

Računalniški praktikum 1

Uvod v bash

Vida Groznik

Zanka while

```
1. #!/bin/bash
2. # zanka while
3.
4. counter=1
5. while [ $counter -le 10 ]
6. do
7.     echo $counter
8.     ((counter++))
9. done
```

While (dokler) je izraz/test (test je med oglatimi oklepaji []) **true**, izvajaj vrstice kode:

```
while [ <test> ]
do
    <ukazi>
done
```

Vrstica 4 – Inicializacija spremenljivke counter z začetno vrednostjo.

Vrstica 5 – **While** (dokler) je test **true** (counter je < ali = 10) **do** izvajaj naslednje ukaze.

Vrstica 7 – Izpiši vrednost spremenljivke counter.

Vrstica 8 – Povečaj vrednost spremenljivke counter za 1.

Vrstica 9 – Smo na koncu zanke, pojdi nazaj na vrstico 5 in ponovno izvedi test. Če je vrednost testa true, ponovno izvedi ukaze. Če je vrednost testa false, potem nadaljuj z izvajanjem ukazov, ki so za besedo done.

Zanka until

```
1. #!/bin/bash
2. # zanka until
3.
4. counter=1
5. until [ $counter -gt 10 ]
6. do
7.     echo $counter
8.     ((counter++))
9. done
```

Zanka **until** je podobna zanki while. Razlika je v tem, da bo izvajala ukaze znotraj zanke **dokler test ne postane true**.

```
until [ <test> ]
do
    <ukazi>
done
```

Vrstica 5 – **Until** (dokler) ni test **true** (counter je > 10) **do** izvajaj naslednje ukaze.

Vrstica 7 – Izpiši vrednost spremenljivke counter.

Vrstica 8 – Povečaj vrednost spremenljivke counter za 1.

Vaje

1. Napiši skripto, ki z uporabo zanke while na zaslon izpiše številke od največje do najmanjše: primer: 5, 4, 3, 2, 1
2. Napiši skripto, ki podano številko izpiše v obratnem vrstnem redu (npr. če je številka 123, mora izpisati 321).
3. Napiši skripto, ki izpiše vsoto cifer podanega števila (npr. če je število 123 bo vsota cifer tega števila $1+2+3 = 6$).

Zanka for

```
1. #!/bin/bash
2. # osnovna zanka for
3.
4. names='Alen Marko Aleks'
5.
6. for name in $names
7. do
8.     echo $name
9. done
```

Za vsak element v podanem seznamu, izvedi podane ukaze. Zanka for ima naslednji sintakso:

```
for var in <list>
do
    <ukazi>
done
```

Zanka **for** bo vzela posamezen element iz seznama **list** (po vrsti, enega za drugim), dodelila spremenljivki **var** ta element kot njeno vrednost, izvedla ukaze med besedama **do** in **done**, šla nazaj na vrh zanke, vzela naslednji element seznama in vse še enkrat ponovila.

Seznam je definiran kot zaporedje nizov, ki so ločeni s presledki.

Vrstica 4 – Ustvari enostaven seznam, ki vsebuje niz imen.

Vrstica 6 – Vsak element v seznamu **\$names** dodeli spremenljivki **\$name** in izvedi sledeče ukaze.

Vrstica 8 – Izpiši ime na ekran.

Zanka for - razpon

```
1. #!/bin/bash
2. # primer z razponom v zanki for
3.
4. for value in {1..5}
5. do
6.     echo $value
7. Done
```

```
1. #!/bin/bash
2. # primer z razponom po korakih v
3. zanki for
4.
5. for value in {10..0..2}
6. do
7.     echo $value
8. done
```

Obdelujemo lahko tudi niz števil. Za določitev razpona števil, uporabimo v zanki for zavite oklepaje {No1..**No2**} in določimo poljubno začetno in poljubno končno število. Med zavitimi oklepaji **ni presledkov**!

Začteno število je lahko tudi večje od končnega. Posledično se bojo vrednosti zmanjševale.

Prav tako lahko določimo velikost koraka (za koliko se vsaka naslednja vrednost razlikuje od predhodne). To dosežemo, če dodamo še dve piki (..) in vrednost koraka (za koliko se naslednja vrednost razlikuje od predhodne).

Zanka `for` – vključitev spremenljivk

```
1. #!/bin/bash
2. # izračun razpona s spremenljivko
3. N=10
4.
5. for value in $(seq 2 $N)
6. do
7.     echo $value
8. done
```

```
1. #!/bin/bash
2. # izračun razpona s spremenljivko
3. N=10
4.
5. for ((val=1;val<=$N;val++));
6. do
7.     echo $val
8. done
```

Pogosto si želimo v pogoju zanke `for` uporabiti vrednost neke spremenljivke. Ker klasičen način s seznamom oz. razponom tega ne omogoča, se lahko poslužimo uporabe matematičnih izrazov.

Vaje

Napiši skripte z uporabo **zanke for**, ki bodo na ekran izrisale naslednje oblike:

a)

```
1
22
333
4444
55555
```

b)

```
1
12
123
1234
12345
```

c)

```
*
**
***
****
*****
```

d)

```
*
**
***
****
*****
*****
*****
****
***
**
*
```

e)

```
      1
    2 2
  3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
```

f)

```
      *
    * *
  * * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```


break & continue

V nekaterih primerih si želimo spremeniti delovanje zanke. V ta namen lahko uporabimo dva ukaza: **break** in **continue**.

break

Z ukazom **break** nemudoma zaključimo z izvajanjem zanke.

continue

Z ukazom **continue** prenehamo z izvajanjem trenutne iteracije/obhoda zanke in začnemo z izvajanjem naslednje iteracije/obhoda.

Mehanizem select

```
1.  #!/bin/bash
2.  # Enostaven primer s select
3.
4.  names='Alen Marko Aleks Quit'
5.
6.  PS3='Select person: '
7.
8.  select name in $names
9.  do
10.     if [ $name == 'Quit' ]
11.     then
12.         break
13.     fi
14.     echo Hello $name
15. done
16.
17. echo Bye
```

Mehanizem **select** omogoča ustvarjanje enostavnega menijskega sistema. Format:

```
select var in <list>
do
    <ukazi>
done
```

Na ekran izpiše vse elemente seznama, ki jih oštevilči z zaporednimi številkami. Za tem se izpiše „poziv“, ki omogoča, da uporabnik izbere številko. Ko je številka vnešena in uporabnik pritisne enter, se bo izbrana vrednost dodelila spremenljivki **var** izvedli pa se bodo ukazi med **do** in **done**.

Ko se ukazi izvedejo, se ponovno izpiše poziv, da lahko uporabnik izbere še kaj drugega.

Vrstica 4 – Določitev vrednosti spremenljivki - seznam imen in kot zadnjo opcijo še vrednost, ki jo lahko uporabnik izbere, da se zaključi izvajanje mehanizma.

Vrstica 6 – Spremeni vrednost spremenljivke PS3 z nečim bolj opisnim. (Privzeto je #?)

Vrstice 10 - 13 – Če je izbrana zadnja opcija, 'Quit', zaključi z izvajanjem mehanizma select.