

## ÖN BİLGİ

Bu belge

- [https://www.syslogs.org/docs/Linux\\_Kabuklari\\_ve\\_Ozellikleri.doc](https://www.syslogs.org/docs/Linux_Kabuklari_ve_Ozellikleri.doc)

resmi adresindeki veya linkin kırık olması ihtimaline karşın alternatif olarak

- [https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber\\_Guvenlik\\_Teknik\\_Makaleler/Teori/BaskalarinaAitMakaleler/Syslogs.org\\_6\\_Linux\\_Kabuklari\\_ve\\_Ozellikleri.pdf](https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber_Guvenlik_Teknik_Makaleler/Teori/BaskalarinaAitMakaleler/Syslogs.org_6_Linux_Kabuklari_ve_Ozellikleri.pdf)

adresindeki makaleye çalışılarak elde edilen notlarımı kapsamaktadır. Bu çıkarılan notlar belgemde alıntılar ve/veya kişisel ilavelerim mevcuttur.

1)

Kabuklar kullanıcıların sistem ile etkileşimini sağlayan programlardır. Bu programlar sayesinde kullanıcı istediği komutları çalıştırabilir ve sonuçları görebilir. Sisteme girdiğiniz anda kabuk programı çalıştırılır. Linux sistemler ile birlikte en çok *bash* kabuğu kullanılmaktadır. Sistemde bulunan kabuklar */etc/shells* dosyasında da bulunmaktadır.

```
> cat /etc/shells
```

Output:

```
/bin/ash  
/bin/bash  
/bin/bash2  
/bin/bsh  
/bin/csh  
/bin/false  
/bin/sh  
/bin/tcsh
```

(Page 2)

2)

Sistemin hangi kabuğu kullandığını aşağıdaki komut ile öğrenebilirsiniz:

```
> echo $SHELL
```

Output:

```
/bin/bash
```

(Page 3)

3)

Kabuk programları genellikle */bin* dizini altında yer alır. Kullanıcının sisteme girerken hangi kabuğu kullanacağı */etc/passwd* dosyasında yer alır. Bunu değiştirmek için *chsh* (change shell) komutunu kullanabilirsiniz.

```
> chsh
```

```
Changing the login shell for hefese
```

```
Login Shell [/bin/sh]: /bin/bash // sh'dan bash'a değiştirilir.
```

(Page 3)

4)

Unix boruları ( Pipelines | ) en çok grep, wc ve sort gibi filtreleme komutları tarafından tercih edilmektedir.

(Page 8)

5)

**wc** komutu bir dosyadaki byte'ları, kelimeleri ya da satırları saymak için kullanılır.

- -c : Dosyadaki byte'ları say emri veren parametredir.
- -l : Dosyadaki satırları say emri veren parametredir.
- -w : Dosyadaki kelimeleri say emri veren parametredir.
- -L : Dosya içindeki en uzun satırın boyutunu gösterir.

deneme.txt

hasan

fatih

şimşek

> cat deneme.txt | wc -L

Output:

6

(Page 8)

6)

Metin dosyalarının satırlarının belirlenen kritere göre sıralı bir şekilde ekrana yansıtılması için **sort** komutu kullanılır.

deneme.txt

hasan

fatih

şimşek

```
> cat deneme.txt | sort
```

Output:

```
fatih
hasan
şimşek
```

(Page 9)

7)

Metin içinde arama yapmak için **grep** komutu kullanılır.

```
deneme.txt
```

```
hasan
fatih
şimşek
```

```
> cat deneme.txt | grep "a" // a karakterinin geçtiği satırlar yazdırılır.
```

Output:

```
hasan
fatih
```

Grep ile kullanılan parametreler aşağıdaki gibidir:

- l : Metin parçasının geçtiği satırların gösterilmesi yerine metin parçasının geçtiği dosyaların isimleri gösterilir. Çoğunlukla bir dizin içinde belirli bir kelimeyi içeren dosyaların bulunması işlevinde kullanılır.
- c : Metin parçasının geçtiği satırların gösterilmesi yerine metin parçasının bulunduğu satır sayısının gösterilmesini sağlar.
- i : Büyük – küçük harf ayrımını ortadan kaldırır.
- n : Metin parçasının geçtiği satırların gösterilmesiyle beraber metin parçasının bulunduğu satırın dosyadaki satır numarası da gösterilir.

(Page 9)

8)

### **Bash Kabuđu Özellikleri**

Bash'in kullanıcıya zaman kazandıran en önemli özelliklerinden birisi de dosya isimlerini tamamlamasıdır. Komut satırında tamamlanmamış bir komut veya dosya ismi yazdıktan sonra TAB tuşuna basılırsa satır tamamlanacaktır. Eğer komut satırındaki karakter kümesiyle başlayan birden fazla komut varsa bir sinyal sesi duyulacak ve sizden yeteri kadar karakteri yazmanızı bekleyecektir.

Komut satırındaki karakterler dosyayı veya komutu tanıtmaya yetmiyorsa, <TAB> tuşunun iki kez ard arda basılmasıyla ekrana mümkün olan tüm dosya isimleri getirilebilir.

Bash, komut satırında iken satırın kolayca değiştirilebilmesini sağlar. Böylece komut çalıştırılmadan önce birkaç tuş darbesiyle üzerinde değişiklik yapılabilir. Klavye üzerindeki alt ve üst yön tuşları, daha önce yazdığınız komutları görmeyi ve arasında seçim yapmanızı sağlar. Sağ ve sol yön tuşları ile seçtiğiniz komutun üzerinde değişiklikleri uygulayabilirsiniz.

(Page 20)

9)

### **Takma Adlar (alias)**

**alias** komutu ile bir komut veya komut kümesinin yerine bir isim tahsis edilebilir. İşleyişi bir makroya benzeyen bu komut yardımıyla uzun komutlar, daha kısa komutlarla tanımlanabilir. Bir alias komutu , anahtar kelimeyle başlar, ardından bir eşittir (=) işareti ve yerine kullanılacağı komut yazılır.

```
> alias liste='ls -al'
```

```
> liste
```

Output:

```
-rwxr-xr-x 9 root root 4096 Mar 6 15:59 deneme.txt
```

```
-rwxr-xr-x 9 root root 4096 Mar 6 15:59 dns.txt
```

```
-rwxr-xr-x 9 root root 4096 Mar 6 15:59 kaliMasaustu.txt
```

NOT: alias'ları /etc/profile dosyasına yerleştirirseniz sisteme her giriş-çıkışınızda yeniden aynısını tanımlamak zorunda kalmazsınız.

alias komutu tek başına kullanıldığında sistem tanımlı olan tüm lakaplar listelenecektir.

```
> alias
```

Output:

```
alias cp='cp -i'  
alias liste='ls -al'  
  
vs....
```

Tanımlı bir lakabı sistemden kaldırmak için **unalias** komutu kullanılır.

```
> unalias liste
```

(Page 21)

**10)**

### **Özel Kabuk (Shell) Değişkenleri**

Sisteme girdiğiniz zaman Linux size bir kabuk tahsis eder ve kabuk üzerinde değişkenler tanımlayabilmenize izin verir. Bir kabuk altında tanımlanan değişken o kabuğa özgüdür. Bir kabuk altında erişilebilen değişkene diğer bir kabuk erişemez. Bir çevre değişkeni belirtmek için export komutu kullanılır. Aşağıdaki örnekte *dosyam* isimli çevre değişkenini tanımlıyor.

```
> export dosyam="internet.txt"  
> echo $dosyam
```

Output:

```
internet.txt
```

printenv komutu ile ya da env komutu ile sistemde öntanımlı yapılmış olan ve sonradan tanımlanmış olan tüm değişkenler ekrana basılır:

```
> env
```

Output:

```
HOME=/home/mayasoft  
SHELL=/bin/bash  
LS_OPTIONS=--8bit --color=tty -F -b -T 0  
PS1=\h:\w\$\n  
PS2=>  
LOGNAME=mayasoft  
OSTYPE=Linux
```

Sisteme girdiğiniz anda tanımlanan yukarıdaki değişkenlerden bazısının açıklamaları aşağıda verilmiştir:

**PATH** : PATH değişkeninde bir komut yazıldığı anda sistem tarafından aranacak olan dizin listesi görüntülenir. Örnek olarak who komutu /usr/bin/ dizini altındadır ve bu bilgi PATH değişkeninde saklı tutulur. Kullanıcının yaptığı programları çalıştırabilmesi için PATH değişkeninde bulunduğu dizini de tanımlaması gerekir. PATH değişkeninde her dizin ':' işareti ile birbirinden ayrılır. Örnek olarak /usr/local/bin:/usr/bin dizisi sırasıyla /usr/local/bin ve /usr/bin dizinlerine karşılık gelir.

Kullanıcı PATH değişkenine yeni girdiler ekleyebilir. Aşağıdaki örnekte bunun nasıl uygulandığı görülüyor. Kullanıcı kısaca PATH değişkeninin sonuna kendisinin istediği patikayı veya patikaları ekler.

```
> echo $PATH                                # PATH degiskenini ekrana
bas
> export PATH=$PATH:/usr/net:~/bin          # PATH'e /usr/net patikasini ekle
```

**SHELL** : O an kullanılan kabuğun patika ismini verir. Kabuk programları genellikle /bin dizini altında tutulurlar. Her kabuğun patika ismi sistemdeki /etc/shells dosyasında bulunur.

```
> echo $SHELL
```

Output:

```
/bin/bash
```

**HOME** : Kullanıcının ev dizinini gösteren patikayı ekrana basar. Her kullanıcının ev dizini, sistemde hesap açılırken sistem görevlisi tarafından belirlenir. Linux Slackware dağıtımı altında bu dizin öntanımlı olarak /home 'dur. Aşağıdaki örnek kullanıcının ev dizinine ait patikayı ekrana basar.

```
> echo $HOME
```

Output:

```
/home/hefese
```

**LOGNAME** : Sistemdeki kullanıcı hesabının ismini tutar. Her kullanıcı için farklı bir değerde olur.

```
> echo $LOGNAME
```

Output:

```
hefese
```

11)

### **Sisteme Giriş ve Çıkış (Login) Dosyaları**

Sisteme girerken her giriş anında bash kabuğu tarafından çalıştırılan birtakım dosyalar bulunur. Bu dosyalar Bash Kabuğunun yapılandırma dosyalarıdır. Bunlardan bazılarını birlikte gözatalım.

#### **a. /etc/profile ve ~bash.profile Dosyaları**

Bu dosya bash kabuğunun başlangıç dosyasıdır. Sistemde bulunan tüm kullanıcılar sisteme oturum açtıklarında, ilk önce /etc/profile dosyasında bulunan komutları çalıştırır. Daha sonra ise kendi ev dizinlerinde bulunan .bash\_profile dosyası çalıştırılır. Bu dosyalar genellikle özel kabuk değişkenlerinin atanması için kullanılır. Örneğin .bash\_profile dosyasında özel değişkenlerden PATH ile ilgili tanımlamalar ekleyebiliriz. Varolan değerleri de değiştirebiliriz.

```
bash_profile
...
PATH=$PATH:$HOME/bin
export PATH
```

Diğer dosya olan /etc/profile ise yukarıda bahsettiğimiz gibi genel tüm kullanıcılar için geçerli olacak ayarları içermektedir. Tüm kullanıcılara tanımlamak istediğimiz bir kabuk değişkeni varsa bunu her kullanıcının dizinindeki .bash\_profile dosyasına yazmak yerine sadece /etc/profile dosyasına yazarak tüm kullanıcılara uygulanmasını sağlıyoruz. Örneğin bu dosyada tanımlanan histsize değerini

```
HISTSIZE=10000
```

yerine

```
HISTSIZE=500
```

yaparsak kullanıcıların bash\_history dosyasında tutacakları komut sayısı 500 olacaktır. Kullanıcıların kendi dizinlerinde bulunan .bash\_profile dosyası, /etc/profile dosyasındaki değerleri geçersiz kılacaktır.

#### **b. /etc/bashrc ve ~.bashrc Dosyaları**

Bu dosyalarda /etc/profile ve ~bash.profile dosyaları gibi kullanıcı sisteme girdiğinde otomatik olarak çalışan dosyalardır. **.bashrc** dosyası, her bash kabuğu veya alt kabuk çalıştırıldığı anda okunur. Her kabuk programı çalıştırılınca bir alt kabuğa ihtiyaç olduğundan bu durumda **.bashrc** dosyası da okunacak ve içerdiği değişkenler program çalışmadan önce tanımlanacaktır.

Genellikle aliaslar , umask değerleri **.bashrc ve /etc/bashrc** dosyasında tanımlanır.



(Page 24)

### c. **.bash\_logout** Dosyası

Sistemden çıkarken varsa çalıştırılan dosyanın ismi **.bash\_logout** 'tur. Bu dosya ile kullanıcı sistemi terkederken fazladan işlemler yapabilir. Aşağıdaki örnekteki **.bash\_logout** dosyasında kullanıcı sistemden her çıktığında ekran temizlenecektir.

```
~/bash_logout  
...  
clear
```

(Page 23-25)