

## Giriş

Bu bölümde Unix, Linux and BSD sistemleri için gelişmiş komutlar hakkında bilgi vereceğiz bu komutlar dosyaları silme, kopyalama, taşıma ve yeniden biçimlendirme gibi işlemleri gerçekleştirme için kullanılır. Bu bölümde ayrıca gelişmiş kabuk özellikleri olan otomatik tamamlama, joker karakterler, borular (es zamanlı işlemler) ve yeniden yönlendirme gibi konularda ele alınacaktır.

### Bu bölümde incelenecek komutlar

Komut	Kullanım amacı
<b>mv</b>	Dosya ya da dizinleri taşır veya yeniden adlandırır
<b>cp</b>	Dosya ve dizinleri kopyalar
<b>rm</b>	Dosyaları diler
<b>mkdir</b> <b>rmdir</b>	Dizin oluşturur, siler
<b>touch</b>	Dosyanın zaman etiketlerini günceller (erişim oluşturma, değiştirme zamanı vb.)
<b>ls</b>	Açık dosyaları listeler
<b>du</b>	Açık dosyalar hakkında bilgi görüntüler
<b>cksum</b>	Dosya bütünlüğü doğrulamasını görüntüler
<b>md5sum</b>	Dosyanın md5 doğrulama kodunu görüntüler
<b>ln</b>	Dosya yada dizinlere kısayol (link) oluşturur
<b>alias</b>	Komut satırında komutlar için kısayol oluşturur
<b>gzip</b> <b>gunzip</b>	Dosya sıkıştırır, açar
<b>split</b>	Büyük boyutlu dosyaları küçük boyutlu parçalara böler
<b>shred</b>	Güvenli dosya siler
<b>watch</b>	Periyodik olarak (belirli aralıklarla) belirtilen komutları çalıştırır
<b>env</b>	Ortam değişkenlerini görüntüler

### Otomatik

### tamamlama

Bir çok kabuk komut satırında otomatik tamamlamayı destekler. Otomatik tamamlama komut

satırında işlem yaparken komutların ve dosya yollarının kabuk tarafında bir kısmı yazıldıktan sonra tab tuşuna basıldığında tamamlamasıdır. Otomatik tamamlama aşağıdaki örnekte verildiği gibi kullanılır.

```
$ whe<TAB>
$ whereis
```

Yukarıda verilen örnekte whe yazıldıktan sonra Tab tuşuna basıldığında whereis komutuna ait diğer karakterleri yazmaya gerek kalmadan kabuk komutu whereis şeklinde tamamlayacaktır. Otomatik tamamlama aynı zamanda dosya yollarını tamamlamak içinde kullanılır. Örneğin ls -l /etc/en yazıp tab tuşuna bastığımız da otomatik tamamlama dosya yolunu aşağıdaki örnekte verildiği gibi /etc/environment

```
$ ls -l /etc/en<TAB>
$ ls -l /etc/environment
```

Otomatik tamamlama işlemi sırasında birden fazla eşleşme bulunduğu kabuk tüm sonuçları görüntüleyecektir. Bir sonraki örneğimizde ls -l /etc/host yazılıp Tab tuşuna basıldığında /etc dizinindeki tüm eşleşmeler listelenecektir

```
$ ls -l /etc/host<TAB>
host.conf hostname hosts.allow hosts.deny
```

#### Bilgi

Komut satırında tamamlama işlemine ek olarak bazı kabuklar klavyeden yukarı ok tuşuna basarak daha önce kullanılan kabuk komutlarının yeniden çağırılmasına imkan tanırlar

#### Joker karakterler 1 yada daha fazla

Joker karakterler 1 yada daha fazla metin elemanın yerine kullanılabilen karakterlerdir. Bu karakterler komut satırı üzerinde bir grup dosyanın listelenmesi silinmesi vb. Yığın işlemlerini yapılmasında yardımcı olurlar. Aşağıdaki tablo komut satırında kullanılacak farklı joker karakterlerin listesini vermektedir.

Komut	Kullanım amacı
*	Eşleşen birden çok karakter için kullanılır
?	Eşleşen 1 karakter için kullanılır
[abc]	Listelenen karakterlerden birini eşleşmesi için kullanılır
[a - c]	Karakter aralığındaki eşleşmeler için kullanılır
[! abc]	Listelenen karakterler dışındaki eşleşmeler için kullanılır
[!a - c]	Belirtilen karakter aralığı dışındaki eşleşmeler için kullanılır
{tacos, nachos}	Listelenen kelimelerden biri için eşleşme durumunda kullanılır

\* en basit ve en çok kullanılan joker karakterdir. Aşağıda ki örnek htc dizininde host ile başlayan tüm dosyaların listesinin nasıl alınacağını göstermektedir.

```
$ ls -l /etc/host*
```

Aşağıda verilen ilk örnekte allow veya deny kelimeleri ile biten tüm eşleşmeler listelenecektir. İkinci örnekte ise a harfi ile başlamayanlar enson örnekte ise 5 karakterli ilk 4 karakteri host olan tüm eşleşmeler listelenecektir.

```
$ ls -l /etc/hosts. {allow,deny}
$ ls -l /etc/hosts. [!a]*
$ ls -l /etc/hosts?
```

## Borular

Borular bir komutun çıkışını bir başka komutun girişine yönlendirmek için kullanılır. ls -l /etc yazılıp onaylandığında ls komutu etc dizininin içeriğini ekran boyunca hızlı bir şekilde listeler. Çıktılar ekrana sığmadığında ekran otomatik olarak kaydırılır. Şayet işlemler grafik ortamda değilse komut satırında gerçekleştiriliyorsa ekrandaki kaydırma çubuğu görüntülenmeyeceği için ekranın kayan kısmında kalan çıktılar görüntülenmeyecektir. Bunu engellemek için borulardan yararlanırız. Aynı komutu ls -l /etc | more şeklinde yazdığımızda sonuçlar bir ekran botyu listelenir listelemenin devam edebilmesi için kullanıcıdan bir tuşa basması istenir. Listelemeyi herhangi bir anda iptal etmek için "Q" basmak yeterlidir. Borularla beraber kullanılan bir diğer komut ise grep'tir. Bunu daha sonra ayrıntılarıyla inceleyeceğiz lakin kısaca belirtmek gerekirse grep komutu listeleme sonuçlarının kendisinden sonra verilen karakter grubuna göre filtrelenmesini sağlar örneğin ls -l /etc |grep host şeklindeki bir listelemede etc dizinindeki listeleme çıktısının içerisinde sadece host kelimesini barındıranlar görüntülenir.

## Yeniden yönlendirme

Bir komutun çıktısı bir başka konuma yada bir metin dosyasına yönlendirebilir. Yeniden yönlendirme işlemi ">" karakteri kullanılarak başlatılır.

```
$ date > date.txt
$ ls -l date.txt
```

Yukarıdaki örnekte date komutu tarih görüntüleme işlemini > karakteri nedeniyle ekrana değilse date.txt dosyasına yapmıştır. Şayet yönlendirme işlemi sırasında belirtilen dosya yoksa otomatik olarak oluşturulacaktır. Ancak dosya mevcut ise listeleme işlemi üzerine yapılacaktır. Dosya eski içeriğini kaybedecektir. Şayet listeleme işlemi var olan bir dosyanın sonuna dosya içeriğini silmeden eklemek istersek bu durumda >> karakterleri kullanılmalıdır. Aşağıdaki örnekte de date.txt dosyasının içeriğindeki dosya silinmeden snuç sonuna eklenmiştir.

```
$ date >> date.txt
```

İki	çeşit	çıkış	çıkış	vardır.
1.	Standart	çıkış-Standard	output	(STDOUT)
2.	Hata	çıkış-Error	output	(STDERR)

Aşağıda gösterildiği üzere standart çıktılar ve hata çıktıları belirtilen özel dosyalara yönlendirilebilir. Bu genellikle hata mesajlarını yakalarken faydalı olacaktır.

```
$ ls -l /NonExistantFile 1>ls.txt 2>lserror.txt
$ ls -l ls*
```

Bu örneğimizde de 1> standart çıkışı ve 2> standart hata çıkışını göstermektedir. İstenilen

dosya bulunamadığında hata mesajı lserror.txt dosyasına kaydedilecek aksi durumda hatanın oluşmadığı hertürlü çıkış ls.txt dosyasına yönlendirilecektir. Standart hata ve standart çıkışa ek olarak bir başka konumdan gelebn girdi(örneğin bir dosyadan) bir komuta yönlendirile bilir. Bu işlemde standart giriş olarak tanımlanmaktadır. Bir sonraki örneğimizde bir dosyanın içeriği mail komutuna girdi olarak gönderilmiştir.

\$ mail grepnick@gmail.com < ShoppingList.txt