponou .jpg. (Zkuste využít expanze \${proměnná%přípona}, sed, nebo dirname a basename.)

```
# Varianta pomocí expanze (není ve standardu moc dlouho)
for x in *.jpeg
do
    mv "$x" "${x%jpeg}.jpg"
done
```

```
# Varianta pomocí sedu:
for x in *.jpeg
do
    mv "$x" "$(echo $x | sed 's/\.jpeg$/\.jpg/')"
done
```

```
# Varianta pomocí basename a dirname.
for x in *.jpeg
do
    mv "$x" "$(dirname "$x")/${basename "$x" .jpeg}.jpg"
done
```

Nemáte-li potřebné soubory, tak si je vytvořte:

```
[guy@nix ~]$ for ((i=1;i<=10;i++)); do touch obrazek`printf "%03d.jpeg" $i`; done
```