

# Bash Programlamada Case Yapısı

siberoloji

Aralık, 2016

# İçindekiler

1	Format: . . . . .	2
2	Örnek: . . . . .	2
3	Açıklama: . . . . .	2

Bu yazıda sizlere, Bash Programlama içerisinde kullanılan CASE yapısından bahsetmek istiyorum. Bazen bash betikleri yazarken, if ve elif yapılarının kullanımını çok karışık hale gelebilir. Bu tür durumlarda, işlemleri daha basit hale getiren CASE yapısını kullanabilirsiniz. Case yapısında, belirttiğimiz değişkenin alabileceği değerlere göre farklı komutları yürütmemiz mümkündür.

Case yapısının kullanım formatı aşağıda görülmektedir.

## 1 Format:

```
1 case <değişken> in
2 <ihtimal 1>
3   <komutlar>
4 ;;
5 <ihtimal 2>
6   <komutlar>
7 ;;
8 esac
```

Format içerisinde görüldüğü gibi case, bir değişkenin değerine bakıyor. Bu değişkenin değerine göre hangi ihtimal uygunsa, o ihtimalin altındaki komutlar çalışıyor. Şimdi bunu bir örnekte görelim.

## 2 Örnek:

Aşağıdaki örnekte, df komutu yardımıyla diskimizdeki boş alan hesaplanmakta ve elde edilen değer bos\_alan isimli değişkene atanmaktadır. \$bos\_alan değişkeninin değerine göre case yapısı içerisinde farklı komutların çalışması istenmektedir.

Aşağıdaki kodları kopyalayıp, bir metin editöründe yeni dosya açarak içine yapıştırın. Dosyayı disk\_kullanim.sh olarak kaydedin. Komut satırından, dosyayı kaydettiğiniz klasöre giderek sudo chmod +x disk\_kullanim.sh komutuyla çalıştırılabilir hale getirin. Aynı klasörde, ./disk\_kullanim.sh komutuyla Bash betiğini çalıştırabilirsiniz.

```
1 #!/bin/bash
2 # df komutu yardımıyla diskteki boş alan miktarı tespit edilir.
3 # Kullanım oranına göre kullanıcıya tavsiyeler gösterilir.
4 bos_alan=$( df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%//' )
5
6 case $bos_alan in
7 [1-5]*)
8 echo Oldukça yeterli disk alanınız var. % $bos_alan
9 ;;
10 [6-7]*)
11 echo Yakında disk alanı sorunu yaşayabilirsiniz. % $bos_alan
12 ;;
13 8*)
14 echo Eski dosyalarınızı temizlemek isteyebilirsiniz. % $bos_alan
15 ;;
16 9*)
17 echo Çözmeniz gereken disk boşluk probleminiz var gibi. % $bos_alan
18 ;;
19 *)
20 echo Bir yanlışlık olmuş olmasın.
21 ;;
22 esac
```

## 3 Açıklama:

1. [1-5]\* kullanımı %10 ile %59 arası değerleri ifade etmektedir. Bu kullanımda df komutunun sonucunda elde

edilen rakamın ilk hanesi, köşeli parantez içinde kontrol edilmektedir. İkinci hanesi \* işaretiyle ne olursa olsun anlamına gelmektedir.

2. Case yapısında, kullanımı zorunlu olmamakla birlikte, \*) ifadesi de bulunmaktadır. Bu ifade, yukarıda yazılan ihtimaller haricinde bir ihtimal gerçekleştiğinde ne yapılacağını tanımlamaya yarar. Programcılık mantığıyla baktığınızda kullanmanız tavsiye edilmektedir.
3. esac ifadesi case ifadesinin tersinden yazılmış halidir ve CASE yapısının sonuna geldiğini belirtir.
4. Yukarıda kullanılan df komutunu parçalara ayırarak inceleyebilirsiniz. Bu sayede piping işleminin nasıl çalıştığını da görmüş olacaksınız.

```
1 df -h
2 df -h | awk '{ print $5 }'
3 df -h | awk '{ print $5 }' | sort -n
4 df -h | awk '{ print $5 }' | sort -n | tail -n 1
5 df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%//'
```