

Operációs rendszerek alapjai

IV. gyakorlat

Függvények

Függvények bash-ben

```
function_name () {  
    <commands>  
}
```

```
function function_name {  
    <commands>  
}
```

Függvények bash-ben

- Az előbb látott két függvény specifikáció között nincs lényegi különbség
- A () -jel megtévesztő lehet a függvény neve után, a programozási nyelvek többségében ide írhatjuk a függvények paramétereit, a bash esetében azonban ez csak “dísz”, azt “jelöli”, hogy függvényről van szó
- A függvény definíciójának előbb kell lennie a scriptben mint, hogy meghívnánk a függvényt!

Kérdés

Hol itt a hiba?

```
#!/bin/bash

function function_name1 {
echo 1
}

function_name2 () {

function_name1
function_name3
}

function function_name3 () {
echo 3
}

function_name2
```



Kérdés

És itt hol a hiba?

```
#!/bin/bash

function function_name1 {
echo 1
}

function_name2 () {

function_name1
function_name3
}

function_name2

function function_name3 () {
echo 3
}
```



Függvény és script argumentumok

- ~~myfunction (arg1, arg2, ...)~~ ez az ami nincs
- helyette van:
 - `myfunction arg1 arg2`
- az átadott paraméterek a függvényen belül a \$ jel és a paraméter sorszáma kombinációval érhetők el
 - azaz pl az `arg1` elérése: `$1`
 - `$#` - megadja hány paramétere van a függvénynek illetve scriptnek
 - `$@` - az összes paraméter egy tömbben
 - `$0` - a script fájl neve

Kérdés

Mit ír ki a program?

```
#!/bin/bash

function function_name1 {
echo $2
}

function_name2 () {
function_name1 $2 $1
}

function_name2 1 2

function function_name3 () {
echo 3
}
```



Függvény visszatérési értéke

- A legtöbb programozási nyelvből ismert függvény visszatérési érték nincs a bash-ben
- Vannak módszerek (hackek) rá
 - return status
 - csak (egész)szám lehet
 - a 0 jelentené azt, hogy sikeresen lefutott, a többi szám valamilyen hibát jelölhet
 - a \$? változóban található meg a legutolsó lefutott parancs vagy függvény statusa
 - parancs helyettesítés: value=\$(myfunction)
- Megjegyzés:
 - a return status működik a scriptek esetén is!

Lokális és globális változók

- A változók alapértelmezetten globálisak
 - vagyis a scriptben mindenhol láthatóak
- De vannak lokális változók is
 - `local` a varázsszava: `local var="local is local"`
 - a függvényen belül létrehozott lokális változók csak a függvényen belül láthatók
 - ha ugyanazzal a névvel létezik globális változó is, akkor az külön fog változni

Kérdés

Mit csinál ez?

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

```
ls () {  
    ls -l  
}
```

```
ls
```



Parancsok felülírása

- A `command` kulcsszóval tudjuk jelezni, hogy nem a függvényre, hanem a parancsra gondolunk:

```
#!/bin/bash

ls () {
    command ls -l
}

ls
```

Parancsok felülírása

- A `command` kulcsszóval tudjuk jelezni, hogy nem a függvényre, hanem a parancsra gondolunk:

```
#!/bin/bash

command () {
    command ls -l
}

command
```

G04F01

- Csináljunk egy scriptet ami kiírja a paramétereit fordított sorrendben
- Hint:

Az utolsó paraméter kiírása:

```
#!/bin/bash  
  
i=$#  
echo ${!i}
```

G04F02

- Csináljunk egy scriptet ami lassan elkezd számolni a végtelenségig
 - lassan = minden egyes szám kiírása után vár 2 másodpercet
- A ctrl+c megnyomásakor írja ki, hogy Hello és folytassa a számolást!
- Hint:

```
#!/bin/bash

trap ctrl_c INT

function ctrl_c () {
    echo "Jajj, most elkaptal :("
    ls
    exit
}

ls () {

    ls -l
}

ls
```

G04F03

- Készítsünk egy scriptet ami megszámolja a megadott mappában lévő fájlokat, linkeket és könyvtárakat.
 - A scriptet lehessen 0, 1 vagy 2 paraméterrel futtatni
 - Első paraméter: melyik mappában szeretnénk a számolást végezni (útvonal megadása)
 - Ha nincs paraméter, akkor az aktuális mappában keresi a fájlokat.
 - Második paraméter: miket akarunk megszámolni:
 - -d: könyvtárak
 - -f: fájlok
 - -l: linkek
 - Ha nincs második paraméter, akkor mind a hármat írja ki.

G04F04

- Írj egy scriptet, ami monitorozza, hogy hány szálon (folyamatként) fut egy adott program. A script X másodperces időközönként egy fájlba írja a kimenetét, majd a CTRL+C-vel megállítva kiírja a szálak átlagos számát.
 - Amennyiben a fájl már létezik, a szkriptnek törölnie kell a monitorozás megkezdése előtt.
 - A script bemenet (pl.: G04F04 chromium 3):
 - a monitorozandó folyamat neve
 - időköz (másodpercben)
 - Kimeneti fájl tartalma:
 - a fájl neve: (folyamat neve)_processes.log
 - a fájl egy sorának tartalma: idő (HH:MM:SS) - a folyamatok száma
 - Kimenet a terminálra:
 - a folyamatok átlagos száma