

# Linux İşletim Sistemi

Son güncelleme : 12/2025

İçindekiler

► [Kabuk](#)

July  
17

## Tarihçe

Linux'un nasıl ortaya çıktığını öğrenmek için, 1969'a, Ken Thompson ve Dennis Ritchie'nin Bell Laboratuvarlarında UNIX işletim sistemini geliştirdikleri zamana dönelim. Daha sonra taşınabilirliği artırmak için C dilinde yeniden yazıldı ve sonunda yaygın olarak kullanılan bir işletim sistemi haline geldi.

Yaklaşık on yıl sonra, Richard Stallman, GNU (GNU, UNIX Değildir) projesi üzerinde çalışmaya başladı. Bu proje kapsamında Hurd adında bir GNU çekirdeği geliştirildi, ancak maalesef asla tamamlanmadı. Bunun sonucu olarak, özgür yazılım lisansı olan GNU Genel Kamu Lisansı (GPL) de oluşturuldu.

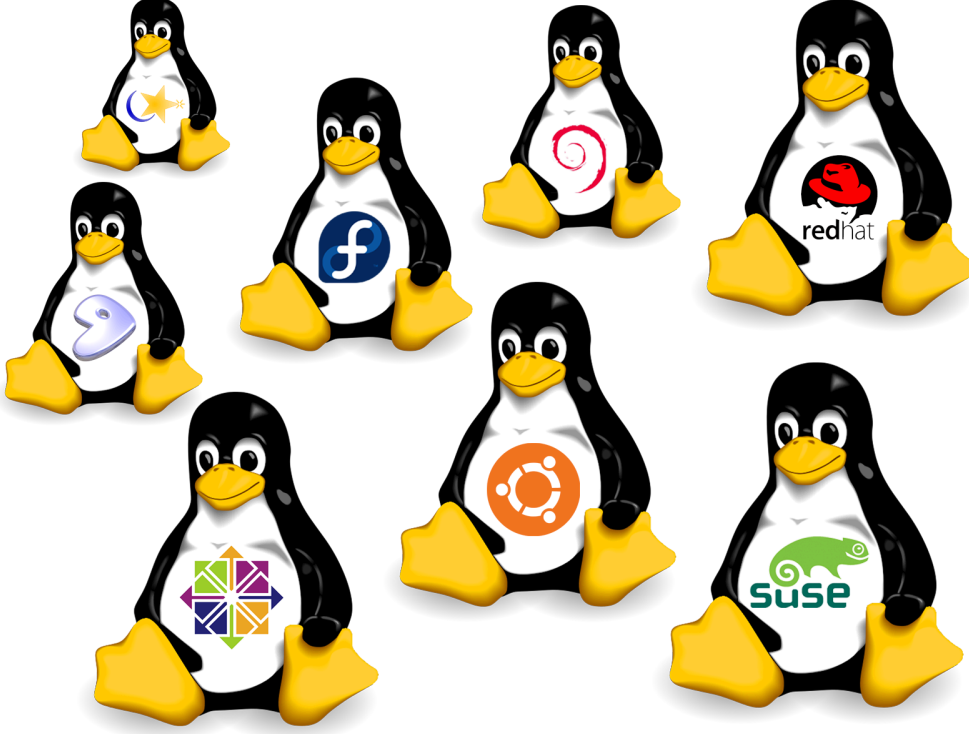
Çekirdek, işletim sisteminin en önemli parçasıdır. Donanımın yazılımla iletişim kurmasını sağlar. Biz kullanıcılar sistemde bulunan yazılımlar ile çekirdeğe emirler veririz, çekirdekde donanıma o işi yaptırır. Çekirdeğin sisteminizde olup biten her şeyi kontrol eder.

Bu dönemde BSD, MINIX vb. gibi diğer UNIX benzeri sistemler geliştirildi. Ancak, tüm bu UNIX benzeri sistemlerin ortak noktası, tek bir çekirdek eksikliği idi.

Ardından 1991'de Linus Torvalds adında genç bir adam, bugün bildiğimiz Linux çekirdeğini geliştirmeye başladı.

Sonuç olarak GNU yazılımları ve linux çekirdeğinin birleşimi ile ortaya açık kaynak, özgür bir işletim sistemi çıkmıştır.

# 🔥 Linux Dağıtımları



Bir Linux sistemi üç ana bölümden oluşur:

- **Donanım:** Bu, sisteminizin çalıştığı tüm donanımları, bellek, CPU, diskler vb. içerir.
- **Linux Çekirdeği:** Yukarıda belirttiğimiz gibi, çekirdek işletim sisteminin merkezidir. Donanımı yönetir ve sistemle nasıl etkileşim kuracağını söyler.
- **Kullanıcı Alanı:** Bu, bizler gibi kullanıcıların çeşitli yazımlar ile doğrudan sistemle etkileşim kuracağı yerdir.

Seçilebilecek birçok Linux dağıtımı vardır, sadece en popüler seçeneklere göz atacağız.

## ► Debian Dağıtımı

### Genel Bakış

Debian, tamamen özgür ve açık kaynaklı yazılımlardan oluşan bir işletim sistemidir. Geniş çapta bilinen ve 20 yılı aşkın süredir geliştirilmektedir. Kullanabileceğiniz üç ana sürümü vardır: Stable ( kararlı), Testing (test ) ve Unstable (kararsız).

### Sürümler

- **Stable:** Genel olarak kullanılması iyi olan bir sürüm. Sisteminizde kararlılık ve güvenlik öncelikliyse bu sürümü tercih edebilirsiniz.
- **Testing ve Unstable:** Sürekli güncelleme (rolling release) alan dallardır. Bu, bu dallardaki aşamalı değişikliklerin sonunda Stable sürümüne dahil olacağı anlamına gelir. Örneğin, Windows XP'den Windows 10'a yükseltme yapmak istiyorsanız, tam bir Windows 10 kurulumu yapmanız gerekir. Ancak Testing sürümünü kullanıyorsanız, tam bir kurulum yapmadan bir sonraki işletim sistemi sürümü olana kadar otomatik olarak güncellemeleri alacaksınız.

### 📦 Paket Yönetimi

Debian, kendi paket yönetim (APT) araçlarını kullanır. Her Linux dağıtımı paketleri farklı şekilde kurar, yönetir ve farklı paket yönetim araçları kullanır.

### Yapılandırılabilirlik

Debian en son güncellemeleri almasa da son derece kararlıdır. İyi bir "temel" işletim sistemi arıyorsanız, bu sizin için doğru tercih olabilir.

### Kullanım Alanları

Debian, her platform için genel olarak harika bir işletim sistemidir.

## ► Red Hat Enterprise Linux Dağıtımı

### Genel Bakış

Red Hat Enterprise Linux, genellikle RHEL olarak adlandırılır ve Red Hat tarafından geliştirilir. RHEL, ücretsiz yeniden dağıtımı kısıtlamak için katı kurallara sahiptir, ancak yine de kaynak kodunu ücretsiz olarak sağlar.

### Paket Yönetimi

RHEL, Debian'dan farklı bir paket yöneticisi olan RPM paket yöneticisini kullanır.

### Yapılandırılabilirlik

RHEL tabanlı işletim sistemleri, Debian tabanlı işletim sistemlerinden biraz farklılık gösterecek, özellikle paket yönetiminde daha belirgin olacaktır.

### Kullanım Alanları

Adından da anlaşılacağı gibi, çoğunlukla kurumsal alanda kullanılır, bu nedenle sağlam bir sunucu işletim sistemine ihtiyacınız varsa bu iyi bir tercih olacaktır.

## ► Ubuntu Dağıtımı

### Genel Bakış

Kişisel bilgisayarlar için en popüler Linux dağıtımlarından biri Ubuntu'dur. Ubuntu ayrıca varsayılan olarak kendi masaüstü ortamı yöneticisi Unity'yi yayınlar.

### Paket Yönetimi

Ubuntu, Canonical tarafından geliştirilen Debian tabanlı bir işletim sistemidir. Dolayısıyla temel olarak Debian paket yönetim sistemini kullanır.

### Yapılandırılabilirlik

Linux'a başlamak isteyen yeni başlayanlar için Ubuntu harika bir seçimdir. Ubuntu, kullanıcı dostu arayüzü ve geniş çapta benimsenilmesine yol açan kullanım kolaylığı sunar. Yaygın olarak kullanılmakta ve desteklenmektedir ve kullanılabilirlik açısından diğer işletim sistemleri gibi OSX ve Windows'a en çok benzerlik gösterir.

### Kullanım Alanları

Masaüstü, dizüstü bilgisayar ve sunucu dahil olmak üzere her platform için uygundur.

## ► Fedora Dağıtımı

### Genel Bakış

Red Hat tarafından desteklenen Fedora Projesi, açık kaynaklı ve ücretsiz yazılımları içeren, topluluk odaklı bir projedir. Red Hat Enterprise Linux, Fedora'dan dallanarak geliştirilir, bu nedenle Fedora'yı bir upstream RHEL işletim sistemi olarak düşünebilirsiniz. Sonuç olarak, Red Hat Enterprise Linux, kapsamlı test ve kalite güvencesinden sonra Fedora'dan güncellemeler alacaktır. Fedora'yı, Debian yerine Red Hat altyapısı kullanan bir Ubuntu eşdeğeri olarak düşünebilirsiniz.

### Paket Yönetimi

Fedora, Red Hat paket yöneticisini kullanır.

### Yapılandırılabilirlik

Red Hat tabanlı bir işletim sistemi kullanmak istiyorsanız, bu kullanıcı dostu bir versiyondur.

### Kullanım Alanları

Fedora, Red Hat tabanlı bir işletim sistemini fiyat etiketi olmadan kullanmak istiyorsanız harika bir seçimdir. Masaüstü ve dizüstü bilgisayarlar için önerilir.

## ► Linux Mint Dağıtımı

### Genel Bakış

Linux Mint, Ubuntu tabanlı bir işletim sistemidir. Ubuntu'nun yazılım depolarını kullanır, böylece her iki dağıtımda da aynı paketler kullanılabilir. Ubuntu'dan daha hafif bir dağıtım tercih ediyorsanız, Linux Mint ilginizi çekebilir.

### Paket Yönetimi

Linux Mint, Ubuntu tabanlı olduğundan Debian paket yöneticisini kullanır.

### Yapılandırılabilirlik

Harika bir kullanıcı arayüzü sunar, yeni başlayanlar için uygundur ve Ubuntu'dan daha az gereksiz yazılım içerir.

### Kullanım Alanları

Masaüstü ve dizüstü bilgisayarlar için uygundur.

## ► Arch Linux Dağıtımı

### Genel Bakış

Arch Linux, %100 topluluk tarafından yönetilen, hafif ve esnek bir Linux dağıtımdır. Debian'a benzer şekilde, Arch da sürekli güncelleme modelini (rolling release) kullanır, bu nedenle kademeli güncellemeler sonunda Stable (kararlı) sürüm haline gelir. Sistemi ve işlevlerini anlamak için gerçekten uygulamalı olarak öğrenmeniz gerekir, ancak karşılığında sisteminiz üzerinde tam ve eksiksiz kontrol elde edersiniz.

### Paket Yönetimi

Paketleri kurmak, güncellemek ve yönetmek için kendi paket yöneticisi Pacman'ı kullanır.

### Yapılandırılabilirlik

Hafif bir işletim sistemi istiyor ve Linux'u gerçekten anlamak istiyorsanız Arch'ı kullanın! Biraz öğrenme eğrisi olsa da, hardcore Linux kullanıcıları için harika bir seçimdir.

### Kullanım Alanları

Masaüstü ve dizüstü bilgisayarlar için uygundur. Ayrıca Raspberry Pi gibi küçük bir cihazınız varsa ve üzerine hafif bir işletim sistemi kurmanız gerekiyorsa, Arch'ı tercih edebilirsiniz.

## ► openSUSE Dağıtımı

### Genel Bakış

openSUSE Linux, tüm dünyadaki Özgür ve Açık Kaynaklı Yazılım topluluğunun bir parçası olarak açık, şeffaf ve dostça bir şekilde birlikte çalışan openSUSE Projesi tarafından yaratılmıştır. openSUSE, halen çalışmakta olan ikinci en eski Linux dağıtımdır ve ödüllü SUSE Linux Enterprise ürünleriyle taban sistemini paylaşır.

### Paket Yönetimi

RPM paket yöneticisini kullanır.

### Kullanılabilirlik

openSUSE, yeni bir Linux kullanıcısı için harika bir seçimdir. Kullanımı kolay bir grafiksel kurulum/yönetim uygulaması (YaST) ve düzenli bir temel sistem sunar, kurcalamaya kolay açıktır. openSUSE, ister fotoğraflarınız, videolarınız, müzikleriniz ister kodunuz olsun, İnternet'in keyfini virüslerden/casus yazılımlardan endişe duymadan çıkarmanız ve yaratıcılığınızı ortaya koymanız için ihtiyacınız olan her şeyi içerir.

## Kullanım Alanları

openSUSE Leap, masaüstü PC ve dizüstü bilgisayarda kullanıma tamamen uygundur.

## Kabuk (Shell)

Kabuk, temelde klavyenizden komutlarınızı alıp bunları işletim sistemine göndererek gerçekleştirilmesini sağlayan bir programdır. Daha önce bir GUI (grafiksel arayüz) kullandıysanız, "Terminal" veya "Konsol" gibi programları görmüşsünüzdür. Bunlar sizin için bir kabuk başlatan programlardır.

Bu belgede kabuk programı bash (Bourne Again SHell) kullanacağız, hemen hemen tüm Linux dağıtımları varsayılan olarak bash kabuğunu kullanır. Ksh, zsh, tsch gibi başka kabuklar da mevcuttur, ancak en çok kullanılan kabuk programı bash'dir.

Genel görünümü (prompt) aşağıdaki gibidir.

```
kullanıcı_adı@bilgisayar_adı:su_anki_dizin  
ali@pc:/home/ali $
```

Promptun sonunda \$ sembolünü gördünüz mü? Farklı kabukların farklı promptları olacaktır, bizim durumumuzda \$ sembolü Bash, Bourne veya Korn kabuğunu kullanan normal bir kullanıcı içindir, komutu yazarken bu sembolü eklemeyin.

Basit bir komut olan echo ile başlayalım. echo komutu, metin argümanlarını ekrana yazdırır.

```
$ echo Merhaba Dünya
```

## pwd (Print Working Directory / Çalışma Dizini Yazdır)

Linux'ta her şey bir dosyadır, Linux'u derinlemesine öğrendikçe bunu anlayacaksınız, ancak şimdilik sadece bunu aklınızda bulundurun. Her dosya, hiyerarşik bir dizin ağacında organize edilir. Dosya sistemindeki ilk dizin, kök dizin olarak adlandırılır. Kök dizinde, daha fazla klasör ve dosya depolayabileceğiniz birçok klasör ve dosya bulunur.

Bu dosya ve dizinlerin konumları yollar olarak adlandırılır.

Dosya sisteminde gezinmek, tıpkı gerçek hayatta olduğu gibi, nerede olduğunuzu ve nereye gideceğinizi bilmeniz yararlıdır. Nerede olduğunuzu görmek için pwd komutunu kullanabilirsiniz, bu komut "çalışma dizinini yazdır" anlamına gelir ve yalnızca hangi dizinde olduğunuzu gösterir, yolun kök dizinden geldiğini unutmayın.

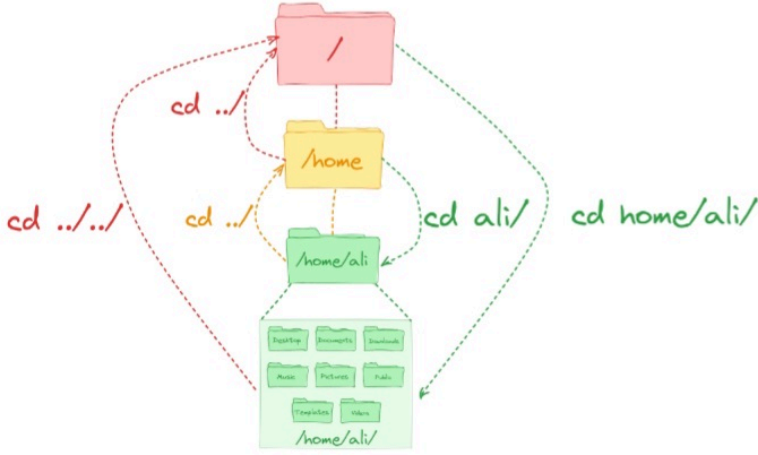
```
$ pwd
```

## cd (Change Directory / Dizin Değiştir)

Şimdi nerede olduğunuzu öğrendiğinize göre, dosya sisteminde biraz dolaşabileceğimize bakalım. Dosya sisteminde gezinmek için yolları kullanmamız gerektiğini unutmayın. Yol belirtmenin mutlak ve göreceli olmak üzere iki farklı yolu vardır.

**Mutlak yol:** Bu, kök dizinden itibaren olan yoldur. Kök dizin en önemli dizindir. Kök dizin genellikle bir eğik çizgi "/" olarak gösterilir. Yolunuz her zaman "/" ile başladığında, kök dizinden başladığınız anlamına gelir. Örneğin, `/home/ali/Masaüstü`

**Görelî yol:** Bu, dosya sistemindeki bulunduğunuz konumdan itibaren olan yoldur. Eğer `/home/ali/Dökümanlar` konumunda olsaydım ve Dökümanlar içinde vergiler adında bir dizine gitmek isteseydim, `/home/ali/Dökümanlar/vergiler` gibi kök dizinden tüm yolu belirtmeme gerek yok, bunun yerine sadece `vergiler/` dizinine gidebilirim.



Artık yolların nasıl çalıştığını bildiğimize göre, istediğimiz dizine geçmemize yardımcı olacak bir şeye ihtiyacımız var. Neyse ki, bunu yapmak için `cd` "dizin değiştir" komutu kullanılır.

```
$ cd /home/ali/Pictures
```

Böylece şimdi dizin konumumu `/home/ali/Pictures` olarak değiştirdim.

Şimdi bu dizinden Hawaii adında bir klasörüm var, şu şekilde o klasöre gidebilirim:

```
ali@pc:/home/ali/Pictures $ cd Hawaii
```

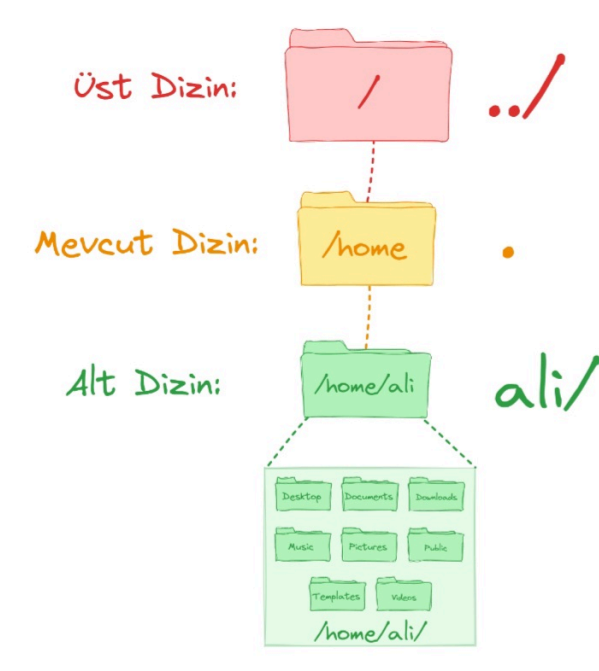
Sadece klasörün adını nasıl kullandığımı fark ettiniz mi? Çünkü zaten `/home/ali/Pictures` konumundaydım.

Her zaman mutlak ve görelî yollarla gezinmek oldukça yorucu olabilir, Neyse ki, size yardımcı olacak bazı kısayollar var.

- **.(geçerli dizin):** Şu anda bulunduğunuz dizindir.
- **..(üst dizin):** Sizi şu anki konumunuzun bir üst dizinine götürür.
- **~(ana dizin):** Bu dizin varsayılan olarak "ana dizininize" (`~/home/kullanıcı_adı`) gider.
- **- (önceki dizin):** Bu sizi az önce bulunduğunuz önceki dizine götürür.

Örnekler:

```
$ cd . # geçerli dizinde kal
$ cd .. # bir üst dizine git
$ cd ~ # ana dizine git
$ cd - # önceki dizine git
```



## ls (List Directories)

Dizin içeriklerini listelemek `ls` komutunu kullanabiliriz. `ls` komutu varsayılan olarak geçerli dizindeki dizinleri ve dosyaları listeler, ancak hangi dizinin dizinlerini listelemek istediğinizi belirtebilirsiniz.

```
$ ls
```

```
$ ls /home/pete
```

`ls` oldukça kullanışlı bir araçtır, aynı zamanda baktığınız dosyalar ve dizinler hakkında size ayrıntılı bilgi gösterir.

Ayrıca bir dizindeki tüm dosyaların görünmeyeceğini de unutmayın. Nokta ile başlayan dosya adları gizlidir, ancak bunları `ls` komutuyla görebilirsiniz ve `-a` (tümü için a) işareti ekleyebilirsiniz.

```
$ ls -a
```

Bir başka yararlı `ls` işareti daha vardır, `-l` (uzun) uzun formatta ayrıntılı bir dosya listesi gösterir. Bu size ayrıntılı bilgi gösterecektir, soldan başlayarak: dosya izinleri, bağlantı sayısı, sahip adı, sahip grubu, dosya boyutu, son değişiklik zaman damgası ve dosya/dizin adı.

```
$ ls -l
```

Şimdi daha fazlasını görelim:

```
pete@icebox:~$ ls -l

total 80

drwxr-x--- 7 pete penguingroup 4096 Nov 20 16:37 Desktop
drwxr-x--- 2 pete penguingroup 4096 Oct 19 10:46 Documents
drwxr-x--- 4 pete penguingroup 4096 Nov 20 09:30 Downloads
```

drwxr-x---	2	pete	penguingroup	4096	Oct	7	13:13	Music
drwxr-x---	2	pete	penguingroup	4096	Sep	21	14:02	Pictures
drwxr-x---	2	pete	penguingroup	4096	Jul	27	12:41	Public
drwxr-x---	2	pete	penguingroup	4096	Jul	27	12:41	Templates
drwxr-x---	2	pete	penguingroup	4096	Jul	27	12:41	Videos

Komutların daha fazla işlevsellik eklemek için işaretleri (veya argümanları veya seçenekleri, nasıl adlandırmak isterseniz) vardır. `-a` ve `-l` nasıl eklediğimizi gördük, her ikisini de birlikte `-la` ile ekleyebilirsiniz. İşaretlerin sırası hangi sırada gittiğini belirler, çoğu zaman bu gerçekten önemli değildir, bu nedenle `ls -al` yapabilirsiniz ve yine de çalışır.

```
$ ls -la
```

## touch

Şimdi bazı dosyaların nasıl oluşturulacağını öğrenelim. Çok basit bir yol, `touch` komutunu kullanmaktır. Touch, yeni boş dosyalar oluşturmanıza olanak tanır.

```
$ touch superdosyam
```

İşte bu kadar, yeni dosya!

Touch ayrıca mevcut dosya ve izinlerde zaman damgalarını değiştirmek için kullanılır. Deneyin, bir dosyada `ls -l` komutunu kullanın ve zaman damgasını not edin, ardından o dosyaya `touch` komutunu uygulayın, zaman damgası güncellenecektir.

yönlendirme ve metin editörleri gibi başka şeyleri içeren dosya oluşturmanın başka birçok yolu vardır, ancak bunlara Metin İşleme kursunda değineceğiz.

## file

Önceki derste `touch` komutunu öğrendik, biraz ona geri dönelim. Diğer işletim sistemlerinde (Windows gibi) gördüğümüz gibi, dosya adının standart isimlendirmeye uymadığını fark ettiniz mi? Normalde banana.jpeg adlı bir dosyadan ve bir JPEG resim dosyası beklemek normaldir.

Linux'ta, dosya adlarının dosyanın içeriğini temsil etmesi gerekmez. Aslında GIF olmayan `komik.gif` adında bir dosya oluşturabilirsiniz.

Bir dosyanın ne tür bir dosya olduğunu bulmak için `file` komutunu kullanabilirsiniz. Bu komut, dosyanın içeriğinin bir açıklamasını size gösterecektir.

```
$ file banana.jpg
```

## cat

Dosya yönetimi konusunu neredeyse bitirdik, ancak önce bir dosyanın nasıl okunacağını öğrenelim. Kullanılacak basit bir komut `cat` komutudur. Bu komut, concatenate (birleştirmek) kelimesinin kısaltmasıdır, yalnızca dosya içeriğini görüntüleyemez, aynı zamanda birden fazla dosyayı birleştirebilir ve size çıktıları gösterebilir.



```
$ cat dogfile birdfile
```

Ancak büyük dosyaları görüntülemek için pek uygun değildir ve yalnızca kısa içerikler için kullanılır. Bir sonraki derste ele alacağımız daha büyük metin dosyalarını görüntülemek için kullandığımız başka araçlar vardır.

## less

Basit çıktılarından daha büyük metin dosyaları görüntüleyecekseniz, "az daha fazladır" (ironik bir şekilde, aslında benzer bir şey yapan `more` adında bir komut vardır). Metin, sayfa sayfa görüntülenir, böylece bir metin dosyasında sayfa sayfa gezinebilirsiniz.

Devam edin ve bir dosyanın içeriğine `less` komutu ile bakın. `less` komutundayken, dosyada gezinmek için diğer klavye komutlarını kullanabilirsiniz.

```
$ less /home/pete/Dökümanlar/metin1
```

`less` içinde gezinmek için aşağıdaki komutları kullanın:

- **q** - `less` programından çıkıp komut satırına geri dönmek için kullanılır.
- **Sayfa yukarı, Sayfa aşağı, Yukarı ve Aşağı okları** - Ok tuşları ve sayfa tuşlarını kullanarak gezin.
- **g** - Metin dosyasının başına gitmek için kullanılır.
- **G** - Metin dosyasının sonuna gitmek için kullanılır.
- **/arama** - Metin belgesinin içinde belirli metinleri arayabilirsiniz. Aramak istediğiniz kelimelerin öncesine / işareti ekleyin.
- **h** - `less` programını kullanırken nasıl kullanılacağı hakkında biraz yardıma ihtiyacınız varsa, `h` komutunu kullanarak yardım ekranına erişebilirsiniz.

## less

Basit çıktılarından daha büyük metin dosyaları görüntüleyecekseniz, "az daha fazladır" (ironik bir şekilde, aslında benzer bir şey yapan `more` adında bir komut vardır). Metin, sayfa sayfa görüntülenir, böylece bir metin dosyasında sayfa sayfa gezinebilirsiniz.

Devam edin ve bir dosyanın içeriğine `less` komutu ile bakın. `less` komutundayken, dosyada gezinmek için diğer klavye komutlarını kullanabilirsiniz.

```
$ less /home/pete/Dökümanlar/metin1
```

`less` içinde gezinmek için aşağıdaki komutları kullanın:

- **q** - `less` programından çıkıp komut satırına geri dönmek için kullanılır.
- **Sayfa yukarı, Sayfa aşağı, Yukarı ve Aşağı okları** - Ok tuşları ve sayfa tuşlarını kullanarak gezin.
- **g** - Metin dosyasının başına gitmek için kullanılır.
- **G** - Metin dosyasının sonuna gitmek için kullanılır.
- **/arama** - Metin belgesinin içinde belirli metinleri arayabilirsiniz. Aramak istediğiniz kelimelerin öncesine / işareti ekleyin.
- **h** - `less` programını kullanırken nasıl kullanılacağı hakkında biraz yardıma ihtiyacınız varsa, `h` komutunu kullanarak yardım ekranına erişebilirsiniz.

# history

Kabukta, daha önce girdiğiniz komutların bir geçmişi vardır, aslında bu komutlara göz atabilirsiniz. Bu, daha önce kullandığınız bir komutu yeniden yazmadan bulup çalıştırmak istediğinizde oldukça faydalıdır.

- **Komut geçmişinizi görmek:**

```
$ history
```

- **Önceki komutu tekrar çalıştırmak:** Yukarı ok tuşuna basın.
- **Son komutu tekrar çalıştırmak:**

```
!!
```

Örneğin, `cat dosya1` yazdıysanız ve tekrar çalıştırmak istiyorsanız, sadece `!!` yazıp Enter'a basabilirsiniz. Bu, en son çalıştırdığınız komutu çalıştıracaktır.

- **Ters arama:** Ctrl-R tuşlarına birlikte basın. Bu, ters arama komutudur. Ctrl-R'ye basıp aradığınız komutun bir kısmını yazmaya başlarsanız, size eşleşmeleri gösterecektir. Ctrl-R tuşuna tekrar basarak bunlar arasında gezinebilirsiniz. Kullanmak istediğiniz komutu bulduktan sonra Enter tuşuna basmanız yeterlidir.
- **Ekranı temizleme:**

```
$ clear
```

- **Tab tuşu ile tamamlama:** Komut satırı ortamında en kullanışlı özelliklerden biri tab tuşu ile tamamlamadır. Bir komutun, dosyanın, dizinin vb. başlangıcını yazmaya başlarsanız ve Tab tuşuna basarsanız, arama yaptığınız dizinde bulunduğu şeye göre otomatik tamamlama yapacaktır. Örneğin, `chrome` komutunu çalıştırmaya çalışıyorsanız, `chr` yazıp Tab tuşuna basabilirsiniz, otomatik olarak `chrome` tamamlanacaktır.

# cp (Copy)

Dosyaları diğer işletim sistemlerinde kopyalayıp yapıştırmaya benzer şekilde, kabuk bize bunu yapmanın daha da basit bir yolunu sunar.

- **Tek bir dosya kopyalama:**

```
$ cp kopyalanacak_dosya hedef_konum
```

`kopyalanacak_dosya` kopyalamak istediğiniz dosyadır ve `hedef_konum` dosyayı kopyaladığınız yerdir.

Örnek:

```
$ cp superdosyam /home/pete/Dökümanlar/harika_dosyalar
```

Bu komut, `süperdosyam` adlı dosyayı `/home/pete/Dökümanlar/harika_dosyalar` dizinine kopyalar.

- **Çoklu dosya ve izin kopyalama:**

Birden fazla dosya ve dizini kopyalayabilirsiniz ve ayrıca joker karakterleri de kullanabilirsiniz. Joker karakter, arama daha fazla esneklik kazandıran bir desen tabanlı seçimi temsil eden bir karakterdir. Daha fazla esneklik için her komutta joker karakterleri kullanabilirsiniz.

- **Joker karakterler:**

- `*`: Tüm tek karakterleri veya herhangi bir dizeyi temsil eder.
- `?`: Tek bir karakteri temsil eder.
- `[]`: Köşeli parantez içinde yer alan herhangi bir karakteri temsil eder.

Örnek:

```
$ cp *.jpg /home/pete/Resimler
```

Bu komut, geçerli dizininizdeki tüm .jpg uzantılı dosyaları `Resimler` dizinine kopyalar.

- **Yinelenen dizin kopyalama:**

Yararlı bir komut, -r (recursive, yinelenen) işaretini kullanmaktır. Bu, bir dizin içindeki dosyaları ve dizinleri yinelemeli olarak kopyalar.

Dökümanlar dizinize birkaç dosya içeren bir dizini kopyalamaya çalışın. İşe yaramadı mı? Bunun nedeni, -r komutu ile birlikte içerideki dosyaları ve dizinleri de kopyalamanız gerektiğindendir.

Örnek:

```
$ cp -r Kabak /home/pete/Dökümanlar
```

Bir not: Aynı ada sahip bir dosyayı bir dizine kopyalarsanız, kopyaladığınız şey her neyse, var olan dosya üzerine yazılır. Bu, yanlışlıkla üzerine yazılmasını istemediğiniz bir dosyanız varsa iyi değildir. Dosyayı üzerine yazmadan önce size sormak için `-i` (interactive, etkileşimli) işaretini kullanabilirsiniz.

Örnek:

```
$ cp -i süperdosyam /home/pete/Resimler
```

## mv (Move)

`mv` komutu, dosyaları taşımak ve yeniden adlandırmak için kullanılır. `cp` komutuna benzer şekilde çalışır ancak dosyaları kopyalamak yerine taşır.

### Dosya Yeniden Adlandırma

Dosyaları şu şekilde yeniden adlandırabilirsiniz:

```
$ mv eski_dosya yeni_dosya
```

### Dosya Taşıma

Bir dosyayı farklı bir dizine şu şekilde taşıyabilirsiniz:

```
$ mv dosya2 /home/pete/Dökümanlar
```

### Çoklu Dosya Taşıma

Birden fazla dosyayı şu şekilde taşıyabilirsiniz:

```
$ mv dosya_1 dosya_2 /bir_dizin
```

### Dizin Yeniden Adlandırma

Dizinleri de şu şekilde yeniden adlandırabilirsiniz:

```
$ mv dizin1 dizin2
```

### Üzerine Yazma

Bir dosyayı veya dizini `mv` ile taşırsanız, aynı dizindeki herhangi bir şeyin üzerine yazar. Bu davranışı değiştirmek için `-i` işaretini kullanabilirsiniz.

```
$ mv -i dizin1 dizin2
```

### Yedek Oluşturma

Taşıma işlemini gerçekleştirmek ve üzerine yazmak istediğinizi varsayalım. Ayrıca o dosyanın bir yedeğini oluşturabilir ve eski sürümü yalnızca bir `~` ile yeniden adlandırabilirsiniz.

```
$ mv -b dizin1 dizin2
```

## mkdir (Make Directory)

Oluşturduğumuz tüm dosyaları depolamak için dizinlere ihtiyacımız olacak. `mkdir` (Make Directory) komutu bunun için kullanılır, var olmayan bir dizin oluşturur. Aynı anda birden fazla dizin bile oluşturabilirsiniz.

```
$ mkdir kitaplar resimler
```

Ayrıca `-p` (parent, üst dizin) işareti ile aynı anda alt dizinler de oluşturabilirsiniz.

```
$ mkdir -p kitaplar/hemingway/favoriler
```

## mkdir (Make Directory)

Oluşturduğumuz tüm dosyaları depolamak için dizinlere ihtiyacımız olacak. `mkdir` (Make Directory) komutu bunun için kullanılır, var olmayan bir dizin oluşturur. Aynı anda birden fazla dizin bile oluşturabilirsiniz.

```
$ mkdir kitaplar resimler
```

Ayrıca `-p` (parent, üst dizin) işareti ile aynı anda alt dizinler de oluşturabilirsiniz.

```
$ mkdir -p kitaplar/hemingway/favoriler
```

## rm (Remove)

Birçok dosya oluşturduk, şimdi bazılarını silelim. Dosyaları silmek için `rm` komutunu kullanabilirsiniz. `rm` (remove) komutu, dosya ve dizinleri silmek için kullanılır.

```
$ rm dosya1
```

**Dikkat:** `rm` komutunu kullanırken dikkatli olun. Silinen dosyaları geri getirmek için sihirli bir çöp kutusu yoktur. Silindikten sonra sonsuza kadar kaybolurlar. Bu yüzden dikkatli olun.

Neyse ki, ortalama bir kullanıcının önemli dosyaları kolayca silmesini önlemek için bazı güvenlik önlemleri alınmıştır. Yazma korumalı dosyalar, silinmeden önce sizden onay ister. Bir dizin yazma korumalıysa, kolayca silinemez.

Artık bunların hiçbirini önemsemiyorsanız, kesinlikle bir sürü dosya silebilirsiniz.

- **-f** veya **force** seçeneği, `rm` komutuna tüm dosyaları silmesini (yazma korumalı olsalar bile) kullanıcıya sormadan silmesini söyler (tabii ki gerekli izinlere sahipseniz).

```
$ rm -f dosya1
```

- Diğer birçok komutta olduğu gibi `-i` işaretini eklemek, dosyaları veya dizinleri gerçekten silmek isteyip istemediğinizi soran bir uyarı görüntüler.

```
$ rm -i dosya
```

- Varsayılan olarak `rm` ile bir dizini silemezsiniz. İçerdiği tüm dosyaları ve alt dizinleri silmek için `-r` (recursive, yinelemeli) işaretini eklemeniz gerekir.

```
$ rm -r dizin
```

- `rmdir` komutuyla boş bir dizini silebilirsiniz.

```
$ rmdir dizin
```

## find

Sistemde bu kadar çok dosya varken, belirli bir dosyayı bulmaya çalışmak biraz zor olabilir. Neyse ki, bunun için kullanabileceğimiz bir komut var: `find`

`find` komutunu kullanarak hangi dizinde arama yapacağınızı ve ne aradığınızı belirtmeniz gerekir. Bu örnekte, `puppies.jpg` adlı bir dosya aramaya çalışıyoruz.

- **Dosya adına göre arama:**

```
$ find /home -name puppies.jpg
```

- **Dosya türüne göre arama:**

Aradığınız dosyanın türünü de belirtebilirsiniz. Örneğin, bir klasör aramak için `-type d` seçeneğini kullanabilirsiniz.

```
$ find /home -type d -name MyFolder
```

Bu komutta, aradığımız dosya türünü `(d)` (dizin) olarak ayarladık ve yine `MyFolder` adına göre arama yapıyoruz.

**Önemli Not:** `find` komutu yalnızca aradığınız dizinde arama yapmaz, aynı zamanda o dizinin içinde olabilecek alt dizinlerin içine de bakar.

## help

Linux, bir komutu nasıl kullanacağınızı öğrenmenize veya bir komut için hangi işaretlerin (flag) mevcut olduğunu denetlemenize yardımcı olacak harika yerleşik araçlara sahiptir.

- **help komutu:**

`help` komutu, diğer bash komutları (`echo`, `logout`, `pwd`, vb.) hakkında yardım sağlayan yerleşik bir bash komutudur. Kullanmak istediğiniz komut hakkında bilgi almak için aşağıdaki gibi yazabilirsiniz:

```
$ help echo
```

Bu komut, `echo` komutunu çalıştırmak istediğinizde kullanabileceğiniz açıklamayı ve seçenekleri size gösterecektir.

- **--help seçeneği:**

Diğer yürütülebilir programlar için, genellikle `--help` veya benzeri bir seçenek kullanmak bir gelenektir.

```
$ echo --help
```

Yürütülebilir dosyaları dağıtan tüm geliştiriciler bu standarda uymayacaktır, ancak bir program hakkında yardım bulmak için muhtemelen en iyi seçeneğinizdir.

## man

Linux programları hakkında daha fazla bilgi edinmek istiyorsanız, `man` komutunu kullanarak man sayfalarına erişebilirsiniz. Man sayfaları, komutların ayrıntılı açıklamalarını, seçeneklerini ve kullanım örneklerini içerir.

Örneğin, `ls` komutu hakkında daha fazla bilgi edinmek için:

```
$ man ls
```

Man sayfaları bazen teknik olabilir, ancak bir komutun tüm özelliklerini öğrenmek için harika bir kaynaktır.

## whatis

Bir komutun ne işe yaradığından şüphe duyuyorsanız, `whatis` komutunu kullanarak kısa bir açıklama alabilirsiniz. `whatis` komutu, komut satırı programları hakkında özlü bilgiler sağlar.

### Kullanım:

```
$ whatis komut_adı
```

### Örnek:

```
$ whatis cat
```

Bu örnekte, `cat` komutunun ne işe yaradığı hakkında kısa bir açıklama görürsünüz. Açıklama, komutun man sayfasından alınır.

## alias

Uzun komutları yazmaktan yoruldunuz mu? Belki de aynı komutu tekrar tekrar mı kullanıyorsunuz?

Linux size, sık kullandığınız komutlar için takma adlar oluşturma imkanı sunar. Bu takma adlar sayesinde komutları daha kısa ve yazması daha kolay hale getirebilirsiniz.

### Takma Ad Oluşturma:

Bir takma ad oluşturmak için `alias` komutunu kullanın. Takma adın ismini istediğiniz gibi seçebilirsiniz, ardından eşittir işaretini (=) yazın ve takma adın hangi komutu çalıştırmasını istediğinizi belirtin.

Örneğin, `ls -la` komutunu sık sık kullanıyorsanız, bunun için `foobar` adında bir takma ad oluşturabilirsiniz:

```
$ alias foobar='ls -la'
```

Bundan sonra, `ls -la` yazmak yerine `foobar` yazabilirsiniz. `foobar` yazdığınızda, aslında `ls -la` komutu çalıştırılacaktır.

### Kalıcı Takma Adlar:

Bu komutla oluşturduğunuz takma adlar, terminal oturumunu kapattığınızda kaybolur. Eğer takma adın sürekli olarak kullanılabilir olmasını istiyorsanız, onu konfigürasyon dosyalarından birine eklemeniz gerekir.

Genellikle bash kullanıcıları için takma adlar `.bashrc` dosyasına eklenir. Bu dosyayı bir metin editörü ile açıp, takma adınızı şu şekilde ekleyebilirsiniz:

```
alias foobar='ls -la'
```

Daha sonra dosyayı kaydedin. Artık terminal oturumunu kapatıp açsanız bile `foobar` takma adını kullanmaya devam edebilirsiniz.

### Takma Ad Silme:

Oluşturduğunuz bir takma ada artık ihtiyacınız yoksa, `unalias` komutunu kullanarak silebilirsiniz.

```
$ unalias foobar
```

Bu komuttan sonra `foobar` takma adını kullanamazsınız.

## exit

Linux komutlarının temellerini öğrenmekte harika bir iş çıkardınız! Bu sadece başlangıçtı, emeklemeyi öğrendiniz, şimdi bir sonraki aşamaya geçebilirsiniz. Gelecek derslerde size yürümeyi öğreteceğim.

Şimdilik kendinizi arkanıza yaslanıp biraz dinlenebilirsiniz. Shell'den çıkmak için aşağıdaki komutlardan birini kullanabilirsiniz:

- `exit`: Bu en yaygın çıkış komutudur.
- `logout`: `exit` komutuyla aynı işlevi görür.

Eğer terminal emülatörü kullanıyorsanız, pencereyi kapatarak da çıkabilirsiniz.

Linux dünyasında başarılar!



# YAPIM AŞAMASINDA

Bu bölüm henüz tamamlanmamıştır,  
İçerik üzerinde aktif olarak çalışılıyor...

