



Way to Reinvent Yourself

Linux nedir? & Temel Linux Komutları

Linux, Linux çekirdeği bazlı açık kaynaklı **Unix** işletim sistemleri ailesidir.

Buna Linux bazlı en popüler **Ubuntu**, **Fedora**, **Mint**, **Debian** vb. gibi sistemler de dahildir. Kesin olmak gerekirse, bu sistemlere dağıtım adı verilmektedir.

1991'de piyasaya sürüldüğünden beri açık kaynaklı olmasından dolayı popülerliği artmaya devam etmiştir. Herkes Linux'u özgürce düzenleyebilir ve kendisi yeniden dağıtabilir.

Bir Linux işletim sistemini kullanırken **shell** (**kabuk**) da kullanmanız gerekir. Shell, işletim sisteminin hizmetlerine erişim veren bir **arayüzdür**. Bir kullanıcı oturum açtığında veya bir terminali başlattığında shell başlatılır. Shell, kullanıcı tarafından okunabilen komutları alır ve bunları **kernel** (**çekirdek**) için uygun komutlara dönüştürür. Birçok Linux dağıtımı kullanıcılarına kullanım kolaylığı sağlamak için shell olarak **grafiksel kullanıcı arayüzü** (**GUI**) kullanmaktadır.

Bununla beraber, bir **komut satırı arayüzü** (**CLI**) kullanmak daha fazla önerilir çünkü daha güçlü ve etkilidir. **GUI** üzerinden çok aşamalı işlemler gerektiren görevler **CLI**'a birkaç komut girerek saniyeler içinde yapılabilir.

Ayrıca eğer Linux kullanmayı düşünüyorsanız **temel Linux komutları** ve kullanımları nelerdir bilmeniz gerekecektir. Bu dokümanda temel Linux komutlarını anlatacağız.

Linux komutlarına geçmeden önce bazı ipuçları:

Eğer terminal çok fazla geçmiş komutlarla dolduysa **clear** komutunu girerek terminali temizleyin.

Yazdığınızı otomatik olarak tamamlamak için **TAB** tuşuna basın. Örneğin, eğer Documents yazmak istiyorsanız komutu yazmaya başlayın (**cd Docu** ile başlayın ve daha sonra ise TAB tuşuna basın) ve terminal gerisini dolduracak ve **cd Documents** olarak size gösterecektir.

Ctrl+C ve **Ctrl+Z** ise şu anda çalışan herhangi bir komutu durdurmak için kullanılır. Ctrl+C komutu durduracak ve sonlandıracaktır, Ctrl+Z ise komutu sadece durduracaktır.

Eğer terminali **Ctrl+S** tuşlarına basarak terminali dondurursanız **Ctrl+Q** tuşlarına basarak bunu geri alabilirsiniz.

Ctrl+A tuşlarıysa sizi satırın başına alırken **Ctrl+E** ise sizi sona götürür.

Tek bir komutta birden fazla komutu çalıştırabilirsiniz. Bunun için ";" kullanarak onları ayırın. Örneğin **Komut1; Komut2; Komut3**.

Veya **&&** kullanarak ikinci veya üçüncü komutun sadece bir önceki komut başarılı olursa çalışmasını sağlarsınız.

Temel Linux Komutları:

Linux tabanlı işletim sistemleri çoğunlukla terminal üzerinden kontrol edilmektedir.

Linux, GNU standardında açık kaynak kodlu, özgür bir yazılım olduğundan çeşitli geliştiriciler tarafından farklı komutlar geliştirilmiştir. Bu komutlar [disk işlemleri](#), [ağ işlemleri](#), [dosya paylaşım işlemleri](#) vb. gruplara ayrılabilir.

Sıklıkla kullanılan Linux komutları da [temel komutlar](#) olarak adlandırılabilir.

Komutlar ile ilgili bilgi almak için **<komut> --help** parametresi kullanıyoruz.

Komut sonuçlarının düzgün okunabilmesi için **<komut> -h** parametresi kullanıyoruz.

1. sudo

“**SuperUser Do**” kısaltması olan bu komut yönetici veya kök izinlerini gerektiren görevleri gerçekleştirmenizi sağlayan popüler bir Linux komutudur. Ancak, bu komut günlük kullanım için önerilmez çünkü yanlış bir şey yaparsanız hata oluşması oldukça kolaydır.

Sudo kullanıldığında sistem, kullanıcıların parolayla kimlik doğrulaması yapmalarını ister. Ardından, Linux sistemi [tracker](#) olarak bir [timestamp](#) kaydı yapar. Varsayılan olarak, her root user 15 dakika/oturum için **sudo** komutlarını çalıştırabilir.

2. pwd

pwd komutu hangi dizinde (klasörde) olduğunuzu bulmak için kullanılır. Komut size kesin ve tam bir konum ([path](#)) verecektir. Konum genellikle bir bölme işaretiyle (/) başlar: [/home/username](#)

3. cd

Linux dosya sisteminde gezinmek, dizinler arası geçiş yapmak için **cd** komutu kullanılır.

Bulduğunuz mevcut dizine ([working directory](#)) bağlı olarak ya tam konum ([full path](#)) ya da dizinin adını ([directory name](#)) gerektirir.

Diyelim ki [/home/username/Documents](#) dizinindeyiz ve [Documents](#)’ın alt dizini olan [Python](#)’a gitmek istiyorsunuz. Bunu yapmak için gerekli komut:

cd Python

Diyelim ki tamamıyla yeni bir dizine, mesela [/home/teamwork/Linux](#)’e geçmek istiyorsunuz. Bu durumda **cd** ve daha sonra da dizinin tam konumunu girmeniz gerekmektedir:

cd /home/teamwork/Linux

Eğer daha hızlı konum değiştirmek istiyorsanız bazı kısayollar mevcuttur:

- **cd “dir name”** veya **cd dir\ name** içinde boşluk içeren bir dizine gitmek için.
- **cd ..** bir üst dizine gider.
- **cd-** bir önceki dizine gider.
- **cd** veya **cd ~** direkt olarak home directory’e (ana klasöre) gider.
- **cd /** root director’e gider.
- **cd /etc** ana dizindeki etc dizinine gider.

Ek olarak, Linux kabuğu büyük küçük harfe duyarlıdır. Bu yüzden dizinin adını olduğu gibi girmelisiniz.

4. ls

ls komutu bir dizinin içeriğini görmek için kullanılır. Varsayılan olarak, bu komut mevcut dizinin içeriğini görüntüleyecektir.

Eğer başka dizinlerin içeriğini görmek istiyorsanız **ls** ve dizinin konumunu girmelisiniz. Örneğin, `/home/username/Documents`'in içeriğini görmek için :

ls /home/username/Documents komutunu girin.

ls komutu ile kullanabileceğiniz bazı varyasyonlar mevcuttur:

- **ls -R** alt dizinlerdeki bütün dosyaları da listeleyecektir
- **ls -a** bütün gizli dosyaları gösterecektir
- **ls -al** izinler, boyut, sahip gibi detaylı bilgilerle dosyaları ve dizinleri listeleyecektir.
- **ls -lh** dosya boyutunu MB, GB, TB gibi formatlarda gösterir.

5. cp

cp komutunu mevcut dizinden dosyaları veya dizinleri ve içeriklerini kopyalamak için kullanılır:

cp ornek.jpg /home/kullaniciadi/Pictures

komutu `ornek.jpg` dosyasının `Pictures` dizininde bir kopyasını oluşturur.

Birden fazla dosyayı bir dizine kopyalamak için dosya adlarını ve hedef dizini girin:

cp filename1.txt filename2.txt filename3.txt /home/username/Documents

komutu 3 ayrı dosyayı `Documents` dizinine kopyalar.

Bir dosyanın içeriğini aynı dizindeki yeni bir dosyaya kopyalamak için **cp** ve ardından `kaynak dosyayı ve hedef dosyayı` girin:

cp filename1.txt filename2.txt komutu `filename1.txt` adlı dosyanın içeriğini aynı dizindeki yeni bir dosyaya (`filename2.txt`) kopyalar.

Tüm dizini kopyalamak için kaynak dizini yazmadan önce **-R** ve ardından hedef dizini girin:

cp -R /home/username/Documents /home/username/Documents_backup

6. mv

Bu komutun ana kullanımı dosya ve dizinleri taşımak ve aynı zamanda yeniden adlandırmaktır. Yeni isim vermezseniz aynı adla taşımış olursunuz:

Bu komutun kullanımı da **cp** komutuna oldukça benzerdir. Önce **mv**, daha sonra dosyanın adı ve hedef dizini girin:

mv dosya.txt /home/username/Documents

Dosyaları yeniden adlandırmak için:

mv dosya.txt file.txt

`dosya.txt` dosyasının ismini `file.txt` olarak değiştirdik.

Dosyaları bir klasörden başka bir klasöre ismini değiştirerek aktarmak için:

mv c:/users/edwin/deneme/ornek.txt c:/users/edwin/yeniklasör/yeni.txt

`c:/users/edwin/deneme/` klasöründeki `ornek.txt` dosyasını `c:/users/edwin/yeniklasör` klasörüne `yeni.txt` ismiyle kopyaladık.

7. mkdir

mkdir komutunu yeni bir dizin oluşturmak için kullanılır:

mkdir python komutu, **python** adlı yeni bir dizin oluşturacaktır.

Başka bir dizin içinde yeni bir dizin oluşturmak için :

mkdir python/assignments

python dizini içinde **assignments** adlı bir dizin oluşturduk.

Mevcut iki dizin arasında yeni bir dizin oluşturmak için **p** seçeneğini kullanın.

mkdir -p python/2022/assignments

yeni "2022" dizinini oluşturacaktır.

Yeni bir klasör oluştururken ekrana bilgi mesajı yazmak için **-v** kullanılır:

mkdir -v html ekrana **html** klasörün oluşturulduğu mesajını yazdırarak bu klasörü oluşturur.

8. rmdir

Boş bir dizini kalıcı olarak silmek için **rmdir** komutunu kullanılır:

rmdir python/

Boş olan **python** klasörünü sildik.

rmdir -p python/assignments

Boş olan alt klasör **assignments**'ı **python** ana klasöründen sildik.

9. rm

rm komutu bir klasör içindeki dosyaları diler. Eğer içindekilerle birlikte dizini de silmek istiyorsanız **rm -r** kullanın.

Not: Bu komutu gerçekleştiren kullanıcının yazma izinlerine sahip olması gerekmektedir. Bu komuta fazlasıyla dikkat edin ve hangi dizinde olduğunuzu kontrol edin. Bu her şeyi silecektir ve geri dönüşü yoktur.

rm -r python

python klasörünü içindekilerle birlikte sildik.

Birden fazla dosya silmek için:

rm filename1 filename2 filename3

bir dosyayı silmeden önce sistem onayını istemesi için:

rm -i -r python

10. touch

Command line üzerinden yeni boş dosyalar oluşturmaya yarar.

python dizini altında **deneme.py** adlı bir python dosyası oluşturmak için:

touch /home/users/python/deneme.py

girilen dosya yolu içinde **deneme.py** adlı bir dosya oluşturduk.

touch deneme.py

bulduğumuz **working directory** içinde **deneme.py** adlı bir dosya oluşturduk.

11. vim

vim komutu, 90'lerden beri kullanılan ücretsiz ve açık kaynaklı bir terminal metin düzenleyicisidir. Verimli tuş bağlantıları kullanarak düz metin dosyalarını düzenlemenizi sağlar.

Vim'i başlatmak için şunu yazın: **vim**

Ekрана bir metin düzenleyici açılacaktır. Metninizi yazdıktan sonra **ESC** tuşuna basarsanız **command moduna** geçer. Ve sonra **:w** + **Enter** ile yazdıklarınızı kaydeder. **:q** + **Enter** ile editörden çıkabilirsiniz.

12. cat

cat Linux'da sıklıkla kullanılan komutlardan biridir. Standart çıktıdaki (output) bir dosyanın içeriğini görmek için kullanılır. Bu komutu çalıştırmak için önce **cat** daha sonra da dosya adını ve uzantısını girin:

cat dosya.txt

yeni bir dosya oluşturmak için:

cat > dosyaadi

file1 ve **file2** adındaki iki dosyayı birleştirip sonucu **file3** adında yeni bir dosyaya kaydetmek için:

cat file1 file2 > file3

13. echo

Bu komut bir dosyaya veri taşımak için kullanılır. Örneğin “Merhaba, ben Clarusway’de python öğreniyorum” yazısını **mesaj.txt** adlı dosyaya eklemek için:

echo Merhaba, ben Clarusway’de python öğreniyorum >> mesaj.txt .

-n, -e, \a, \b, \c, -E ile birlikte kullanımları vardır.

14. locate

Windows'daki search komutu gibi bir dosyanın konumunu bulmak için kullanabilirsiniz. **-i** seçeneği ile kullanırsanız case sensitive özelliğini (büyük-küçük harfe duyarlı) kaldıracaktır.

İki ya da daha fazla kelime içeren bir dosyayı aramak için yıldız işareti (*) kullanılır : **locate -i deneme*ornek**

“deneme” ve “ornek” kelimelerini içeren bütün dosyaları büyük veya küçük karakter fark etmeksizin arattırdık.

15. find

locate komutuna benzer olarak **find** komutu da dosya aramak için kullanılır. Farkı ise belirli bir dizindeki dosyaları bulmak ve sonraki işlemleri uygulamak için kullanılmasıdır:

find /home/ -name notlar.txt

notlar.txt adlı bir dosyayı **home** dizini ve onun alt dizinlerinde aradık.

find -name notlar.txt

notlar.txt adlı dosyayı mevcut dizinde bulmak için:

Dizin aramak için:

```
find ./ -type d -name python
```

python dizinini arattık.

16. grep

Belirli bir dosyanın içindeki tüm metinde (text'te) bir sözcük arama yapmak için kullanılır:

```
grep degisken deneme.txt
```

deneme.txt dosyasında degisken kelimesini aradık. Ekranda bu kelimeyi içeren satırlar gösterilecektir.

17. diff

difference'in kısaltılmışı olan **diff** komutu iki dosyanın içeriğini satır satır karşılaştırır. Dosyaları analiz ettikten sonra uymayan satırları sonuç olarak verecektir. Programcılar bu komutu programda değişiklik yapmak için sıklıkla bütün kaynak kodunu yeniden yazmak yerine kullanır:

```
diff dosya.txt dosya_guncel.txt
```

- c** iki dosya arasındaki farkı bir içerik (context) formunda görüntüler.
- u** yedek bilgi olmadan output'u görüntüler.
- i** diff komutunu case sensitive (büyük-küçük harfe duyarlı) yapar.

18. head

Herhangi bir metin dosyasının ilk 10 satırını görüntüler. Ancak eklenecek opsiyon ile bu rakamı değiştirebilirsiniz:

```
head -n 5 deneme.txt
```

deneme.txt dosyasının ilk beş satırını görüntüledik.

Dosya adını belirten başlığı yazdırmamak için **-q** veya **-quiet** parametresi kullanılır:

```
head -n 5 -q (or -quiet) deneme.txt
```

19. tail

head komutuna benzer bir fonksiyona sahiptir. İlk satırları göstermek yerine bir metin dosyasının son on satırını görüntüler.

20. chmod

Dosyaların ve dizinlerin okuma (**read**), yazma (**write**) ve çalıştırma (**execute**) izinlerini değiştirmek için kullanılan önemli bir komuttur.

read'in sayısal değeri 4; **write**'in sayısal değeri : 2; **execute**'un sayısal değeri :1 dir.

Linux'ta, her dosya üç kullanıcı sınıfıyla ilişkilidir: **owner**, **group member** ve **others**.

deneme.txt dosyası için group members ve others'lar dahil herkese **read (r)**, **write (w)** ve **execute (x)** yetkisi vermek için:

```
chmod -rwxrwxrwx deneme.txt
```

veya buna karşılık gelen sayı olarak 777 girilebilir:

```
chmod -777 deneme.txt
```

21. whoami

whoami komutu (“**who am i**” kısaltması) o anda kullanımda olan kullanıcı adını görüntüler: **whoami** çıktı olarak sistemde o an kullanımda olan **user** bilgisini verir.

Aynı sonucu echo ve \$USER değişkenini kullanarak elde edersiniz:

echo \$USER

22. man

Hangi komutun hangi işleve sahip olduğunu unuttuysanız man komutu; terminal'de çalıştırabileceğiniz ad, açıklama ve seçenekler de dahil olmak üzere tüm komut ve yardımcı programlar için kullanıcı kılavuzu sağlar.

tail komutunun kullanım talimatını görmek için:

man tail

bu kılavuzun 2. bölümünü görmek için:

man 2 tail

23. which

Çalıştırılabilir dosyaların veya bir programın dosya sistemindeki (file system) yolunu gösterir.

Python'un yolunu (path) kontrol etmek için:

which python

/c/Users/edwin/AppData/Local/Microsoft/WindowsApps/python olarak çıktı verdi.

24. df

Sistemin disk alanı kullanımını yüzde ve kilobayt (KB) cinsinden bildirmek için kullanılır. Eğer raporu megabayt olarak görmek isterseniz **df -m**, kilobayt olarak görmek isterseniz **df -k**, terabayt olarak görmek isterseniz **df -T** kullanın.

25. du

Bir dosyanın veya dizinin ne kadar alan kapladığını kontrol etmek için **du** komutu kullanılır. Bu komutu, sistemin hangi bölümünün depolamayı aşırı derecede kullandığını belirlemek için çalıştırabilirsiniz.

du komutunu kullanırken dizin yolunu belirtmeniz gerektiğini unutmayın.

-s belirtilen klasörün toplam boyutunu sunar.

-m klasör ve dosya bilgilerini MB cinsinden sunar.

-k bilgileri KB olarak sunar.

-h görüntülenen klasör ve dosyaların son değiştirilme tarihini bildirir.

26. zip, unzip

Dosyalarını bir zip dosyasına sıkıştırmak için **zip** komutu, sıkıştırılmış dosyaları bir zip arşivinden çıkarmak içinse **unzip** komutunu kullanılır:

zip archive.zip deneme.txt

deneme.txt dosyasını mevcut dizinde **archive.zip** isminde bir dosyanın içine sıkıştırdık.

unzip archive.zip
archive.zip dosyasını mevcut dizinde açtık.

27. tar

ZIP'e benzer ortak bir Linux formatıdır. Birden fazla dosyayı bir TAR dosyası içine arşivler. Sıkıştırarak arşivleme tercihe bağlıdır.

/home/user/Documents klasörü içinde archive.tar isimli bir dosya oluşturmak için:
tar -cvf archive.tar /home/user/Documents
-x bir dosyayı extract yapar.
-t bir dosyanın içeriğini listeler
-u arşivler ve mevcut bir arşiv dosyasına ekler.

28. chown

Linux'da bütün dosyaların belirli bir sahibi vardır. **chown** komutu bir dosyanın sahipliğini belirli bir kullanıcıya transfer etmek için kullanılır.

chown user2 deneme.txt
user2 kullanıcıını deneme.txt'nin sahibi yaptık.

29. jobs

Bir iş (job); shell'in (kabuğun) başlattığı bir process'tir. Bütün devam eden (çalışan) işleri durumlarıyla birlikte görüntüler.

Bu komut yalnızca csh, bash, tcsh ve ksh kabuklarında kullanılabilir

30. kill

Eğer yanıt vermeyen bir program varsa **kill** komutunu kullanarak programı elle sonlandırabilirsiniz. Bu komut yanıt vermeyen uygulamaya belirli bir sinyal yollayacak ve uygulamaya kendini sonlandırması için talimat verecektir.

Toplamda kullanabileceğiniz 64 sinyal bulunmaktadır ancak genellikle bu iki sinyal kullanılmaktadır:

- **SIGTERM** — bir programının çalışmayı durdurmasını ister ve progresi kaydetmesi için ona biraz zaman verir. Eğer kill komutunu girerken sinyali belirlemezseniz default olarak bu sinyal kullanılacaktır.
- **SIGKILL** — programları zorla hemen sonlandırır. Kaydedilmemiş progresi kaybetmiş olursunuz.

Sinyalleri bilmenin dışında ayrıca sonlandırmak istediğiniz programın işlem kimlik numarasını (PID) bilmeniz gerekmektedir.

Eğer PID'i bilmiyorsanız **ps ux** komutunu çalıştırın.

Hangi sinyali kullanacağınızdan ve programın PID'ini bildiğinizden emin olduktan sonra aşağıdaki syntax'ı girin:

kill [sinyal seçeneği] PID.

31. ping

Bir sunucuya bağlantınızı kontrol etmek için bu komut kullanılır.

Google'a bağlanıp bağlanamadığınızı öğrenmek ve yanıt süresini ölçmek için:

ping google.com

32. wget

Bu komutun yardımıyla İnternet'ten dosya indirebilirsiniz. Bunu yapmak için **wget** ve daha sonra ise indirmek istediğiniz dosyanın bağlantısını girin.

wget https://www.python.org/ftp/python/3.10.8/python-3.10.8-amd64.exe
python'un official sitesinden python'un 3.10.8 sürümünü indirdik.

33. uname

Kısaltılmışı Unix Name olan **uname** komutu Linux sisteminiz hakkında makine adı, işletim sistemi, çekirdek vb. gibi bilgileri verecektir.

33. top

Windows'daki Görev Yöneticisi'nin bir eşleniği olan **top** komutu; çalışmakta olan programların listesini ve bu programların ne kadar **CPU** kullandığını gösterecektir. Sistem kaynakları kullanımını izlemek oldukça kullanışlıdır ve özellikle de fazla sistem kaynağı kullanan programları sonlandırabilmeniz için size yardımcı olacaktır.

34. history

Linux'u bir süre kullandıktan sonra her gün yüzlerce komut çalıştırdığınızı fark edeceksiniz. Eğer daha önce girdiğiniz komutları görmek istiyorsanız bu komutu çalıştırabilirsiniz.

35. hostname

Eğer host'un veya ağın adını bilmek istiyorsanız bu komutu girin. Bu komutun sonuna **-i** eklemek ağınızın IP adresini görüntüleyecektir.

36. useradd, userdel

Linux çok kullanıcılı bir sistem olduğundan aynı anda birden fazla kişi aynı sistemle aynı anda etkileşime geçebilir. **useradd** komutu yeni bir kullanıcı eklemek için kullanılırken **passwd** komutu bir kullanıcı hesabına şifre eklemek için kullanılır.

Edwin adlı yeni bir kullanıcı eklemek için:

useradd Edwin

Bu kullanıcıya şifre eklemek için:

passwd 123456789

Bu kullanıcının hesabını silmek için:

userdel Edwin