

**Környezeti változók, alias, első shell szkriptek****1. Környezet**

1. Hozzuk létre a `test1_shell` nevű szkriptet, melynek tartalma a következő két sor legyen:

```
#!/bin/bash
echo $var
```

2. Tegyük futtathatóvá a szkriptet, majd az interaktív terminálban adjuk ki a következő parancsokat:

```
$ var=pingvin
$ echo $var
```

Eredmény: \_\_\_\_\_

```
$ ./test1_shell
```

Eredmény: \_\_\_\_\_

3. Konklúzió?
- \_\_\_\_\_

4. Az interaktív shellben adjuk ki a következő parancsokat:

```
$ export var=pingvin
$ ./test1_shell
```

Eredmény: \_\_\_\_\_

5. Konklúzió?
- \_\_\_\_\_

6. Hozunk létre egy `test2_shell` nevű szkriptet is:

```
#!/bin/bash
var=roka
echo $var
```

majd az interaktív shellben futtassuk le a következő parancsokat:

```
$ ./test2_shell
```

Eredmény: \_\_\_\_\_

```
$ echo $var
```

Eredmény: \_\_\_\_\_

7. Konklúzió?
- \_\_\_\_\_

8. Futtassuk le a `test2_shell` szkriptet az aktuális interaktív shellen belül, azaz a szkriptet ne egy gyermek shellben hajtsuk végre:

\_\_\_\_\_

Mi a `var` változó tartalma?

\_\_\_\_\_

9. Hozzuk létre a `test3_shell` szkriptet:

```
#!/bin/bash
echo szia
ls *.java >& /dev/null
exit 3
```

Az `ls` parancs eredménye át lett irányítva a `/dev/null` kimenetre, vagyis a parancs kimenete elveszett (a feladat szempontjából ez most nem lényeges).

Futtassuk le ezt a szkriptet egy gyermek shellben. Az interaktív shellben hogyan tudjuk lekérdezni a szkript befejezőkor visszaadott kilépési kódját (mely jelen esetben 3)?

\_\_\_\_\_

10. Most közvetlenül az interaktív shellben futtassuk le a parancsot:

```
ls *.java >& /dev/null
```

Mi a parancs által visszaadott kilépési kód?

\_\_\_\_\_

Ugyanez a kérdés a következő parancs esetében is:

```
ls >& /dev/null
```

Konklúzió:

\_\_\_\_\_

**2. Alias**

Az alias parancs segítségével egy hosszabb utasítást vagy utasítássorozatot tudunk lerövidíteni.

1. Állapítsa meg, hogy az `ll` beállított alias-e:
- \_\_\_\_\_

2. Készítsen az `ls -l` parancshoz egy `ll` nevű alias:
- \_\_\_\_\_

Ellenőrizze le, hogy sikeresen létrejött-e az alias:

\_\_\_\_\_

3. Próbálja ki az előzőleg létrehozott alias:
- \_\_\_\_\_

Majd szüntesse meg ezt az alias beállítást:

\_\_\_\_\_

4. Hozza létre az `ls` alias az `ls -F` parancshoz.
- \_\_\_\_\_

5. Az előző pontban létrehozott `ls` alias megszüntetése nélkül hogyan tudjuk meghívni az „alap” `ls` parancsot (vagyis a `-F` kapcsoló nélkül)?

---

6. Egy alias hivatkozhat egy másik aliasra? És saját magára?

---

7. Készítsünk egy `xc` nevű alias-t, mely megnyitja az `xclock` programot.

---

8. Készítsünk egy `xcb` nevű alias-t, mely az `xclock` programot a *háttérben* nyitja meg.

---

9. Nyissunk meg egy újabb terminált s próbáljuk meg használni az `xc` alias-t ebben az új ablakban. Mit tapasztalunk?

---

10. Melyik fájlban kell elhelyezni az aliasokat, ha nem akarjuk őket a terminálablak bezárása után sem elveszíteni?

---

11. Az alábbi listában összegyűjtöttünk néhány hasznos alias-t:

```
alias md='mkdir'  
alias rm='rm -i'  
alias cp='cp -i'  
alias mv='mv -i'  
alias d='ls -al'  
alias cls='clear'  
alias nh='nautilus . 2>/dev/null'  
alias p='python'
```

12. Kérdezzük le az összes elérhető alias listáját:

---

### 3. Az első shell szkriptem

1. Hozzuk létre a `hello.sh` nevű fájlt a következő tartalommal:

```
#!/bin/bash  
echo Hello World
```

2. Adjunk a szkriptnek végrehajtási jogot majd futtassuk le.

### 4. Üdvözlés

1. Írjunk egy `szia` nevű shell szkriptet, mely név szerint üdvözlöl minket a login nevünket felhasználva:

```
$ ./szia  
Szia <login_név>!
```

A szkript nem kap egyetlen argumentumot sem. (Emlékezzünk az első gyakorlaton látott `whoami` parancsra.)

2. Futtassuk le úgy a szkriptünket, hogy a képernyőn megjelenő üzenettel egyidőben egy `szia.txt` fájl is jöjjön létre, melynek tartalma ugyanez az üdvözlő szöveg legyen. (Lásd a `tee` parancs használatát.)