DATE : 06.01.2025

DT/NT : DT

LESSON : LINUX

SUBJECT: SHELL SCRIPTING BASIC

SESSION : 8

BATCH : B 303

AWS-DEVOPS

TECHPRO EDUCATION

techproeducation.com +1 (585) 304 29 59

Shell Scripting

BASH

THE BOURNE-AGAIN SHELL

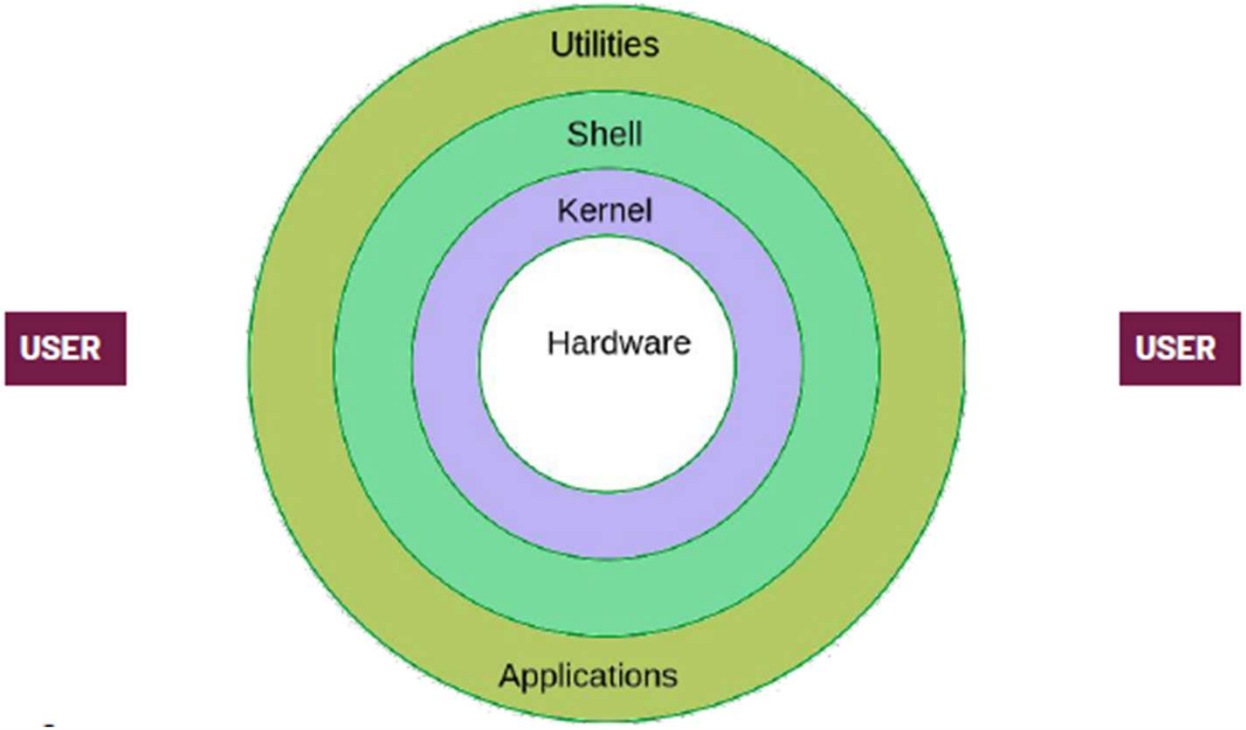
TECHPRO EDUCATION

Table of Contents

Review

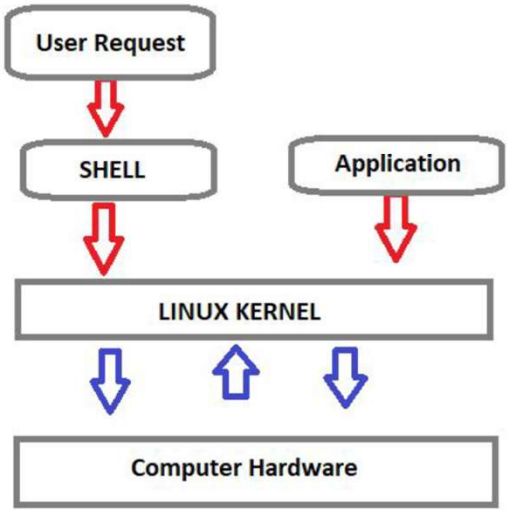
- Shell
- Bash

Components of Linux



What is SHELL?

- Shell, kullanıcının komutlarını alan ve bunları işlemesi ve çıktıları görüntülemesi için işletim sistemine veren bir programdır.
- Standart Linux kabuğu hem bir komut satırı yorumlayıcısı hem de bir programlama dilidir.



SHELL

1. **Bash : Bourne Again shell**
   The standard GNU shell, intuitive and flexible

2. **ksh : Korn shell**
   A superset of the Bourne shell

3. **csh : C shell**
   The syntax of this shell resembles that of the C programming language

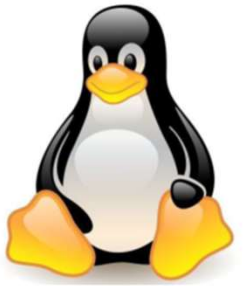
4. **tcsh: TENEX C shell**
   A Superset of the common C shell, enhancing user-friendliness and speed

5. **zsh : Z Shell**
   An extended Bourne shell with a large number of improvements, including some features of Bash, ksh, and tcsh.

Bash Prompt


LINUX@BASH-PROMPT:~$
EXAMPLE OF BASH PROMPT
PS1='\u@\h:\w\$ '
PS1 - shell prompt variable
\u - the username of the current user
\h - the hostname up to the first '.'
\w - the current working directory
\$ - if the effective UID is 0 (root / superuser), a #, otherwise a $
LINUXCONFIG.ORG

The Bash prompt



- Prompt(Bilgi istemi), sistemin bir sonraki komutu gerçekleştirmeye hazır olduğunu göstermek için kullanılan metin veya simgelerdir.
- PS1 değişkeni "Prompt String One" anlamına gelir ve birincil istem dizesini temsil eder.

```
user@host:~$
```

The Bash prompt

BASH komut istemi, bash PS1 değişkeni değiştirilerek kolayca değiştirilebilir.

```
linuxuser@techproeducation:
linuxuser@techproeducation:~$
linuxuser@techproeducation:~$ PS1="\[\e[1;32m\]\u@\h:\[\e[1;34m\]\w\[\e[m\]
\$ "
linuxuser@techproeducation:~$ PS1='\u@\h \[\e[1;34m\]\w\[\e[m\] 🚀 \$ '
linuxuser@techproeducation ~ 🚀 $ PS1='[\t] \u@\h:\w\$ '
[22:12:18] linuxuser@techproeducation:~$ PS1='\u@\h:\w$(__git_ps1 " (%s)")\
$ '
linuxuser@techproeducation:~$ PS1="\[\e[1;32m\]\u@\h:\[\e[1;34m\]\w\[\e[m\]
\$ "
linuxuser@techproeducation:~$
```

The Bash prompt

Bash prompt can be customized by using special characters. Here is the most used characters and their meaning.

Special Character	Explanation
\d	The date in "Weekday Month Date" format (e.g., "Sun Apr 12")
\h	the hostname up to the first `.'
\H	the hostname
\s	the name of the shell, the basename of $0 (the portion following the final slash)
\t	the current time in 24-hour HH:MM:SS format
\T	the current time in 12-hour HH:MM:SS format
\@	the current time in 12-hour am/pm format
\u	the username of the current user

The Bash prompt

Bash prompt can be customized by using special characters. Here is the most used characters and their meaning.

Special Character	Explanation
\v	the version of bash (e.g., 2.00)
\V	the release of bash, version + patch level (e.g., 2.00.0)
\w	the current working directory
\W	the basename of the current working directory
\!	the history number of this command

The Bash prompt

HOMEWORK

PS1'deki değişikliklerimizi nasıl kalıcı hale getirebiliriz?

Shell Scripts

Shell Scripts

What is Shell Scripting?

Shell Scripting, aşağıdakilerden biri olabilecek Unix/Linux kabuğu tarafından çalıştırılmak üzere tasarlanmış açık kaynaklı bir bilgisayar programıdır:

- The Bourne Shell
- The C Shell
- The Korn Shell
- The GNU Bourne-Again Shell

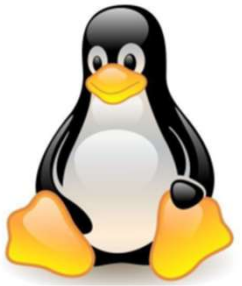
Shell Scripts

What is Shell Scripting?

- Dosya işleme, program yürütme ve metin yazdırma gibi bir kabukta yapılabilecek tipik etkinlikler aynı zamanda Shell komut dosyasıyla da yapılabilir.

- Uzun ve tekrarlayan komut dizileri, herhangi bir zamanda saklanabilen ve çalıştırılabilen tek bir komut dosyasında birleştirilebilir.

Shell Scripts

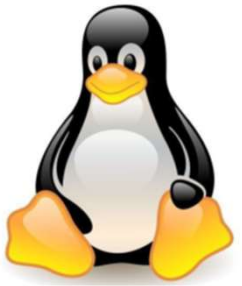


```
linuxuser@techproeducation:~$ vim script.sh
linuxuser@techproeducation:~$ chmod +x script.sh
linuxuser@techproeducation:~$ ./script.sh
Hello World
linuxuser@techproeducation:~$
```

Shebang (#!)

```
#!/bin/bash

echo "Hello World"
~
~
~
"script.sh" 3L, 32B
```

Shell Scripts

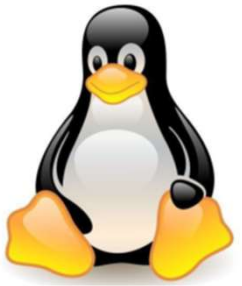


```
linuxuser@techproeducation:~$ vim script.sh
linuxuser@techproeducation:~$ chmod +x script.sh
linuxuser@techproeducation:~$ ./script.sh
Hello World
linuxuser@techproeducation:~$
```

```
#!/bin/bash

echo "Hello World"
~
~
~
"script.sh" 3L, 32B
```

chmod

Shell Scripts



```
linuxuser@techproeducation:~$ vim script.sh
linuxuser@techproeducation:~$ chmod +x script.sh
linuxuser@techproeducation:~$ ./script.sh
Hello World
linuxuser@techproeducation:~$
```

```
#!/bin/bash

echo "Hello World"
~
~
~
"script.sh" 3L, 32B
```

"./"

Shell Scripts

```
linuxuser@techproeducation:
linuxuser@techproeducation:~$ ./script.sh
Hello World
Sun Dec  3 22:32:39 +03 2023
I am learning Bash scripting
linuxuser@techproeducation:~$
```

```
linuxuser@techproeducation:
#!/bin/bash

echo "Hello World"
date
echo I am learning Bash scripting
~
"script.sh" 5L, 71B
```

Exercise 1

1. Komut dosyası oluşturun: "my-first-script.sh"
2. Bu şunu yazdırmalıdır: "Bu benim ilk komut dosyam."
3. Komut dosyasını çalıştırılabilir hale getirin.
4. Komut dosyasını yürütün.

Shell Scripts

HOMEWORK

Scriptlerinizi çalıştırırken önce “./”
koymanıza gerek olmayan bir ortam nasıl oluştururum.

Variables

- Bir değişken gerçek verinin işaretçisidir.

Kabuk değişkenleri oluşturmamızı, atamamızı ve silmemizi sağlar.

- Bir değişkenin adı yalnızca harfler (a'dan z'ye veya A'dan Z'ye), rakamlar (0'dan 9'a) veya alt çizgi karakterini (_) içerebilir ve bir harf veya alt çizgi karakteriyle başlayabilir.

- !, *, veya - gibi diğer karakterleri kullanamamanızın sebebi bu karakterlerin kabuk için özel bir anlam taşımasıdır.

Variables

```
$VARIABLE=value
$echo $VARIABLE
value
$
$my_var=my_value
$echo $my_var
my_value
$
$my-var=my-value
my-var=my-value: command not
found
$
$myvar?=my-value
myvar?=my-value: command not
found
```

TECHPRO EDUCATION

Variables

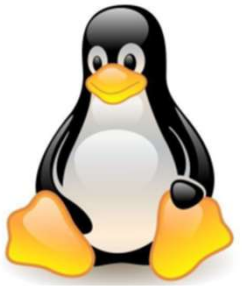
variable=value

Bu, biçimlendirmenin önemli olduğu alanlardan biridir. Oraya not et eşittir (=) işaretinin her iki yanında boşluk yoktur. Ayrıca değişken adını ayarlarken başındaki $ işaretini de bırakıyoruz.

sampledir=/etc
ls $sampledir

```
$ myvar='Hello World'
$ echo $myvar
Hello World
$ newvar="More $myvar"
$ echo $newvar
More Hello World
$ newvar='More $myvar'
$ echo $newvar
More $myvar
```

Console input



read [variable-name]

```
#!/bin/bash

echo "Enter your name: "
read name
echo Hello $name
~
~
```

```
[ec2-user@ip-172-31-36-108 ~]$ ./run.sh
Enter your name:
Raymond
Hello Raymond
[ec2-user@ip-172-31-36-108 ~]$
```

TECHPRO EDUCATION

Console input

read [variable-name]

```
#!/bin/bash

read -p "Enter Your Name: "  username
echo "Welcome $username!"
```

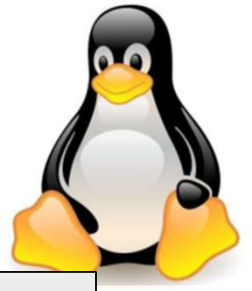
```
#!/bin/bash

read -s -p "Enter Password: " pswd
echo $pswd
```

```
#!/bin/bash

read -sp "Enter Password: " pswd
echo $pswd
```

```
#!/bin/bash

echo What cars do you like?

read car1 car2 car3

echo Your first car was: $car1
echo Your second car was: $car2
echo Your third car was: $car3
```

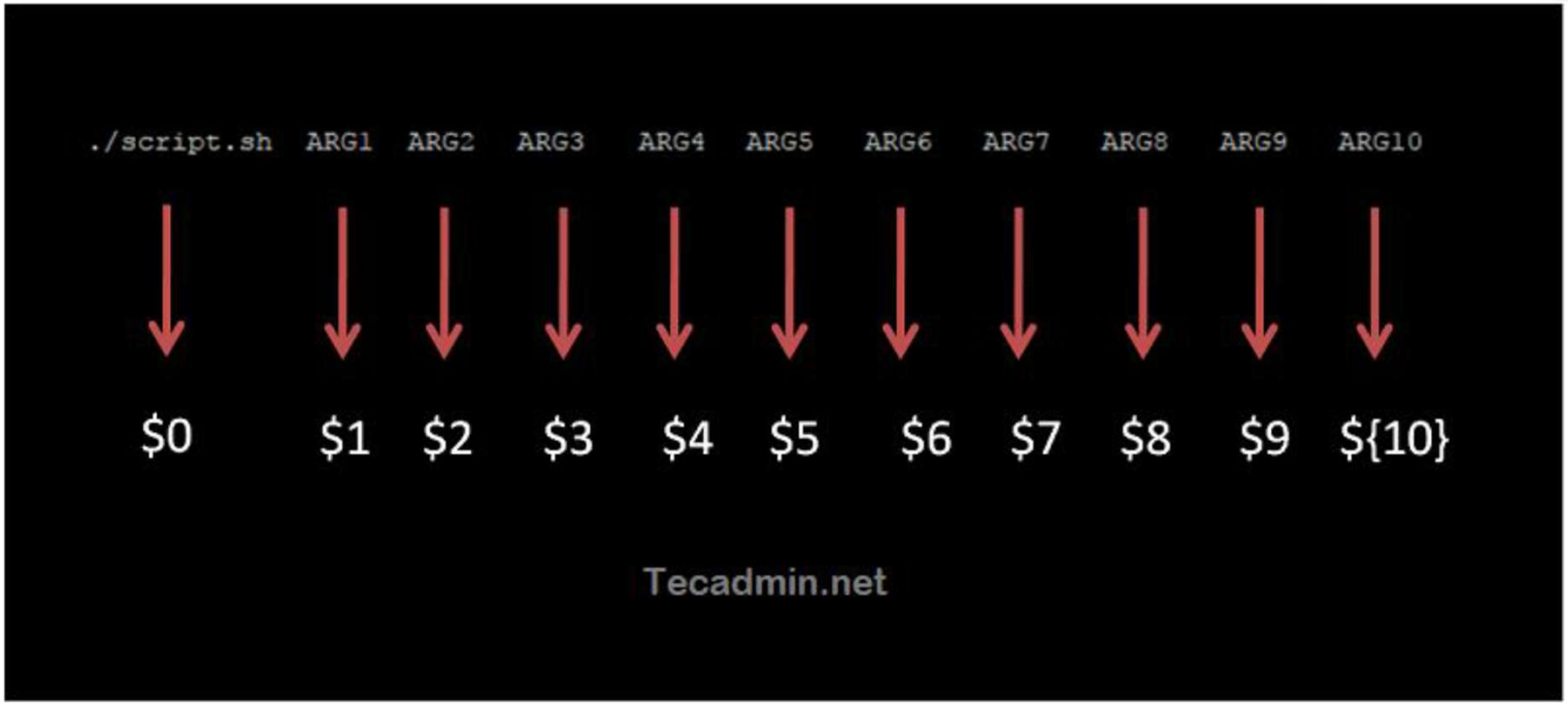
TECHPRO EDUCATION

Command Line Arguments



$0 - Bash betiğinin adı.
$1 - $9 - Bash betiğinin ilk 9 argümanı.
$# - Bash betiğine kaç tane argüman iletildiği.
$@ - Bash betiğine sağlanan tüm argümanlar.
$? - En son çalıştırılan işlemin çıkış durumu.
$$ - Geçerli betiğin işlem kimliği.
$USER - Komut dosyasını çalıştıran kullanıcının kullanıcı adı.
$HOSTNAME - Komut dosyasının üzerinde çalıştığı makinenin ana bilgisayar adı.
$SECONDS - Betiğin başlatılmasından bu yana geçen saniye sayısı.
$RANDOM - Her başvurulduğunda farklı bir rastgele sayı döndürür.
$LINENO - Bash betiğindeki geçerli satır numarasını döndürür.

Command Line Arguments



Simple Arithmetic

expr komutu, ifadenin değerini standart çıktıya yazdırır.

expr item1 operator item2

let, Bash'in basit aritmetik işlemleri yapmamıza yardımcı olan yerleşik bir işlevidir. Cevabı yazdırmak yerine sonucu bir değişkene kaydetmesi dışında expr'ye benzer.

let <arithmetic expression>

Aritmetik ifadeyi çift parantezle de değerlendirebiliriz.

$((arithmetic expression))

Arithmetic Expressions

expr item1 operator item2

```
#!/bin/bash
first_number=8
second_number=2

echo "SUM="`expr $first_number + $second_number`
echo "SUB="`expr $first_number - $second_number`
echo "MUL="`expr $first_number \* $second_number`
echo "DIV="`expr $first_number / $second_number`
```

```
$ chmod +x cal.sh
$ ./cal.sh
SUM=10
SUB=6
MUL=16
DIV=4
```

Arithmetic Expressions

let [expression]

```
#!/bin/bash

number1=8
number2=2

let total=number1+number2
let diff=number1-number2
let mult=number1*number2
let div=number1/number2

echo "Total = $total"
echo "Difference = $diff"
echo "Multiplication = $mult"
echo "Division = $div"
```

```
$ ./run.sh
Total = 10
Difference = 6
Multiplication = 16
Division = 4
```

Arithmetic Expressions

"num++" "++num" "num--" "--num"

```
#!/bin/bash

number=10
let new_number=number++
echo "Number = $number"
echo "New number = $new_number"

number=10
let new_number=--number
echo "Number = $number"
echo "New number = $new_number"

~
```

```
[ec2-user@ip-172-31-91-206 ~]$ ./run.sh
Number = 11
New number = 10
Number = 9
New number = 9
[ec2-user@ip-172-31-91-206 ~]$
```

TECHPRO EDUCATION

Arithmetic Expressions

$((Expression))
((Expression))

```
#!/bin/bash

number1=8
number2=2

echo "Total = $((number1+number2))"

((total=number1+number2))
echo "Total = $total"
~
```

```
[ec2-user@ip-172-31-91-206 ~]$ ./run.sh
Total = 10
Total = 10
[ec2-user@ip-172-31-91-206 ~]$
```

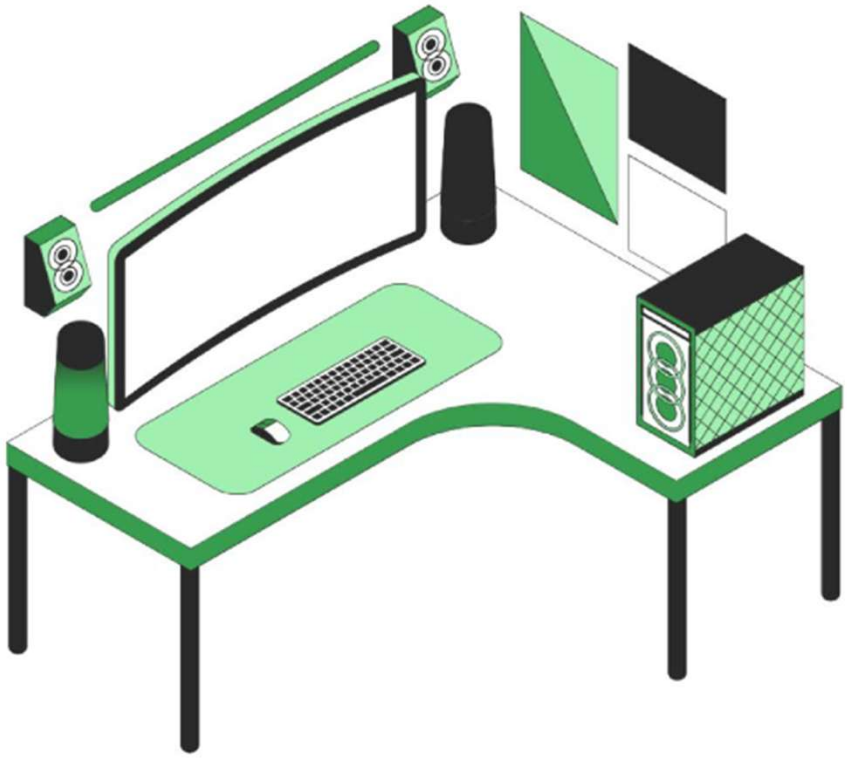
TECHPRO EDUCATION

Exercise 2

1. Kullanıcıdan **num1** ve **num2** değişkenlerine iki sayı girmesini isteyin.
2. 2 sayının **toplamını** hesaplayın.
3. **Toplam** sayıyı **yazdırın** ve 1 **artırın**.
4. **Toplam** sayının yeni değerini **yazdırın**.
5. **num1'i** toplam sayıdan çıkarın ve sonucu yazdırın.
6. **num1** ve **num2** değişkenlerini kullanıcıdan almak yerine Komut satırı argümanlarından iletilecek şekilde değiştirin

Exercise 3

1. calculate.sh adında bir komut dosyası oluşturun:
2. Varsayılan değeri 5 olan **base_value** adlı bir değişken oluşturun.
3. Kullanıcıdan başka bir numara isteyin ve bunu **user_value** değişkenine atayın.
4. **user_value** ile **base_value** değerini toplayıp bunu **toplam** değişkene atayın.
5. **toplam** değeri ekrana **“Toplam değer:”** mesajı ile yazdırın.
6. Komut dosyasını çalıştırılabilir hale getirin.
7. Komut dosyasını yürütün.



Do you have any questions?

Send it to us! We hope you learned something new.