



Bash Shell Script

Part 3





BASH

THE BOURNE-AGAIN SHELL

How to create and run a script in Linux?





This is an introduction to Scripts in Linux

This is the **third part** of a series of 3 lessons. Make sure you go through them all successively to better understand the concept.

Let's get started!



Table of content

1. The for loop
2. The while loop



1

The for loop

The for loop structure

The for loop

- ◇ The **for loop** is used to **iterate through a list of items to perform repetitive tasks**
- ◇ Its **structure** is as follows:

```
for item in (list);  
do  
command 1  
command 2  
...  
command n  
done
```

For each item, the various commands will be executed
The loop will exit when the list is exhausted

The for loop

Example: Write a for loop to delete a list of users with the following usernames: **username1**, **username2**, **username3**

Solution:

```
for item in username1 username2 username3;  
do  
  userdel -r ${item}  
done
```

Let's practise this in the terminal

The for loop

Example: Write a script to create 4 regular users on the system: u6bt, u7bt, u8bt, u9bt

Solution: create a new script with `# vim for.sh` and go to the **INSERT mode**

```
for i in u6bt u7bt u8bt u9bt;
do
useradd ${i}
echo "user $i is successfully created"
sleep 3
done
```

Now, let's give the execute permission to the script: `# chmod +x for.sh`

Run the script: `./for.sh`

The for loop

You can check if the users were successfully created in the `/etc/passwd` file with: **# touch -10 /etc/passwd**

```
[root@puppetagent ~]# tail -10 /etc/passwd
dockerroot:x:988:982:Docker User:/var/lib/docker:/sbin/nologin
u2082020:x:1001:1001:Carlos Monte:/home/u2082020:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
natasha:x:1002:1002:./home/natasha:/bin/bash
harry:x:1003:1003:./home/harry:/bin/bash
serge:x:1004:1004:./home/serge:/bin/bash
u6bt:x:1005:1005:./home/u6bt:/bin/bash
u7bt:x:1006:1006:./home/u7bt:/bin/bash
u8bt:x:1007:1007:./home/u8bt:/bin/bash
u9bt:x:1008:1008:./home/u9bt:/bin/bash
```

I

The for loop

- ◇ Now let's **delete the accounts we previously created**.
- ◇ To do that, you just need to replace the **useradd** command in the script with the **userdel -r** command!

```
for i in u6bt u7bt u8bt u9bt;  
do  
  Userdel -r ${i}  
  echo "user $i is successfully deleted"  
  sleep 3  
done
```

Run the script: **./for.sh**





It may be difficult to list 100 or more users in the script.

A way to avoid this is to **create a file that will contain the list of users** and just **cat** that file in the script with: **for i in \$(cat filename)**

The for loop

Example: create a new file in the `/tmp` directory and write the the usernames in there.

```
# touch /tmp/username
```

```
# vim /tmp/username and go to the INSERT mode
```

```
u6bt  
u7bt  
u8bt  
u9bt
```

```
...
```

You can modify the `for.sh` script to create the users first and delete them after.

```
for i in $(cat /tmp/username)  
do  
...  
done
```

2

The while loop

How do we use loops in scripting?

The while loop

Just like the for loop, the **while loop** is used to iterate thru a list of items to execute some commands

Its structure is as follows:

while [condition]

do

command 1

command 2

command 3

...

command n

done

While the condition is **true**, run the commands
When the condition becomes **false**, exit the loop

The while loop

Example: create a new script to practice the while loop

```
# vim while.sh
```

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

```
while [ 2 -eq 2 ]
```

```
do
```

```
echo "This is a while loop"
```

```
sleep 2
```

```
echo "success"
```

```
done
```

What will happen here?

The loop will run indefinitely
because the condition is always
True: 2 is always equal to 2

Give the execute permission to the script: `# chmod +x while.sh`

Run the script: `./while.sh`

Since the script will continue running, you can kill it with **Ctrl-C**



The while loop

Now, let's put some logic in there **for it to stop at some point:**

```
# vim while.sh
```

```
COUNT = 0
while [ ${COUNT} -lt 6 ]
do
  echo "This is a while loop"
  sleep 2
  echo "success"
  ((COUNT++))
done
```





Here, we are applying these on simple examples for you to understand the notions. But soon, we will get into more serious usage, then you will see how useful this is in the company environment

If you don't understand it now, don't get frustrated, **we will do a lot of practice in class!**



Thanks!

Any questions?

You can find us at:

website: <http://utrainings.org/>

Phone: +1 (302) 689 3440

Email: contact@unixtrainings.com





Click on the link below to
contact the support team
for any issue!

utrans.org/support/

Create a ticket for your problem and we will get back to you soon!

