

Operációs rendszerek alapjai

XI. gyakorlat

indirekt referencia, alias, gyakorlás

Indirekt referencia

- Változót névvel tudunk megadni
- Ahhoz, hogy a tényleges értéket “kinyerjük” belőle, változó behelyettesítést kell használnunk
 - Azaz a változó neve elé kell írunk egy \$ karaktert
- De mi van akkor, ha a változó értéke egy másik változó?
 - Indirekt referencia

Indirekt referencia

- Pl.:

var=12

ref_var=var

- Hogyan kérdezzük le a *var* értékét a *ref_var* változón keresztül?
 - *\$ref_var*
 - *\$\$ref_var*
 - *\\$\$ref_var*
 - *\$(\$ref_var)*
 - *\$(\\$\$ref_var)*

Indirekt referencia - eval

- eval: evaluation
 - shell parancsként futtatja az argumentumát
 - ha több argumentumot is kap, akkor azokat egyetlen stringgé fűzi össze
- Példa:

a='\$b'

b='\$c'

c=d

echo \$a

eval echo \$a

eval eval echo \$a

Indirekt referencia - eval

- Pl.:

```
var=12
```

```
ref_var=var
```

- Hogyan kérdezzük le a `var` értékét a `ref_var` változón keresztül? -> `eval` segítségével

```
eval ref_var=\$$ref_var
```

```
echo "ref_var = $ref_var"
```

Indirekt referencia - indirect expansion

- indirect expansion:
 - `${!<name_of_var>}`
 - Ha a paraméter első karaktere felkiáltójel, akkor a változó indirekt módon lesz kiértékelve
 - vagyis a ! jel utáni rész értéke lesz a változó neve és annak a változónak az értéke kerül behelyettesítésre

Indirekt referencia - indirect expansion

- indirect expansion:

- Példa:

```
var=30
```

```
ref_var=var
```

```
((var++))
```

```
echo "ref_var = "${!ref_var}
```

alias

- Az alias parancs segítségével különböző parancsoknak tudunk adni egy “helyettesítő” parancsot
 - lényegében egy shortcut-ot csinálunk a parancshoz
 - Példa:

alias ll="ls -l"

- ezekután elég annyit írni, hogy ll az ls -l helyett
- alias megszüntetése: unalias ll

alias

- Ha scriptben akarunk alias parancsokat használni, akkor ezt be kell állítani:
 - shopt -s expand_aliases
 - shopt:
 - shell options - különböző opciók beállítása
 - Példa:
alias xyz='cat \$0'
xyz

G11F01

- Készítsünk egy véletlen számokat generáló bash scriptet!
 - Ha a scriptet **paraméter nélkül** futtatjuk, akkor **0 és 10 közötti egész számokat generáljon** egyenletes eloszlással (a 0 és a 10 is legyen benne).
 - Ha egy **egész N** számot kap **paraméterül**, akkor **0 és N közötti egész számokat generáljon** egyenletes eloszlással (a 0 és az N is legyen benne).
 - Ha két egész számot (**N illetve M**) kap **paraméterül**, akkor **N és M közötti egész számokat generáljon** egyenletes eloszlással (az N és az M is legyen benne).
 - Ha **kettőnél több paraméterrel** futtatjuk, **írja ki, hogy túl sok a paraméter**.
- A generált véletlen számot a konzolra írja ki
- Feltételezhetjük, hogy a program csak 32767-nél kisebb, -32767-nél nagyobb egész számokat kap paraméterül.
 - azaz például a törtszámokat vagy szöveges paramétereket nem kell lekezelni
 - a negatív egész számokat viszont igen, azokra is jól kell működjön!
- Pl.: gipja11_ZH02F02.txt 0 -2 esetén a lehetséges kimenetek: 0,-1,-2

G11F02

- Adott egy számsorozat, pl.: 11
- A feladat, hogy “megnézzük és elmondjuk mit látunk, majd írjuk le.”
 - Azaz a fenti példánál maradva amit látunk az 2 db 1-gyes, ekkor amit le kell írunk az a 21
 - Ezután amit látunk az 1 db 2-tes és 1 db 1-gyes, vagyis 1211
 - és így tovább...
- Készítsünk awk scriptet, amely egy kiindulási számsorozatra visszatér a fentieknek megfelelően, azzal, hogy mit látunk.
- Teszteljük le a programunkat, 1-gyes kiindulási szekvenciával, 5 lépésen keresztül.

Hint: substr(szoveg, 2, 1) - a szoveg változó 2. karaktere

G11F03

- Készítsünk egy bash scriptet, amely képes a paraméterül kapott két stringről eldönteni, hogy azok anagrammák-e vagy sem.
 - anagramma:
 - olyan szövegek, amelyek ugyanazon karakterekből állnak, csak más sorrendben (kis és nagybetű valamint a szóköz nem számít, a darabszám viszont igen)
 - Pl.:
 - Elbow - Below
 - The eyes - They see
 - Eleven plus two - Twelve plus one