

## 9. hafta 1. ders

-a : AND

-o : OR

! : NOT

ÖRNEK: test -r "mydoc.doc" -a -w "mydoc.doc"

mydoc.doc dosyasını yazma ve okuma izni var mı?

----- Case (çift ; kullanılır ;; )

CELAL 2 dosyası;

clear

echo "1. Date and time"

echo

echo "2. Directory listing"

ccho

echo "3. Users information"

echo

echo "4. Current Directory"

echo

echo "Enter choice (1, 2,3 or 4):"

read choice

case choice in

1) date;;

2) ls -l;;

3) who ;;

4) pwd ; ;

\*) echo wrong choice ;;

esac

[#for](#) colors in s@

for colors in Red Blue Green Yellow Orange Black Gray White

do

echo Sclor3

done

-----For loop

celal 3 dosyası:

```
for i in $*      // parametreler string olarak verilir. Renkler ile aynı durum geçerli. Ayraç olarak boşluk kabul edilir.
do              // forun başlangıcı
    if who | grep -s $i > /dev/null    // who komutu üzerinden parametre olarak verilen değişken aranıyor ($*) boş çıktı
    then
        echo $i is logged in // true döner
    else
        echo $i not available // false döner
    fi
done
```

bash celal3 ogrenci root celal çalışır. teker teker sonuç döner.

-----While loop

----- örnek 1

#!/bin/bash

while echo "Please enter command" // komut girilir.

read response // girilen komut okunur.

do // döngü başlar

case "\$response" in

'done') break ;; // eğer done yazılmışsa işlem biter

"" continue ;; // döngü başa döner

\*) eval \$response ;; // başka bir komut girilirse eval ile çalıştırılır.

esac // case biter

done

----- örnek 2

echo What kind of tree bears acorns \? // meşe palamudu ne taşır?

read response // değer girilir ve girilen değer responsa atılır.

case \$response in // değer kontrol edilir

[Oo][Aa][Kk]) echo \$response is correct ;; // oak için tüm durumlar kabul edilir.

\*) echo Sorry, response is wrong // diğer durumlarda yanlış cevap.

esac // echo biter

----- örnek 3

```
clear
echo What is the Capital of Saudi Arabia \? // arabistanın başkenti?
read answer // cevap okunur
while test $answer != Riyadh // test içine aldığı değerden 0 ya da 1 döndürür. (0dan büyük olanlar hep true döner.
do
    echo No, Wrong please try again. // riyyadh değilse 1 döner (!) cevap yanlıştır.
    read answer
done
echo This is correct. // riyyadh ise doğrudur.
```

----- örnek 4

```
clear
echo "Please Enter the user login name: \n" // isim girilir
read login_name // login_name ye atılır
until who | grep $login_name // burası doğru olana kadar döner.
do
    sleep 5
    echo "Wrong name! Please try again: \n" // yanlış olduğu sürece tekrar giriş bekler.
    read login_name // tekrar giriş bekler
done
echo The user $login_name has logged in // doğru olduğunda çıktı gelecek.
```

----- Select loop

```
select name in word1 ... wordN // değişken ismi ve değişkenler.
do
    list
done
```

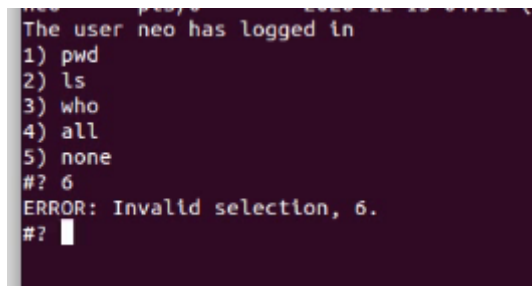
şeklinde kullanılır.

-her bir item bir numara ile gösterilir. seçim yapılması beklenir

-seçim için bir giriş verilmesi gerekir.

----- örnek

```
select op in pwd ls who all none    // bunlardan birini seç diyor.
do
    case $op in    // değerleri okuyor
        pwd| ls |who ) eval $op ;;    // değeri okuyup çalıştırıyor komutu
        all) pwd; ls; who;; // all ise hepsini koşuyor
        none) break ;; // none ise çıkış yapar
        *) echo "ERROR: Invalid selection, $REPLY." ;;    // başka bir şey ise sistemde tanımlanan değişkenden değer döner
    esac
done
```



```
The user neo has logged in
1) pwd
2) ls
3) who
4) all
5) none
#? 6
ERROR: Invalid selection, 6.
#? 
```

çıktı bu şekilde olur.

\$REPLAY → \* olan duruma girilen değeri döner. sistem tarafından tanımlanmış bir değişkendir.

export → çevre değişkenlerine ekleme yapar.

basename: girilen pathın son değişkenini döndürür. en sondaki dosya adı gelir.

parametreleri vardır. Ama hoca değinmemiş.

\$> basename /usr/bin/sh değer olarak sh döner.

----- printf

echodan farklı olarak \n bastırıyor.

echoda \n var printf de yok.

printf ecohoya göre biçimsel değişkenleri kullanabilir. %i, %d, %s gibi gibi

echo bunları bastırmaz. yapamıyor daha küçük :D büyüyünce echoda yapabilir belki.

**printf format arguments**

format: %[-]m.nx → %32s, %7f

x → tipi için kullanılır. s, c, f, i, d vs vs vs

m → min, n → max değeridir.

s	String
c	Character
d	Decimal (integer) number
x	Hexadecimal number
o	Octal number
e	Exponential floating-point number
f	Fixed floating-point number
g	Compact floating-point number

## 9. hafta 2. ders

----- örnek

```
#!/bin/bash
printf "%32s %s\n" "File Name" "File Type" // 32s ile file name %s ile file type eşleşir.
for i in *; // bulunduğumuz dizindeki dosyaları sırayla alır.
do
    printf "%32s " "$i" // dosya adını yazar
    if [ -d "$i" ]; then echo "directory" // düzenli dosya ise directory yazar.
    elif [ -h "$i" ]; then echo "symbolic link" // sembolik linkse -l de aynı işlemi yapar.
    elif [ -f "$i" ]; then echo "file" // file ise
    else echo "unknown" // bilinmiyorsa
    fi; // if biter.
done // döngü biter.
```

%32s -> 32 birimlik boşluktan sağa doğru yazılır veri. 32 karakterse soldan başlamış gibi görünür.

Celal5 dosyası

```
#!/bin/bash
printf "%52s %s\n" "File Name" "File Type"
for i in *;
do
    printf "%52s " "$i" // print ilr dosya adı yazdırılır.
    if [ -d "$i" ]; then echo "directory" // echo ile dosya tipi yazılır ve alt satıra iner.
    elif [ -L "$i" ]; then echo "symbolic link"
    elif [ -f "$i" ]; then printf $i | wc -c
    else echo "unknown"
    fi
done
```

```
neogubuntu:~$ bash celal5
File Name File Type
awkkodu- 8
celal2 6
celal3 6
celal4 6
celal4- 7
celal5 6
cikktici 7
Desktop directory
Documents directory
dosya- 6
dosya1- 7
Downloads directory
Music directory
Pictures directory
Public directory
regular- 8
regular-c- 10
soru2.sh- 9
Templates directory
Videos directory
neogubuntu:~$ ^C
neogubuntu:~$
```

çıktı.

---- ödevler:

dfs

wc

argumanları yazan for döngüsü

celal5