

Bash script alapok

1. rész



Linux alapok

Varga Tibi 2020

A Bash

- A shell az operációs rendszer legkülső része, mely a felhasználókkal való kommunikációra szolgál, felhasználói parancsokkal. A shell-nek több típusa is van.
- Az egyik legelterjedtebb a BASH shell (Bourne Again SHell),
- Ha több linux parancsot szeretnénk egymás után végrehajtani, bonyolultabb lekérdezések, fájlműveletek, végrehajtások esetén hasznos lehet egy olyan script, amelyen belül ezeket az utasításokat egy rendszerbe foglalhatjuk és egy **BASH script megindításával** lefuttathatjuk.

A script fájl

A BASH script-jeink kiterjesztése *.sh* lesz. Úgyhogy kezdhetjük azzal a további pontok átnézését, hogy létrehozunk egy fájlt ezzel a kiterjesztéssel.

A futtatáshoz a script fájlt futtathatóvá kell tenni:

- Példa: **# chmod u+rx fájlnev.sh**
- Script futtatása: a script fájlt a **# ./fájlnev.sh**

vagy a **# bash fájlnev.sh** parancsokkal futtathatjuk

Futtatás parancssori paraméterekkel: az indító parancs után paramétereket helyezünk el melyre a scriptben hivatkozhattuk

./fájlnev.sh param1 param2

Bash script szintaxis

Minden BASH script első sora: Ez az ún. "Hash-Bang" vagy "She-Bang". Ez mondja meg a shell-nek, hogy milyen programmal interpretálja a scriptet

#!/bin/bash

Komment a bash scriptekben

- Egy soros:

#

- Több soros:

:

tetszőleges szöveg

,

Parancssori paraméterek

Néhány frekvenciátaltabban használt paraméter:

- **\$#** - A parancssori paraméterek száma.
- **\$?** - A legutoljára végrehajtott parancs visszatérési értéke.
- **\$n** - Az n-edik parancssori paraméter értéke.
(Tehát pl. \$1 az 1., \$2 a 2., ... paraméter értéke.)
- **\$0** - A pillanatnyi héjprogram neve.
- **\$*** - Valamennyi parancssori paraméter egyben, egyetlen karakterláncként (" \$1 \$2 ... \$9 ...").

Példa

```
1  #!/bin/bash
2  echo "Ez a script parancs sori paramétereit mutatja be"
3  #ez egy egysoros komment
4  echo "a paraméterek száma: $#"
```

5 echo "első paraméter: \$1"

6 echo "második paraméter: \$2"

7 echo "a helyi program neve: \$0"

8 echo "minden paraméter: \$*"

9

```
10  << ////
11  ez
12  pedig
13  több soros
14  komment
15
16  ////
```

```
tibi@server:~/test$ ./param.sh 33 77
Ez a script parancs sori paramétereit mutatja be
a paraméterek száma: 2
első paraméter: 33
második paraméter: 77
a helyi program neve: ./param.sh
minden paraméter: 33 77
```

Változók

\$változonev vagy **\${változonev}**

- A változó neve nem kezdődhet számmal. A változó behelyettesítés használatakor a változónév elé egy \$ karaktert írunk. Az adott helyre a BASH a változó értékét helyettesíti be.

- **Nagyon fontos, hogy értékadáskor NEM lehet helyköz sem az = bal, sem jobb oldalán.**

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

```
VAL="Bela"
```

```
echo $VAL
```

```
echo ${VAL}25
```

- A kiírt szöveg a "Bela" majd "Bela25" lesz.
- Tehát egymagában ez így **NEM** állná meg a helyét (változó kiíratás esetén): echo VAL

Érték beolvasása

read változonev

Belvasott értékre hivatkozás: **a \$+név**

Kiírás: **echo " " vagy printf " "**

Példa:

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

```
read VAL
```

```
echo $VAL
```

```
printf $VAL
```


Idézőjelek használata

- " " - felismeri a speciális karaktereket.
- ' ' - nem ismeri fel a speciális karaktereket.
- `` - (AltGr + 7) parancsvégrehajtást jelöl.

- Példa:

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

```
VAL="Bela"
```

```
echo "VAL = $VAL"
```

```
echo 'VAL = $VAL'
```

```
VAL=`echo $VAL | tr a-z A-Z`
```

```
echo $VAL
```

- Kiíratott szöveg:

```
VAL = Bela
```

```
VAL = $VAL
```

```
BELA
```

Beágyazott utasítás

`$(utasitas)`

- Ha a programkódon belül szeretnénk végrehajtani egy utasítást.

`#!/bin/bash`

`VAL=$(pwd)`

`echo $VAL`

- A VAL tartalma az aktuális elérési útvonal lesz. Ezt íratjuk ki az utolsó sorban.
- Tehát egymagában ez így NEM állná meg a helyét (utasításra hivatkozás esetén): `VAL=pwd`

Példa

```
1  #! /bin/bash
2  echo "Ez a sript beolvasás és kiírást mutat be"
3  #érték bekérése
4  echo "add meg a szöveget"
5  read szoveg
6  echo "A beírt szöveg: $szoveg"
7  printf "A printf is kiír: $szoveg"
8
9  echo "Most kiírunk néhány környezeti változót"
10 echo "Te neved $USER Hello $USER !"
11 echo "Az alapértelmezett könyvtáraid: $PATH"
12 echo "A használt nyelv $LANG"
13
14 echo "Ez egy beágyazott utasítás minta"
15 echo "Az aktuális könyvtáramiben vagy $(pwd)"
16
```

Példa

```
tibi@server:~/test$ bash beolvas.sh
Ez a sript beolvasás és kiírást mutat be
add meg a szöveget
Linux és máris bux!
A beírt szöveg: Linux és máris bux!

A printf is kiír: Linux és máris bux!
Most kiírunk néhány környezeti változót

Te neved tibi

Az alapértelmezett könyvtáraid :/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b
in:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin

A használt nyelv hu_HU.UTF-8

Ez egy beágyazott utasítás minta

Az aktuális könyvtáramiban vagy /home/tibi/test
```

Logikai operátorok, relációs jelek, műveletek

Használatos operátorok

- **egészértékű aritmetika:** =, +, -, *, **, /, %, ++, --,
- **bitenkénti matematikai:** |, &, ^, ~, <<, >>
- **logikai relációs :** ==, !=, >, <, >=, <=

ezeknek a visszatérési értéke: 0 - false 1 – true

- **logikai:** &&, ||, !, expr ? expr : expr

ezeknek az inputban minden hamis, ami 0, és minden igaz, ami nem hamis

ezeknek a visszatérési értéke: 0 - false 1 - true

Matematikai kifejezések

Az alábbi példában 3 lehetőségünk is van (A VAL értéke mindig 5 lesz):

- **expr**

VAL=`expr 3 + 2`

Kötelezően helyközzel az értékek között!

- **\$(())**

VAL=\$((3+2))

- **let**

let VAL=3+2 # - ez a legpraktikusabb

- Tehát egymagában ez így **NEM** állná meg a helyét (művelet végrehajtás esetén): VAL=3+2

Példa

```
1  #!/bin/bash
2  echo "Ez a script matematikai műveleteket mutat be"
3  #érték bekérése
4  echo "add a első számot"
5  read szam1
6  echo "add a második számot"
7  read szam2
8  let eredmeny=$szam1+$szam2
9  printf "Az összegük: $eredmeny "
10 let eredmeny=$szam1-$szam2
11 printf "A különbségük: $eredmeny "
12 let eredmeny=$szam1%$szam2
13 printf "Az maradék osztásnál: $eredmeny"
14 let eredmeny=$szam1*$szam2
15 printf "A szorzatuk: $eredmeny "
16 let eredmeny=$szam1**2
17 printf "Az első szám négyzete: $eredmeny
18 "
```

Példa

```
tibi@server:~/test$ bash beolvas.sh
Ez a sript matematikai műveleteket mutat be
add a első számot
5
add a második számot
2
Az összegük: 7
A különbségük: 3
Az maradék osztásnál: 1
A szorzatuk: 10
Az első szám négyzete: 25
```