

Řešení příkladů z 6. cvičení ZOS

1. Napište skript, který spojí dva soubory, které budou zadány jako parametry a před každým souborem vypíše jeho jméno (soubor1.txt obsah soubor2.txt obsah)

```
#!/bin/bash

clear
echo $1
cat $1
echo $2
cat $2
```

Pozn.: příkaz clear zde samozřejmě být nemusí, slouží pouze k vyčištění obrazovky. (clear the terminal screen).

2. Modifikujte skript z bodu jedna, aby čísloval řádky v souboru – jde to kontinuálně?

```
#!/bin/bash

clear
echo $1
cat $1 | nl
echo $2
cat $2 | nl
```

kontinuální číslování:

```
#!/bin/bash
X=${1+`cat $1 | wc -l`}
echo $1
cat $1 | nl
echo $2
cat $2 | nl -v $X
```

3. Modifikujte skript z bodu jedna pro libovolný počet souborů.

```
#!/bin/bash
for I in $*;
do
    echo $I
    cat $I
done
```

Cyklus *for* se často používá pro iteraci přes množinu souborů (např. *.txt), nebo přes množinu argumentů, jako v tomto případě.

Dejte si především pozor na nutnost uvést příkaz *do* na **dalším řádku** – pokud by byl na stejném řádku jako příkaz *for*, je potřeba na konci *for* uvést **středník** (zde je i tak). Stejně tak si povšimněte práce s proměnou *I*, když se jí přiřazuje hodnota, je *I* samotné, při přístupu k hodnotě proměnné *I* předchází \$, tj. \$*I*. Jiný příklad (v přiřazení nesmí být mezera):

```
V=10; echo "Napisu test na $V bodu."
```

Kontinuální řešení:

```
#!/bin/bash

X=1
for f in $@
do
    echo $f
    cat $f | nl -v $X
    X=$((X+`cat $f | wc -l`))
done
```

4. Modifikujte skript, aby vypisoval nejdříve “Ahoj, já jsem skript jméno a tady je obsah souborů: “

5. Dalším krokem bude vypsání počtu parametrů

```
#!/bin/bash

clear
echo "Ahoj, ja jsem skript $0 a mam $# parametru"
for I in $*;
do
    echo $I
    cat $I
done
```

6. Ošetřete vstup, aby nebylo možné zadat více jak 10 souborů.

```
#!/bin/bash

MAX_PARAMS=10

if [ $# -le $MAX_PARAMS ];
then
    echo "Ahoj, ja jsem skript $0 a mam $# parametru"
    for I in $*;
    do
        echo $I
        cat $I
    done
else
    echo "Lituji, ale bylo zadano vice jake $MAX_PARAMS
parametru"
fi
```

Povšimněte si, že z hlediska štabní kultury je vhodné nadefinovat na začátku skriptu magické hodnoty (MAX_PARAMS), aby je mohl případný uživatel snadno modifikovat.

7. Popřípadě, že bude víc souborů, vypište jedenáctý, který je navíc.

```
#!/bin/bash

MAX_PARAMS=10
FIRST_PREB=$(( $MAX_PARAMS + 1 ))

if [ $# -le $MAX_PARAMS ];
then
    echo "Ahoj, ja jsem skript $0 a mam $# parametru"
    for I in $*;
    do
        echo $I
        cat $I
    done
else
    echo "Lituji, ale bylo zadano vice jake $MAX_PARAMS
parametru"
    echo "Prvni prabytecny je $FIRST_PREB a to je
${!FIRST_PREB}"
fi
```

Další možností, jak vypsat jedenáctý parametr v bashi je:

```
echo ${11}      .. FUNGUJE
echo $11        .. NEFUNGUJE, vypíše první parametr a za něj jedničku
```

Nebo:

```
shift          .. shift argumentů, na místo prvního se dostane původně druhý
shift
echo $9
```

8. Vyzkoušejte spustit skript na pozadí, co se stane, když mu zadáte nezadáte argumenty?

```
./skript par1 par2 par3 &
```

```
./skript & par1 par2 par3
```

v čem je rozdíl? :-)

9. Upravte ho tak, aby soubory vypsal desetkrát po sobě s 5ti vteřinovým intervalem

```
#!/bin/bash

MAX_PARAMS=10
HOW_MUCH=10
SLEEP_TIME=5
X=1

clear
if [ $# -le $MAX_PARAMS ];
then
    echo "Ahoj, ja jsem skript $0 a mam $# parametru, ted
vypisu ${HOW_MUCH}x jejich obsah s $SLEEP_TIME vterinovou
prodlevou..."
    while [ $X -le $HOW_MUCH ];
    do
        for I in $*;
        do
            echo $I
            cat $I
        done
        sleep $SLEEP_TIME
        X=`expr $X + 1`
    done
else
```

```
    echo "Lituji, ale bylo zadano vice jake $MAX_PARAMS
parametru"
fi
```

Povšimněte si způsobu zvětšení hodnoty proměnné X o 1, další varianta je např (pozor, přiřazení bez mezer):

```
X=$((X+1))
```

Dále pozor na mezery v podmínce u if nebo while za [a před] :

```
while [ $X -le $HOW_MUCH ];
```

10. Zkuste ve skriptu vypsat speciální znaky – ahoj, já jsem skript “jmeno“ a budu vypisovat soubory – můj první parametr \$1 je “jmeno souboru” a poslední \$Číslo je “jméno souboru“

zkuste a porovnejte (případné překlapy vyhrazeny :-)):

```
#!/bin/bash
```

```
echo "Ahoj, ja jsem skript \"$0\" a budu vypisovat soubory -
muj prvni parametr \"$1\" je \"$1\" a posledni \\$# je
\\${!#}\" "
```

```
echo 'Ahoj, ja jsem skript \"$0\" a budu vypisovat soubory -
muj prvni parametr \"$1\" je \"$1\" a posledni cislo je
\\$#\\'
```