



# Açık Kaynak İşletim Sistemi (Giriş)

# Temel Kavramlar

## İşletim Sistemi

Bilgisayar kullanıcısı ile bilgisayarı oluşturan donanım arasındaki iletişimi sağlayan, aynı zamanda diğer uygulama yazılımlarını çalıştırmaktan sorumlu olan sistem yazılımıdır.



# Temel Kavramlar

## Açık Kaynak Kodlu Yazılım

- Açık kaynak kodlu yazılım (Open Source), kaynak kodu isteyen herkese açık olan yazılımlardır. Bu tür yazılımların ayırt edici özelliği kullanıcıya yazılımı değiştirme özgürlüğü sağlamasıdır.
- Dünyanın her tarafından bilişim uzmanlarınca geliştirilen açık kaynak kod yazılımları, insanlığın ortak malıdır.
- Açık kaynak kod yazılımlar hemen her alanda çözümler sunarak kapalı kod yazılımlara karşı tüketiciye seçenekler sunar.

# Temel Kavramlar

## Açık Kaynak Kodlu Yazılım

- En iyi bilinen açık kaynak kodlu yazılımlar; Linux, OpenOffice...
- Dünya üzerindeki hemen her ülke kendi gereksinimlerine göre açık kaynak kodlu bir Linux dağıtımı geliştirmiştir.
- Pardus Ulusal İşletim Sistemi, TÜBİTAK.
- 26 Aralık 2005 – Pardus 1.0 (İlk Pardus Sürümü)
- Pardus 17.4 (Son Pardus Sürümü)
- <http://www.pardus.org.tr/>

# Temel Kavramlar

## GNU Genel Kamu Lisansı (GPL)

- Piyasaya sürülen her yazılım, üretici şahıs ya da firmanın haklarını ve kullanıcının haklarını korumaya yönelik bir lisans anlaşması içerir.
- GPL lisans anlaşması, 1983 yılında Richard Stallman tarafından geliştirilmiş lisans türü.
- GPL' in en çok üzerinde durduğu konu yazılımların kaynak kodu ile birlikte dağıtılmasının gerekliliğidir. Üretici firma yazılımını binary olarak dağıtsa bile kaynak kodunu da herkes tarafından erişilebilir bir yere bırakmak zorundadır.

# Temel Kavramlar

## GNU Genel Kamu Lisansı (GPL)

- Şeffaflık, kullanıcının ne kullandığını bilmesi...
- GPL, yazılımın ücretlendirilmesi hakkında hiçbir fikir beyan etmez.
- Pek çok lisans anlaşması ücretsiz kategorisine girerken GPL yazılımları ücretsiz olmak zorunda **değildir**.
- GPL yazılımların çok büyük bir kısmı **ücretsizdir**.
- Genel yaklaşım yazılımdan değil, kullanıcıya sunulan destekten para kazanılması...

# UNIX – Linux

## UNIX İşletim Sistemi

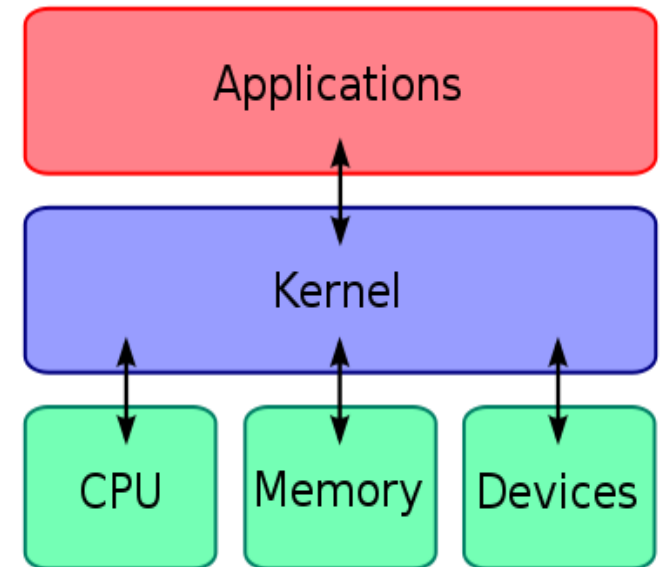
- 1970, Dennis Ritchie – Ken Thompson, C programlama dili...
- BSD Dağıtımları (FreeBSD, OpenBSD, NetBSD)
- Ücretsiz (Free)

## Linux İşletim Sistemi

- 1991, Linus Torvalds, UNIX türevi işletim sistemi...
- Open source, Free
- Çok kullanıcı, çok görevli, güçlü bir işletim sistemi

# Kernel (Çekirdek)

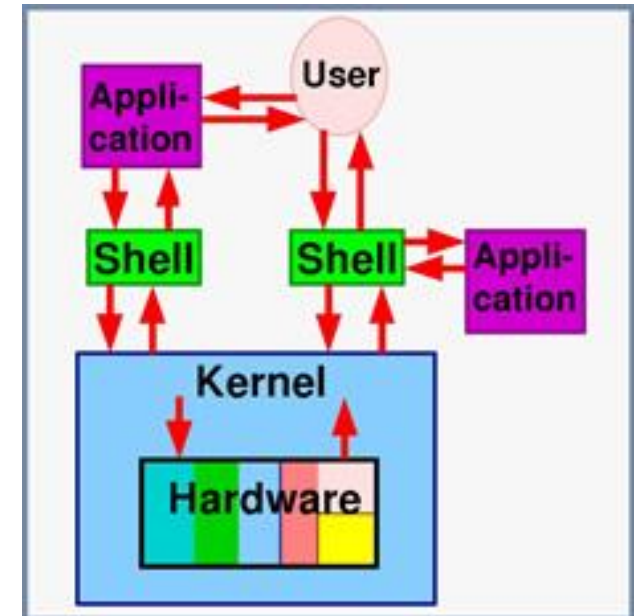
- Software ve Hardware arasındaki bağlantı (arayüz)
- Çalışan tüm programların ve süreçlerin hafıza ile olan ilişkilerini düzenler.
- İşlemcinin tüm bu programlar arasında adil bir şekilde dağıtımını yapar.
- Sistem açılırken belleğe yüklenir ve sistem kapatılıncaya kadar bellekte kalır.



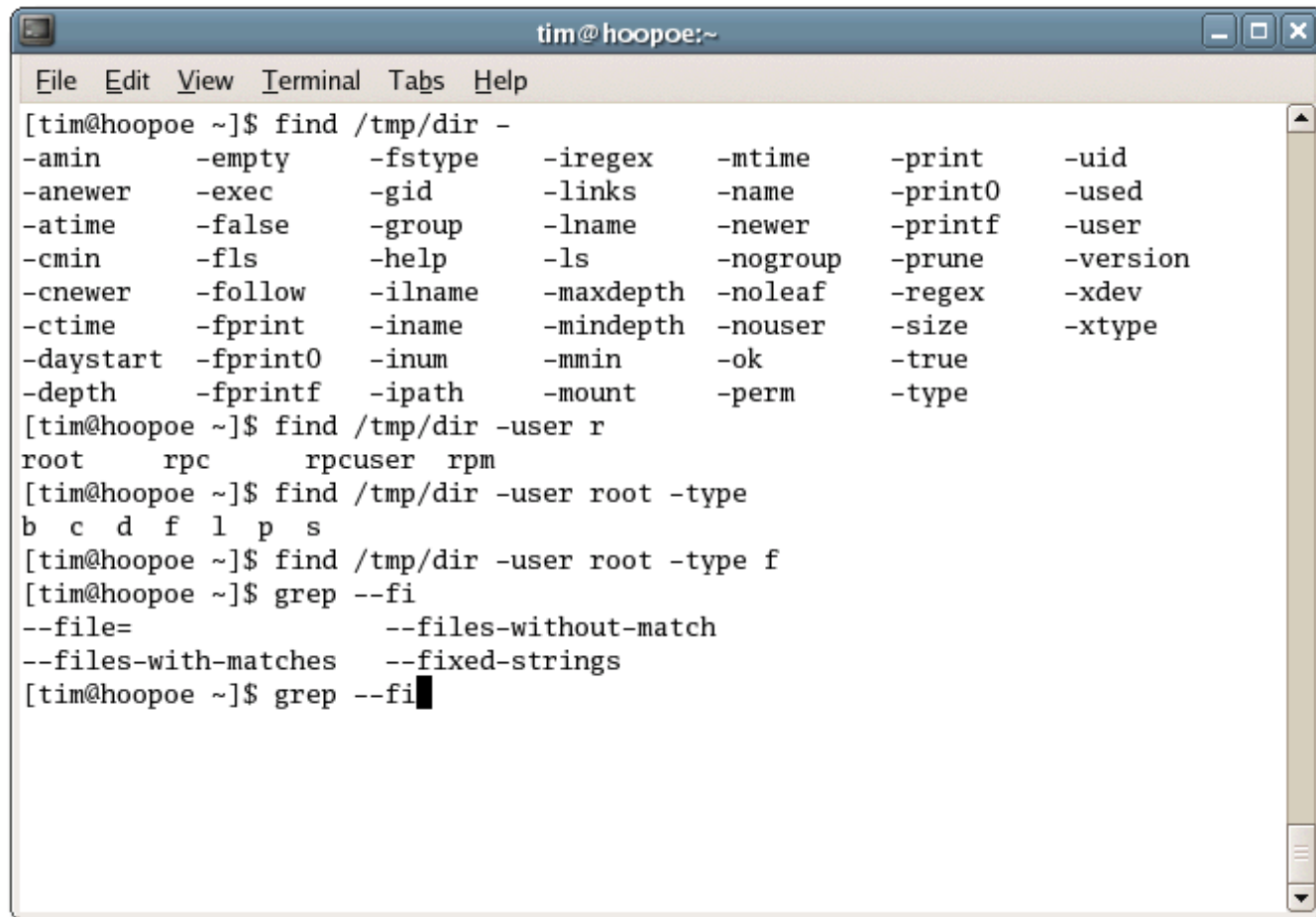


# Shell (Kabuk)

- Kernel ile kullanıcının iletişim kurmasını sağlayan programlardır.
- Linux, Unix gibi sistemlerde komutları yorumlamak ve yönetmek için kullanılan programlardır.
- **bash – sh – csh – ksh**



# Shell (Kabuk) – Terminal



```
tim@hoopoe:~  
File Edit View Terminal Tabs Help  
[tim@hoopoe ~]$ find /tmp/dir -  
-amin      -empty      -fstype     -iregex     -mtime      -print      -uid  
-anewer    -exec        -gid        -links      -name       -print0     -used  
-atime     -false      -group      -lname      -newer      -printf     -user  
-cmin      -fls        -help       -ls         -nogroup    -prune      -version  
-cnewer    -follow     -ilname     -maxdepth   -noleaf     -regex      -xdev  
-ctime     -fprint     -iname      -mindepth   -nouser     -size       -xtype  
-daystart  -fprint0    -inum       -mmmin      -ok         -true  
-depth     -fprintf    -ipath      -mount      -perm       -type  
[tim@hoopoe ~]$ find /tmp/dir -user r  
root      rpc      rpcuser  rpm  
[tim@hoopoe ~]$ find /tmp/dir -user root -type  
b c d f l p s  
[tim@hoopoe ~]$ find /tmp/dir -user root -type f  
[tim@hoopoe ~]$ grep --fi  
--file=                --files-without-match  
--files-with-matches    --fixed-strings  
[tim@hoopoe ~]$ grep --fi
```

# Linux' un Genel Kullanım Amaçları

- **Kişisel Kullanım** (Gelişimleri henüz tamamlanmamış olmasına rağmen, herhangi bir kişisel bilgisayardan beklenebilecek yazı editörleri, hesap cetvelleri, çizim yazılımları, veritabanları birçok ihtiyaca cevap verir.)
- **Internet sunucusu** (WWW, DNS, SMTP, FTP...)
- **Yazılım Geliştirme** (C, C++, Java, Fortran, Pascal)

# Linux (Avantaj/Dezavantajları)

- **Avantaj**

- Open Source (Açık kaynak kod)
- Hızlı ve daha güvenilir
- Ücretsiz
- Yüksek performans
- Hızlı destek ve sorun çözme

- **Dezavantaj**

- Basit kullanıcıların rahatça kullanabileceği düzeyde değil.
- Bazı hataları çözmek biraz bilgi gerektirebiliyor.
- Alışması zaman alabiliyor.
- Sıklıkla kullandığınız programların aynısı bulamayabilirsiniz.

# Linux Dağıtımları



# Linux Dosya Sistemleri

- Disk üzerindeki dosyaların düzenlenmesini ve bir işletim sisteminde dosyaların izlerinin bulunmasını sağlayan yapıya Dosya Sistemi (File System) denir.
- Dosya sistemi tercihi kullanılan bilgisayar sisteminin ne amaçla kullanılacağına ve bu sistemdeki dosyaların hangi boyutta olacağına, performansın veya sağlamlılığın ön planda tutulmasına göre yapılır.
- Linux dosya sisteminde en çok kullanılan dosya sistemleri :

**ext2 - ext3 - ext4 – ReiserFs - XFS**

# Linux Dosya Sistemleri

File System	Maximum file size	Maksimum volume size
FAT16	2 GB	2 GB
FAT32	4 GB	8 TB
ext2	16 GB to 2 TB	2 TB to 32 TB
ext3	16 GB to 2 TB	2 TB to 32 TB
ext4	16 GB to 16 TB	1 EB
ReiserFs	8 TB	16 TB

# Linux Masaüstü Ortamları

GNOME



GNOME

KDE







Computer



test's Home



Trash



My Computer



Control Center



Synaptic



Windows Media Pla...



Trash



Printers



Xkill



KDE logo



iTunes



kdesu KPPP



Audio CD



2.0G Removabl...



Skype



GAIM



Thunderbird



K3b



KAudioCrea  
tor



Xine



Amarok



KMix

rick - D3lphin

File Edit View Go Tools Settings Help

Bookmarks

/home/rick/

Home

Storage Media

Network

cxoffice

Desktop

mp3

packages

temp

bin

6 Items (5 Folders, 1 File)

11.9 GB free



rick - D3lphin









GIMP



13:07

# OpenOffice, LibreOffice

## Open Office – Libre Office

	Sözcük İşlemci	Microsoft Word benzeri işlevi vardır. Ayrıca ek bir program gerektirmeden PDF dosyası oluşturabilirsiniz.
	Hesap Tablosu	Microsoft Excel'e benzeyen bir hesap tablosu programıdır.
	Sunum	Microsoft PowerPoint'e benzer bir sunum programıdır. Sunumları PDF olarak dışa aktarma yeteneklerini de içerir.
	Veritabanı	Microsoft Access'e benzer bir veritabanı programıdır.
	Çizim	CorelDRAW gibi bir vektörel grafik editörüdür. Scribus ve Microsoft Publisher gibi masaüstü yayıncılık programları ile benzer özelliklere sahiptir.
	Denklem Düzenleyicisi	Matematik formülleri yaratmak ve düzenlemek için bir araçtır. Microsoft Denklem Düzenleyicisi'ne benzer.

# LibreOffice



Text Document



Spreadsheet



Presentation



Open...



Drawing



Database



Formula



Templates...



# GIMP

- **GIMP= GNU Image Manipulation Program** (GNU Görüntü İşleme Programı)
- Linux altındaki en gelişmiş serbest yazılım grafik programıdır.
- Ayrıca Photoshop kullanıcılarının GIMP ile rahat edebilmesi için bir de GIMPshop vardır. Böylece GIMP, Photoshop'a benzetilir.





# Ubuntu Yazılım Merkezi

- **Ubuntu Yazılım Merkezi (Ubuntu Software Center)** Ubuntu işletim sisteminde yazılım kurmak, kaldırmak ve güncellemek için kullanılan bir paket yöneticisidir.
- Python programlama dili ile geliştirilmiştir.
- **Linux' un özellikle sunucu işletim sistemlerinde** program kurma, kaldırma işlemi terminal ekranında ilgili komutları yazarak gerçekleştirilir.
- Örneğin,  
`sudo apt-get install phpmyadmin`  
`sudo apt-get remove phpmyadmin`


# Ubuntu Yazılım Merkezi


## App departments


 Accessories


 Books & Magazines


 Developer Tools


 Education


 Fonts


 Games


 Graphics

 Internet

 Office

 Science & Engineering

 Sound & Video

 Themes & Tweaks

 Universal Access

Search for an app:



## Top rated apps on the Ubuntu Software Centre



**SMPlayer**  
Sound & Video | **FREE**  
A great MPlayer front-end

★★★★★  
70 Ratings



**qBittorrent**  
Internet | **FREE**  
Download and share files over BitTorrent

★★★★★  
78 Ratings



**K3b**  
Sound & Video | **FREE**  
Disk writing program

★★★★★  
74 Ratings



**Artha**  
Accessories | **FREE**  
Handy thesaurus based on WordNet

★★★★★  
16 Ratings



**KeePassX**  
Accessories | **FREE**  
Cross Platform Password Manager

★★★★★  
41 Ratings



**GLX-Dock (Cairo-Dock with OpenGL)**  
Accessories | **FREE**  
Cairo-Dock with OpenGL (hardware acceleration)

★★★★★  
77 Ratings







back | back





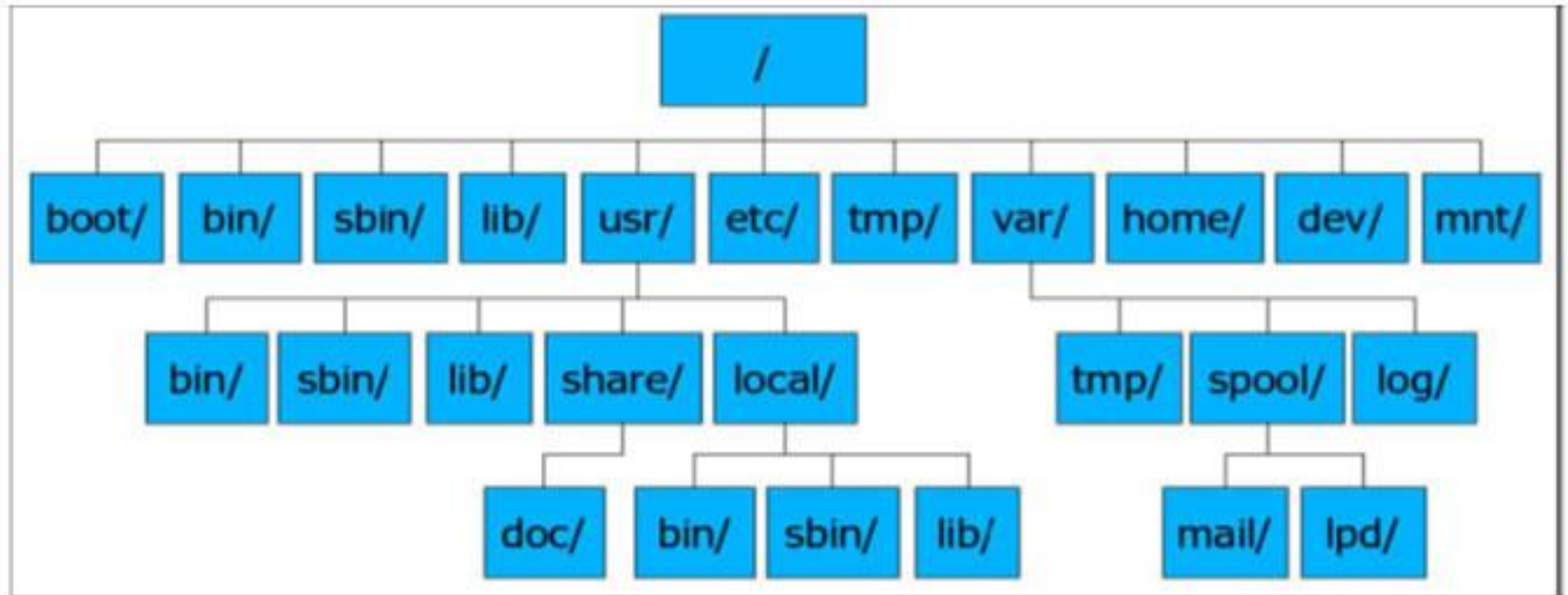
# Linux Dizin Yapısı

# Dosya Sistemi ve Çalışma Mantığı

## Dosya Sistemi

- Disk üzerindeki dosyaların düzenlenmesini ve bir işletim sisteminde dosyaların izlerinin bulunmasını sağlayan yapıya Dosya Sistemi (File System) denir.
- Linux işletim sisteminde de dosya ve dizinler diğer Unix türevi işletim sistemlerinde olduğu gibi hiyerarşik yapıya sahiptir.
- En üst düzeyde olan kök dizini olarak adlandırılan dizin, alt dizinleri içermektedir.

# Linux Dosya Sistemi



# Linux Dosya Sistemi

## Kök Dizin (/)

- Linux işletim sistemlerinde dosya hiyerarşisinde en üstte bulunan dizindir. Diğer bütün dizinler kök dizinin altında bulunur.

## /bin :

- Genel kullanıcıların kullanabileceği komutlara ait dosyaların bulunduğu dizindir.
- ls, mkdir, cat, cp... gibi temel komutlar bu dizin altında bulunur.

# Linux Dosya Sistemi

## /sbin :

- Sadece root kullanıcısının kullanabileceği komutlara ait dosyaların bulunduğu dizin.

## /boot :

- Sistem açılışında kullanılan dosyaların bulunduğu dizin.

## /dev :

- Aygıtlar ve disk bölümlerine ait dosyaların bulunduğu dizin.

# Linux Dosya Sistemi

## /etc :

- Sisteme ait yapılandırma (konfigürasyon) dosyalarının tutulduğu dizin.

**/etc/passwd** : Her kullanıcı hakkında ayrıntılı bilgilerin bulunduğu, kullanıcı veritabanı olarak da düşünebileceğimiz yerdir. Burada kullanıcıların gerçek ismi, kullanıcı isimleri, şifrelenmiş parolaları, ev dizinleri gibi bilgiler yer almaktadır.

**/etc/group** : /etc/passwd dosyasına benzer ama kullanıcılar yerine grupları tanımlar.

**/etc/rc veya /etc/rc.d** : Açılıшта çalışacak betikler ve betik dizinleri burada bulunur.

# Linux Dosya Sistemi

## /home :

- Sistemdeki kullanıcıların ev dizinidir. Kullanıcıların kişisel dosyaları burada tutulur.
- Sisteme yeni bir kullanıcı eklendiği zaman bu dizin altında kullanıcı adıyla bir dizin (klasör) oluşturulur.  
/home/murat     /home/ahmet     ...
- Windows' taki **Users (Kullanıcılar)** klasörüne benzetebiliriz.

# Linux Dosya Sistemi

## /lib :

- Programların ihtiyacı olan kütüphane (library) dosyalarının bulunduğu dizindir.

## /mnt :

- Sistem yöneticisinin geçici olarak herhangi bir dosya sistemine ulaşmak için oluşturabileceği bağlama noktalarını içeren dizin.

**Mount işlemi** (Üzerinde bir dosya sistemi olan bir disk birimine veya parçasına okuma veya yazma amacıyla ulaşılabilmesi için bu birim veya parçanın / dosya yapısında bir alt dizine mount edilmesi (bağlanması) işlemi.)



# Linux Dosya Sistemi

## /opt :

- Dağıtımdan bağımsız ekstra yüklenen paketler için kullanılan dizin.
- **Örneğin**, (Google Earth programını indirip kurmak istediğinizde 'default' olarak kurulacağı dizin **/opt/google-earth** gibi bir adrestir.)

## /var :

- Log dosyaları, e-mail ve printer kuyrukları gibi değişken sistem bilgilerini barındırır. Ayrıca web sunucu yazılımına ait dosyaların bulunduğu dizindir.

# Linux Dosya Sistemi

## /tmp :

- Geçici dosyaların tutulduğu dizindir. Birçok program burayı geçici depolama alanı olarak kullanır.

## /proc :

- Sistem süreçlerinin bilgisinin bulunduğu sanal dosya sisteminin dosyalarının bulunduğu dizin.
- **Örneğin**, ( `cat /proc/swaps` yazarak sisteminizdeki takas dosyalarına dair bilgiler alabilir ya da `cat /proc/cpuinfo` komutuyla işlemcinizin özelliklerini görebilirsiniz.

# Linux Dosya Sistemi

## **/root :**

- Sistem yöneticisinin (root kullanıcısının) ev dizinidir.

## **/usr :**

- Tüm kullanıcılarla paylaşılan verileri (programlar, komutlar, kütüphaneler vb.) içeren dizindir.

## **/media :**

- Kaldırılabilir aygıtların (CD-ROM, Flash Bellek gibi) sisteme eklendiği dizindir.

# Linux Dosya Sistemi

## Her şey bir dosya!

- Linux' ta sistemdeki her şey bir dosyadır. Buna sistemdeki aygıtlar, donanım kaynakları hatta hafızada çalışan programlar da dahildir.
- Linux çekirdeği örneğin bir ethernet kartına ulaşmak istediğinde `/dev/eth0` dosyasına okuma yazma yapar veya bir sabit disk bölümü için `/dev/sda1` dosyasını kullanır.



# Linux Komutları - I

# Yardım Komutları

- Linux' ta komutlar hakkında yardım almak için aşağıdaki komutlar kullanılır :
  - **man**
  - **info**
  - **whatis**
  - **apropos**



# Yardım Komutları

## man komut

- Parametre olarak aldığı komut hakkında bilgiyi ekranda gösterir.

**man ls** (ls komutu hakkında bilgi verir.)

**man pwd** (pwd komutu hakkında bilgi verir.)

- Komut çalıştırıldığında uzun bir açıklama gelebilir.
  - Açılan sayfada ilerlemek için **space (boşluk)** tuşuna basılır.
  - **q (quit-çıkış)** tuşuna basılarak ekrandan çıkılır.

# Dosya Tipleri

- Dosya uzantıları dosyalar ile işlem yaparken kolaylık sağlar, hangi dosyanın hangi uygulama ile açılacağını belirlememizi sağlar.
- Dosya uzantısı dosyanın isminde son noktadan sonraki kısımdır ve her zaman dosyaların uzantısı olmak zorunda değildir.



# Dosya Tipleri

- Linux işletim sisteminde temelde 2 farklı dosya tipi bulunmaktadır.
  - Aygıt Dosyaları
  - Sıradan Dosyalar
- Aygıt dosyaları sistemde bulunan donanım ile haberleşmeyi sağlayan dosyalardır. İki farklı aygıt dosya tipi vardır:
  - Karakter aygıt dosyası: Veri alışverişini byte bazında yapan aygıtlar (Klavye, Mouse gibi)
  - Blok aygıt dosyası: Veri alışverişini bloklar halinde yapan aygıtlar (Harddisk gibi)

# Dosya Tipleri

- Sıradan dosyalar: Aygıt dosyaları dışında kalan dosyalardır.

**ls -l**

<i>-rwxr--r--</i>	<i>Normal (sıradan) dosya</i>
<i>crw-rw-rw-</i>	<i>Karakter aygıt dosyası</i>
<i>brw-r--r--</i>	<i>Blok aygıt dosyası</i>
<i>lrw-r--r--</i>	<i>Sembolik bağlantı (link) dosyası</i>
<i>srw-rw-rw-</i>	<i>Soket dosyası</i>
<i>prw-----</i>	<i>Pipe dosyası</i>
<i>drwxr-xr-x</i>	<b><i>Dizin (Klasör)</i></b>

# Link dosyası oluşturma

**ln -s** dosyaKonumu **linkinAdı**

```
murat@hp:~Masaüstü$ ln -s /boot/grub/grub.cfg acilis  
murat@hp:~Masaüstü$ ls -l
```

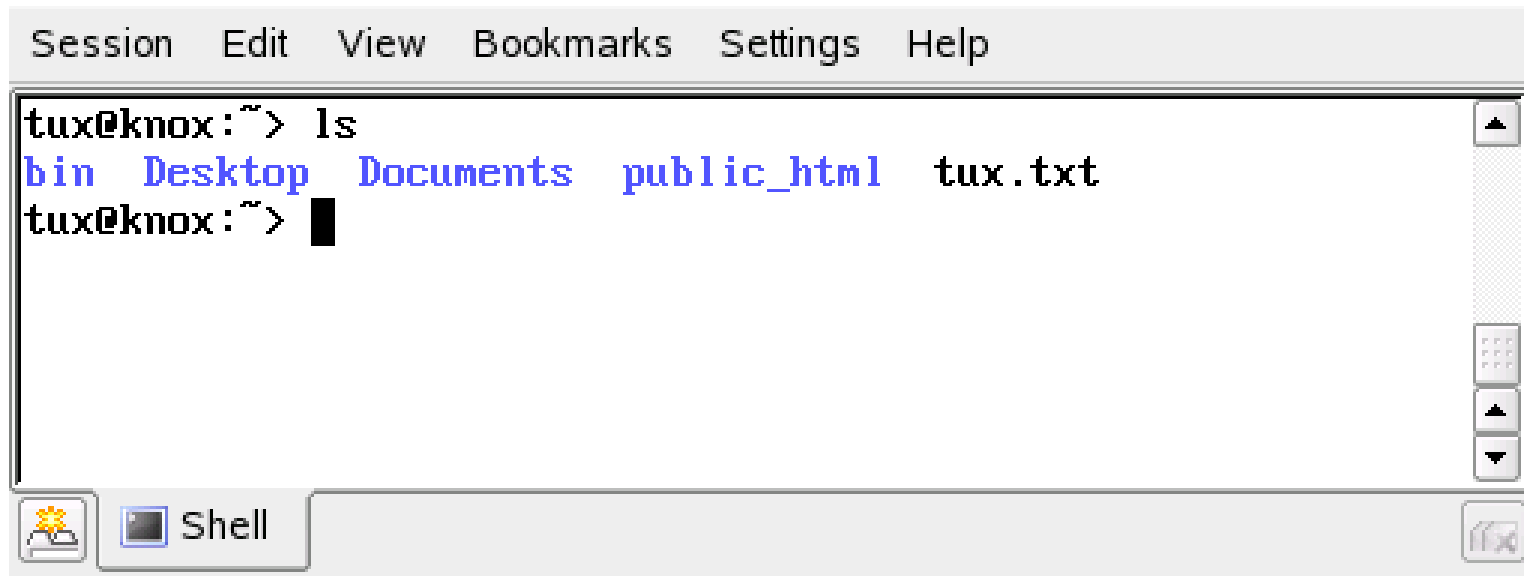
```
lrw-r--r-- 1 root root acilis -> /boot/grub/grub.cfg
```

# Dosyaların isimlendirilmesi

- Linux'ta dosya isimlerinin uzunluğu 256 karaktere kadar uzayabilir.
- Dosya isimleri büyük/küçük harfe karşı duyarlıdır.
- " / " karakteri dosya yolunda kullanıldığı için dosya isimlerinde kullanılamaz.
- " . " işareti ile başlayan dosyalar gizli dosyalardır ve `ls -a` komutu ile görüntülenirler.

# ls komutu

- Bir dizin (klasör) içindeki dosya ve klasörleri listeler.
- DOS' taki **dir** komutunun karşılığıdır.



The screenshot shows a terminal window with a menu bar at the top containing 'Session', 'Edit', 'View', 'Bookmarks', 'Settings', and 'Help'. The terminal text shows the prompt 'tux@knox:~>' followed by the command 'ls'. The output of the command is 'bin Desktop Documents public\_html tux.txt', where the directory names are in blue and the file name is in black. Below the output, the prompt 'tux@knox:~>' is shown again with a black cursor. On the right side of the terminal window, there are vertical scrollbars and a zoom control. At the bottom left, there is a 'Shell' button with a terminal icon. At the bottom right, there is a close button with a red 'X' icon.

```
tux@knox:~> ls
bin Desktop Documents public_html tux.txt
tux@knox:~> █
```

# ls komutu parametreleri

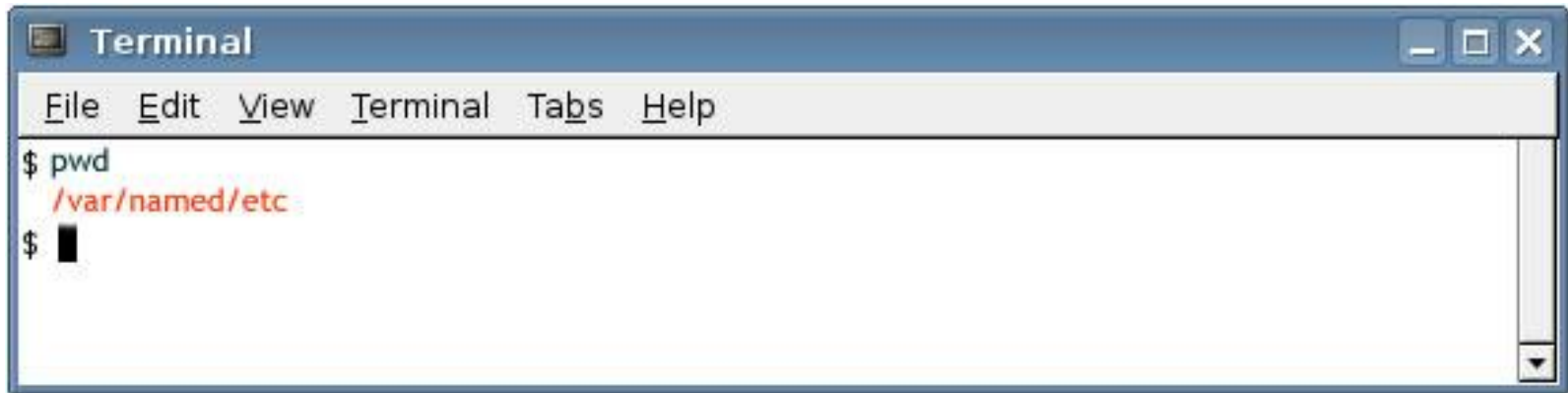
- **-a** : Gizli dosyalar (. ile başlayan) dahil tüm dizinin içeriğini listeler.
- **-l** : Dosyalar hakkında ayrıntılı bilgi verir. (dosyanın sahibi, boyutu, değiştirilme tarihi, izinler...)
- **-t** : Kronolojik sıralama yapar. En son yaratılan dosya/dizinden ilk yaratılana doğru sıralar.
- **ls -al** birlikte de kullanılabilir.

# Wildcard (Joker) karakterler

- `ls -l a*`
- `ls -l test?`
- `ls -l *[abc]`
- `ls -l [A-Z]*`
- `ls -l [^A-Z]*`
- `ls -l ?[0-9]*`

# pwd komutu

- **pwd** (**p**rint **w**orking **d**irectory)
- İçinde bulunduğunuz dizinin adresini görüntülemek için kullanılır.

A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command prompt shows "\$ pwd" followed by the output "/var/named/etc" in red text. Below the output, there is another "\$" prompt with a cursor.

```
Terminal
File Edit View Terminal Tabs Help
$ pwd
/var/named/etc
$
```



# cd (change directory) komutu

. Hali hazırdaki dizini gösterir.

.. Bir önceki dizini gösterir.

**cd ..** Üst dizine geç.

- Dizin(klasör) değiştirmek için **cd** komutu kullanılır.
- **cd /etc** komutuyla kök dizinin altındaki **etc** klasörüne geçiş yapılır.

# touch - mkdir komutları

- Dosya oluşturmak için **touch** , klasör oluşturmak için de **mkdir** komutu kullanılır.
- **touch** deneme
- **touch** dosya1 dosya2 dosya3 ...
- **mkdir** muzik
- **mkdir** murat yazici myo
- **mkdir** /home/murat/Belgeler/dersnotlari

# rm - rmdir komutları

- Dosya silmek için **rm** , dizin silmek için ise **rmdir** komutu kullanılır.
- **rm dosya/dizin\_adi**
- **rmdir dizin\_adi**
- Bir dizin boş iken **rmdir** komutu ile silinebilir, eğer boş değilse bu komut işe yaramayacaktır.

# rm - rmdir komutları

- **rm** komutunun aldığı bazı parametreler :
  - i** : Dosyanın silinmesi için kullanıcıdan onay ister.
  - f** : Kullanıcıdan hiçbir zaman onay istemez.
  - r** : Komut rekürsif çalışıp alt dizin ve dosyaların da silinmesi sağlanır.
- **rm -rf *murat*** komutu ile **murat** klasörü içerisindeki tüm dosya ve alt klasörlerle beraber silinecektir.

# cp komutu

- Dosya ve dizin kopyalamak için **cp** komutu kullanılır.
- **cp kaynakdosya hedef**
- **cp abc.txt /tmp** komutu ile "abc.txt" isimli dosya kök dizin altındaki **tmp** klasörüne kopyalanır.
- **cp -R kaynakdizin hedefDizin**
- **cp -R myo /home/murat/Belgeler**
- myo dizinini içindeki tüm dosya ve alt klasörlerle birlikte /home/murat/Belgeler dizinine kopyalar.

# mv komutu

- Dosya taşımak / Yeniden adlandırmak için kullanılır.  
Alacağı bazı parametreler :
  - i** : Hedef dosya veya dizin varsa üzerine yazma için onay alır.
  - f** : Hedef dosya veya dizin varsa üzerine yazma işlemini onay almadan yapar.
- **mv pf.conf /etc** komutu ile **pf.conf** dosyası /etc dizininin altına taşınacaktır.

# Dosya isimlerini değiştirmek

- Linux'te bir dosyanın ismini değiştirmek demek o dosyanın başka bir isimde taşınması anlamına gelir.
- `mv test test2` komutu ile `test` isimli dosyanın adı `test2` olarak değişir.

# Dosya okuma

- Linux'ta dosya okumak için kullanılan temel komutlar :

- ☐ cat
- ☐ more
- ☐ less
- ☐ tac
- ☐ rev
- ☐ head
- ☐ tail



# cat

- Dosyanın içeriğinin ekranda yazdırılmasını sağlar.
  - ❖ **-n** parametresi ile kullanıldığında satırları numaralandırır.

```
[08:03:43][root]$ cat -n test14
 1 merhaba
 2 bu
 3 bir
 4 deneme
 5 yazisidir.
```

```
[08:04:00][root]$
```

```
[08:04:00][root]$
```

```
 2 yazisidir.
```

# more, less

**more :** Dosyanın içeriğini sayfa sayfa gösterir.

**less :** Aşağı ve yukarı ok tuşlarıyla sayfada ilerlenebilir.

❖ **space (boşluk)** ile bir sonraki sayfaya geçer.

❖ **b** bir önceki sayfaya geçer.

❖ **q** ile çıkış (quit).

# tac, rev

**tac :** Dosyanın içeriğini sondan başa doğru gösterir.

❖ **tac** **dosyaAdı**

**rev :** Dosyanın içeriğini her satırı ters çevirerek gösterir.

❖ **rev** **dosyaAdı**

# head

Dosya içeriğinin baştan belirtilen satır kadar gösterilmesini sağlar. Varsayılan (default) değeri 10' dur.

❖ **head -5 linux**

linux isimli dosyanın ilk 5 satırı ekranda görüntülenir.

❖ **head linux**

linux isimli dosyanın ilk 10 satırı ekranda görüntülenir.

# tail

Dosya içeriğinin sondan belirtilen satır kadar gösterilmesini sağlar. Varsayılan (default) değeri 10' dur.

❖ **tail -5 linux**

linux isimli dosyanın son 5 satırı ekranda görüntülenir.

❖ **tail linux**

linux isimli dosyanın son 10 satırı ekranda görüntülenir.



# Linux Kullanıcı Yönetimi ve Dosya/Dizin İzinleri

# Dosya/Dizin İzinleri

- Linux'te temel dosya sistemi güvenliği kullanıcıların dosya ve dizinler üzerindeki erişim izinlerinin belirlenmesiyle sağlanır. Bir dosya veya dizinlere ait 3 tür izin vardır:
  - Dosya sahibinin izinleri (Dosyanın sahibi : u)
  - Dosya grubunun izinleri (Kullanıcı grubu : g)
  - Diğer kullanıcıların izinleri (Diğer kullanıcılar : o)

# Dosya/Dizin İzinleri

## ☐ Okuma (r)

- Dosya içeriğini görebilir mi ?
- (Klasörler için) dosya listesini alabilir mi ?

## ☐ Yazma (w)

- Dosyaya yazma izni, aynı zamanda dosyayı silme izni ?
- (Klasörler için) Bu klasör içinde dosya veya alt klasör oluşturulabilir mi ?

## ☐ Çalıştırma (x)

- Dosyayı çalıştırabilir mi?
- (Klasörler için) Bu klasöre geçebilir mi?



# Dosya/Dizin İzinleri

**-rwxrw-r--**

Dosya tipinin belirtildiği kısım

"-" Normal dosya

"d" Dizin (Klasör)

"c" Karakter bazında işlem yapan aygıt sürücüler

"b" Blok bazında işlem yapan aygıt sürücüler

"l" Başka bir dosyayı gösteren link dosyaları

# Dosya/Dizin İzinleri

Dosyanın sahibi  
Grup  
Diğer kullanıcılar

**-rwxrwx-r--**

İzinlerin belirtildiği kısım (**ls -l** komutu çıktısı)

# Dosya/Dizin İzinleri

```
root@localhost # ls -l
```

```
drwxr-xr-- 10 murat gs 4096 Nov 14 23:43 isletim
```

d-> Dizin olduğunu gösteriyor.

1. **üçlü** -> (rwx) **murat** isimli kullanıcı tüm haklara sahip

2. **üçlü** -> (r-x) **gs** grubuna üye olan kullanıcılar okuma ve çalıştırma haklarına sahip

3. **Üçlü** -> (r--) Diğer kullanıcılar sadece okuma hakkına sahip

❖ **murat (dosya/dizin sahibi)** - **gs (dosya/dizinin grubu)**

# Dosya/Dizin İzinleri

## chmod komutu (change mode)

- ❑ Unix ve Linux sistemler üzerinde dosya ve dizinlerin erişim yetkilerini belirlemek, değiştirmek için kullanılır.
- ❑ **-R** : Alt klasör ve dosyalardaki izinleri değiştirir. (recursive)

# Dosya/Dizin İzinleri

- ❑ İzin eklemek için **+**, İzin kaldırmak için **-**
- ❑ Okuma: **r** , Yazma: **w** , Çalıştırma: **x**
- ❑ Sahibi: **u** , Grubu: **g** , Diğerleri: **o** , Herkes: **a**
- `chmod u+rw, g-w, o-rwx filename`

# Dosya/Dizin İzinleri

**\$ chmod go-rwx deneme**

Bu durumda grup ve diğerlerinden rwx yetkisinin geri alınacağı anlaşılmalıdır.

**\$ chmod +x dosya**

Herkes çalıştırma izni verilir.

**\$ chmod u+rwx dosya ?**

# chmod izinlerinin sayı sistemiyle gösterimi

<u>Kod</u>	<u>İzin Durumu</u>
400	Dosya sahibi için okuma
200	Dosya sahibi için yazma
100	Dosya sahibi için çalıştırma
040	Gruptakiler için okuma
020	Gruptakiler için yazma
010	Gruptakiler için çalıştırma
004	Diğerleri için okuma
002	Diğerleri için yazma
001	Diğerleri için çalıştırma

r (okuma) : 4

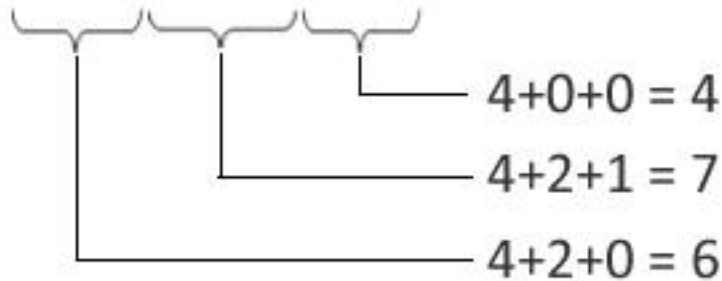
w (yazma) : 2

x (çalıştırma) : 1

# chmod izinlerinin sayı sistemiyle gösterimi

**\$ chmod 674 dosya**

**-rw-rwxr-- murat users ..... dosya**





# Dosya sahibini deęiřtirme

## chown komutu (change owner)

❑ Dosya/dizin sahibini deęiřtirmek için kullanılan komuttur.

**sudo** **chown** **yeniSahip** **dosya/dizin**

**sudo:** Normal bir kullanıcıya root hakları ile komut çalıştırma yetkisi verir.

# Dosya sahibini deęiřtirme

Önceki durum :

**-rwxr-xr-x** **ahmet** users ..... deneme

**\$ sudo chown mehmet deneme**

Sonraki durum :

**-rwxr-xr-x** **mehmet** users ..... deneme

# Grubu değiştirme

## chgrp komutu (change group)

❑ Dosya/dizin grubunu değiştirmek için kullanılan komuttur.

**sudo chgrp yeniGrup dosya/dizin**

❑ Bu komutu çalıştırabilmek için de kullanıcının root yetkisine sahip olması gerekir.

# Grubu değiştirme

Önceki durum :

**-rwxr-xr-x** ahmet users ..... deneme

**\$ sudo chgrp murat deneme**

Sonraki durum :

**-rwxr-xr-x** ahmet murat ..... deneme

# Yeni kullanıcı eklemek – silmek

## adduser - deluser komutları

- ❑ Sisteme yeni kullanıcı eklemek ve silmek için kullanılan komutlardır.

**sudo adduser kullanıcıAdı**

**sudo deluser kullanıcıAdı**

**/etc/deluser.conf** *(Ekstra ayarlar bkz.)*

# Grup oluşturmak – Grubu Silmek

## addgroup – delgroup komutları

- ❑ Sisteme yeni grup eklemek ve grup silmek için kullanılan komutlardır.

**sudo addgroup grubunAdı**

**sudo delgroup grubunAdı**

# Gruba kullanıcı eklemek

**sudo adduser username groupname**

- ❑ Sistemde tanımlanmış bir gruba kullanıcı eklemek için kullanılan komuttur.

**sudo adduser murat gs**

komutuyla **murat** kullanıcısı **gs** grubuna eklenecektir.

# Kullanıcının üyesi olduğu grupları listelemek

- ❑ Bir kullanıcının üyesi olduğu grupları listelemek için **groups** komutu kullanılır.

**murat@localhost # groups**

komutuyla **murat** kullanıcısının üyesi olduğu gruplar ekranda listelenir.



# Bir grubun kullanıcılarını listelemek

❑ Bir gruba üye kullanıcılar `/etc/group` dosyasında tutulur.

**murat@localhost # cat /etc/group**

komutuyla sistemde var olan tüm grupları ve bu gruba üye kullanıcıları ekranda listellersiniz.



# Linux Metin Editörleri

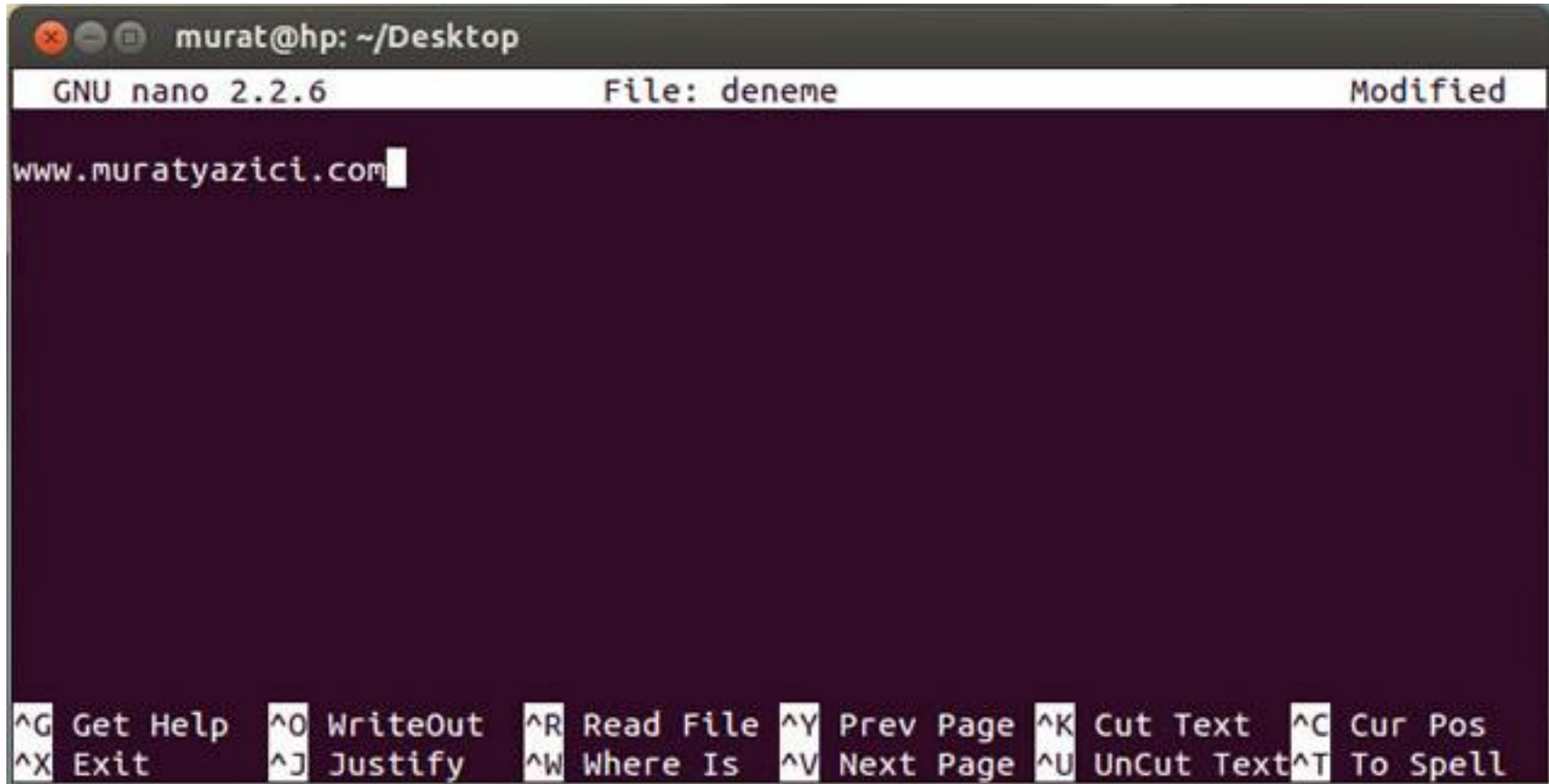
# Linux Metin Editörleri

- Editör, metin düzenleyici demektir. Linux'ta bu amaçla kullanılan editörler :
  - pico
  - nano
  - vi
  - vim (\*)
  - emacs (\*)
  - Kate (\*)
  - Joe (\*)

# pico

- Linux'un en çok kullanılan metin editörlerinden biridir.
- vi editörüne göre kullanımı çok daha kolaydır.
  - ✓ Dosyayı kaydet: **Ctrl + O**
  - ✓ Dosyada arama yap: **Ctrl + W**
  - ✓ Satırı Kes/Sil: **Ctrl + K**
  - ✓ Kesilen metni yapıştır: **Ctrl + U**
  - ✓ Önceki sayfa: **Ctrl + Y**
  - ✓ Sonraki sayfa: **Ctrl + V**

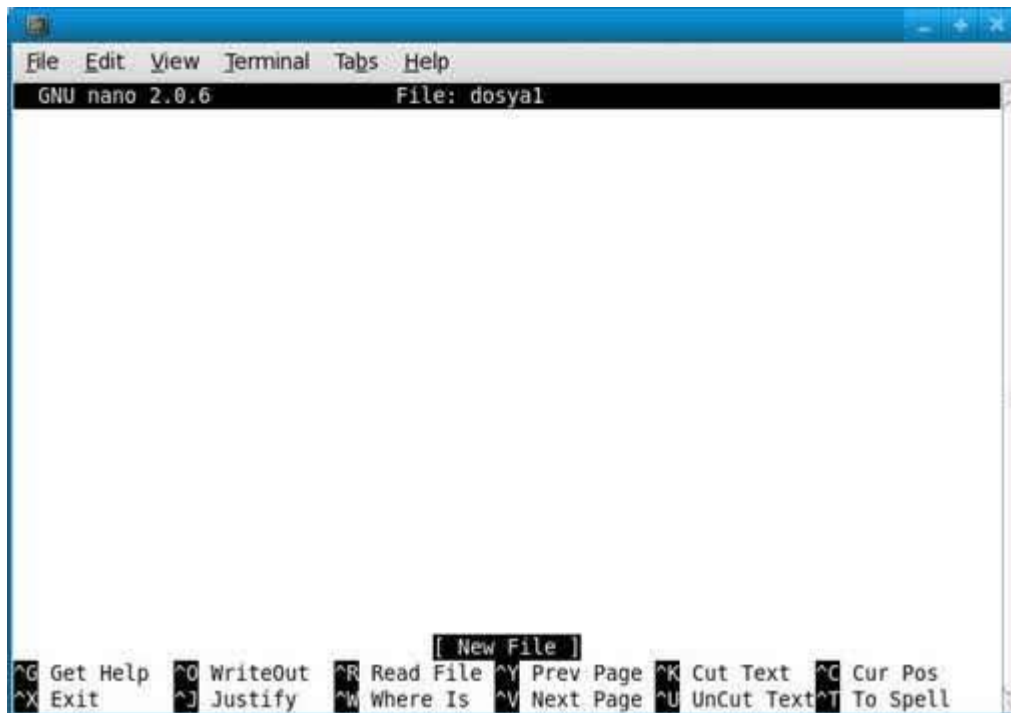
# pico



```
murat@hp: ~/Desktop
GNU nano 2.2.6      File: deneme      Modified
www.muratyazici.com
^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is       ^V Next Page      ^U UnCut Text    ^T To Spell
```

# nano

- Serbest yazılım lisansıyla pico' nun yerini alabilecek bir metin editörü ihtiyacı sonucu geliştirilmiştir.



Komutları,  
pico' nun komutlarına  
çok benzerdir.

# vi

- Yeni başlayanlar için karmaşık gibi görünse de oldukça kullanışlı güzel bir editördür.
- vi editörünü özel kılan, kullanıcının aynı anda hem komut çalıştırıp hem de yazı yazabilmesidir.
- vi editörünü ilk çalıştırdığınızda komut moduna girersiniz.
- **Yazma işlemine başlayabilmek için "i" harfine basmanız gerekir.**
- Yazma işlemi sonlandırıp tekrar komut moduna dönmek için **"Esc"** tuşuna basılmalıdır.

## Yazı (insert) moduna alma

Satırın başına metin ekleme

**I**

Kursörün olduğu yere metin ekleme

**i**

Kursörün üstüne yeni satır açmak

**O**

Kursörün altına yeni satır açmak

**o**

Kursörden sonra metin eklemek

**a**

Satır sonuna metin eklemek

**A**



## Metin silme ve düzenleme işlemleri (Komut modunda)

Kursörün üzerindeki satırı silme	<b>dd</b>
4 satır silme	<b>4dd</b>
Kursör üzerindeki karakteri silme	<b>x</b>
Satırı kopyalama	<b>yy</b>
5 satır kopyalama	<b>5yy</b>
Yapıştırma	<b>p</b>

## Arama ve diğer işlemler (Komut modunda)

Kursörün bulunduğu yerden itibaren dosya.txt dosyasını çalıştığınız metne kopyalamak için

**:r dosya.txt**

Geriye doğru metin arama

**?kelime**

İleriye doğru metin arama

**/kelime**

Bulunan kelimenin sonraki geçtiği yer

**n (next)**

Bir önceki çalıştırılan komutu geri alma

**u (undo)**

## Dosya kaydetme işlemleri (Komut modunda)

Değişiklikleri kaydetme

**:w**

Değişiklikleri kaydedip vi editöründen çıkma

**:wq!**

Değişiklikleri kaydetmeden çıkma

**:q!**

Dosyayı yeni dosya olarak kaydetme

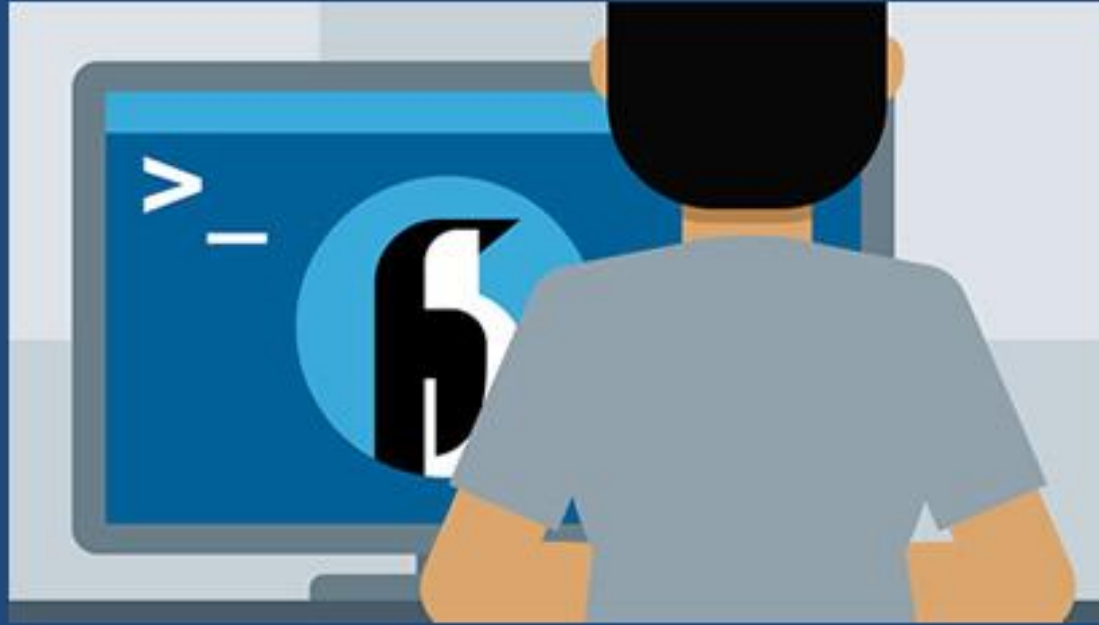
**:w dosya**

- **Ctrl + F** ile bir sonraki sayfaya geçilirken, **Ctrl + B** ile de bir önceki sayfaya geçilir.
- Editörde ilerlemek için ayrıca **yön tuşları** kullanılabilir. **Enter** tuşu kullanılarak da bir alt satıra geçilebilir.
- Hem Unix hem de Linux işletim sistemlerinde tanımlıdır.

# vim (vi improved)

- **vim**, Linux ve Unix kullanan, özellikle programcılar ve sistem yöneticileri tarafından tercih edilen **vi** editörünün gelişmiş versiyonudur.
- vi editöründen farklı olarak renklendirme özelliği de olup oldukça güçlü bir editördür.





## Linux Komutları - II

# Komutlar (Devam)

- Komut satırında **>**, **>>** **simgeleri** özel anlam taşır.

**ls -a > deneme** (ls -a komutunun çıktısı deneme isimli dosyaya yönlendirilir. Eğer dosya yoksa yaratılır, varsa da üzerine yazılır.)

**ls -a >> deneme** (ls -a komutunun çıktısı deneme isimli dosyaya eklenir, üzerine yazılmaz. Eğer dosya yoksa yaratılır.)

# sort komutu

- **sort** komutu ile bir dosyanın ya da farklı bir komuttan gelen bir girdinin içeriği sıralanabilir.
  - r → sıralama tersine çevrilir.
  - u → yinelenen satırları atarak biricikliği sağlar.



# sort komutu

## ■ sort dosya\_adi

```
[murat@fantom ~] $ more deneme
```

Murat

Zeynep

Burhan

```
[murat@fantom ~] $ sort deneme
```

Burhan

Murat

Zeynep

```
[murat@fantom ~] $ sort -r deneme
```

Zeynep

Murat

Burhan

# wc komutu

## ■ wc komutu

Bir dosyadaki veya komut çıktısındaki satır, karakter ve kelime sayılarını bulmak için kullanılır. Alabileceği bazı parametreler :

- l** → satır sayısını bulur. (line)
- w** → kelime sayısını bulur. (word)
- c** → karakter sayısını bulur. (character)

# | (Pipe) işareti

- Linux komut satırının en önemli özelliklerinden biri pipe'dır. Pipe (|) simgesi ile bir komutun çıktısını diğer bir komuta girdi olarak gönderebilirsiniz. Örneğin,

- cat kod | wc -l**

Normalde **cat** komutu **kod** isimli dosyanın içeriğini ekrana bastırırdı. Ancak yukardaki gibi kullanıldığında **cat** komutu çıktısını, yani kod isimli dosyanın içeriğini **wc** komutuna girdi olarak gönderir.

# | (Pipe) işareti

## Örnekler ;

- `head -5 deneme | tail -3`
- `ls -al | more`
- `sort -r dosya | tail -5 | wc -c`

# cal komutu (takvim)

- cal komutu

**cal** → Şu anki ay için bir takvim

**cal ay yıl** → Belirtilen yılın ilgili ayının takvimi

**cal yıl** → Belirtilen yılın bütün aylarını listeler

# grep komutu

- Linux'un en popüler arama komutlarından biridir. Dosyaların içerisinde arama yapar.

**grep** murat **deneme**

Bu komut **deneme** isimli dosyanın içinde **murat** kelimesi geçen tüm satırları listeler.

**grep** murat **/etc/passwd**

/etc/passwd dosyasında **murat** kelimesi geçen satırları listeler.

# grep komutu (devam)

**grep** boot /etc/default/\*

Bu komut **/etc/default** klasöründeki tüm dosyaların içinde **boot** kelimesi geçen dosyaları ve satırlarını listeler.

**grep -r** apache /etc/php5

komutu **apache** karakter dizisini **/etc/php5** dizinindeki dosyalarda ve alt dizinlerde rekürsif olarak arar ve içerisinde **apache** geçen dosya isimlerini ve satırlarını listeler.

**grep -n** murat /home/murat/Belgeler/\*

# grep için bazı önemli parametreler

- **-v:** Komutun davranışını tersine çevirir. Yani, aranılan kelimeyi içermeyen satırlar listelenir.
- **-i:** Arama sırasında büyük/küçük harf eşleştirmesi yapmaz.
- **-r:** Verilen dizinin alt dizinlerinde de arama yapar.
- **-n:** Aranan kelimenin geçtiği satır numarasını da gösterir.
- **-c:** Belirtilen dizinde aradığınız kelimenin kaç defa geçtiğini gösterir.
- **-l:** Şablona uygun satırların bulunduğu dosya adlarını listeler. (Sadece dosya isimleri listelenir, satırlar listelenmez.)



# find komutu

- Zaman zaman adının yalnızca bir kısmını bildiğiniz ama bulunduğu dizini bir türlü hatırlayamadığınız dosyalar ve dizinler olacaktır.
- Tek tek bütün dizinlere girip **ls** komutuyla bu dosya ya da dosyaları aramaktansa **find** komutunu kullanmak çok daha kullanışlı olacaktır.

```
find /home/murat -name bilgisayar
```

```
find /etc -name *.conf
```

```
find /etc -name *apache*
```

# find komutu (arama kriterleri)

- **-user murat** (Sahibinin adı murat olan dosya/dizinler)
- **-group myo** (Sahibi myo grubuna dahil olan dosya/dizinler)
- **-perm 755** (Erişim izni 755 olan dosya/dizinler)
- **-type f** (Dosyalar)
- **-type d** (Dizinler)
- **-size +100k** (100 Kbyte'dan büyük olan dosyalar)

# find komutu (arama kriterleri)

- **-ctime 3** (Tam 3 gün önce değişikliğe uğramış olan dosya/dizinler)
- **-ctime -7** (7 günden daha kısa bir süre önce değişikliğe uğramış olan dosya/dizinler)
- **-ctime +7** (7 günden daha uzun bir süre önce değişikliğe uğramış olan dosya/dizinler)
- **find / -name core -exec /bin/rm {} ";"**

# find komutu

- **find** komutuyla birlikte kullanılabilen kriterleri çeşitli mantık operatörleriyle birleştirebilirsiniz.

- **-a** : **ve**
- **-o** : **veya**
- **\!** : **değil**

**find** . -name "\*.txt" -a -size +1000k

Çalışma dizininde (.) ve varsa altındaki dizinlerde .txt uzantılı **ve** büyüklüğü 1000 Kbyte'dan fazla olan dosyaları listeler.

# ps komutu

- Sistem yöneticisinin sıklıkla kullandığı en önemli komutlardan bir tanesidir. Çalışmakta olan processleri (süreç) listeler.

**-aux** parametresi ile çok sık kullanılır.

**ps -aux** | grep murat

(Üstteki komut sonrasında sadece murat kullanıcısına ait süreçler ekranda listelenir.)

# ps komutu

## ps -aux komutu çıktısı

```
linux:~ # ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.1   0.0    448    76 ?        S      15:38   0:04 init
root         2   0.0   0.0      0      0 ?        SW      15:38   0:00 [keventd]
root         8   0.0   0.0      0      0 ?        SWN     15:38   0:00 [ksoftirqd_CPU0]
root         9   0.0   0.0      0      0 ?        SW      15:38   0:01 [kswapd]
root        10   0.0   0.0      0      0 ?        SW      15:38   0:00 [bdflush]
root        11   0.0   0.0      0      0 ?        SW      15:38   0:00 [kupdated]
root        12   0.0   0.0      0      0 ?        SW      15:38   0:00 [kinoded]
root        16   0.0   0.0      0      0 ?        SW      15:38   0:00 [mdrecoveryd]
root        19   0.0   0.0      0      0 ?        SW      15:38   0:00 [kreiserfsd]
root        71   0.0   0.0      0      0 ?        SW<     15:38   0:00 [lvm-mpd]
root       257   0.0   0.0      0      0 ?        SW      15:38   0:00 [khubd]
at        274   0.0   0.0   1400   104 ?        S      15:38   0:00 /usr/sbin/atd
root      389   0.0   0.0      0      0 ?        SW      15:38   0:00 [eth0]
root      445   0.0   0.0   1476      4 ?        S      15:38   0:00 /sbin/dhpcd -H -D -N
```

# kill komutu

- Çalışmakta olan bir processi sonlandırmak için kullanılır.
- **kill (-sinyal) PID** kullanımı mevcuttur.
  - ❑ **kill -9 1125** (1125 nolu processi sonlandırır.)

# Dosya Sıkıştırma Yöntemleri

- Zip ile sıkıştırma

- ❑ **zip -r deneme.zip dosyaAdı/dizinAdı**

- Zip dosyasını açmak

- ❑ **unzip deneme.zip**



# Dosya Sıkıştırma Yöntemleri

- tar ile gz olarak sıkıştırma
  - ❑ `tar cvzf deneme.tar.gz dosya/dizin`
- tar.gz dosyasını açmak
  - ❑ `tar xvzf deneme.tar.gz`

# Dosya Sıkıştırma Yöntemleri

- tar ile bz2 olarak sıkıştırma
  - ❑ `tar cvjf deneme.tar.bz2 dosya/dizin`
- tar.bz2 dosyasını açmak
  - ❑ `tar xvjf deneme.tar.bz2`

# Dosya Sıkıştırma Yöntemleri

- rar ile sıkıştırma (\*)

- ❑ **rar a -ap deneme.rar dosya/dizin**

- rar dosyasını açmak (\*)

- ❑ **unrar e deneme.rar**

- ❑ **unrar x deneme.rar**



# Linux Paket Kurulumu

# Ubuntu Paket Kurulumu

- **Paket Kurulumu**

`sudo apt-get install paketinAdı`

- **Kurulu Olan Bir Paketi Kaldırmak**

`sudo apt-get remove paketinAdı`

`sudo apt-get purge paketinAdı`

- **Paket Arşivini Güncellemek**

`sudo apt-get update`

# Ubuntu' da SSH Server Kurulumu

SSH (Secure Shell ), ağ üzerinden uzaktaki bir makineye erişip yönetebilmemizi sağlayan bir protokoldür. TCP/22. port üzerinden haberleşir.

```
# sudo apt-get install openssh-server
```

```
# ssh username@IPadress
```

# SSH Portunu Değiştirmek

```
# sudo pico /etc/ssh/sshd_config
```

dosyasını açıp Port 22 yazan satırda, 22 değerini boş olan bir port numarasıyla değiştirin.

```
# sudo /etc/init.d/ssh restart
```

komutuyla servisi yeniden başlattıktan sonra port numarası değişmiş olacaktır.

```
# ssh username@lpadress -p PortNo
```

# Ubuntu' da LAMP Server Kurulumu

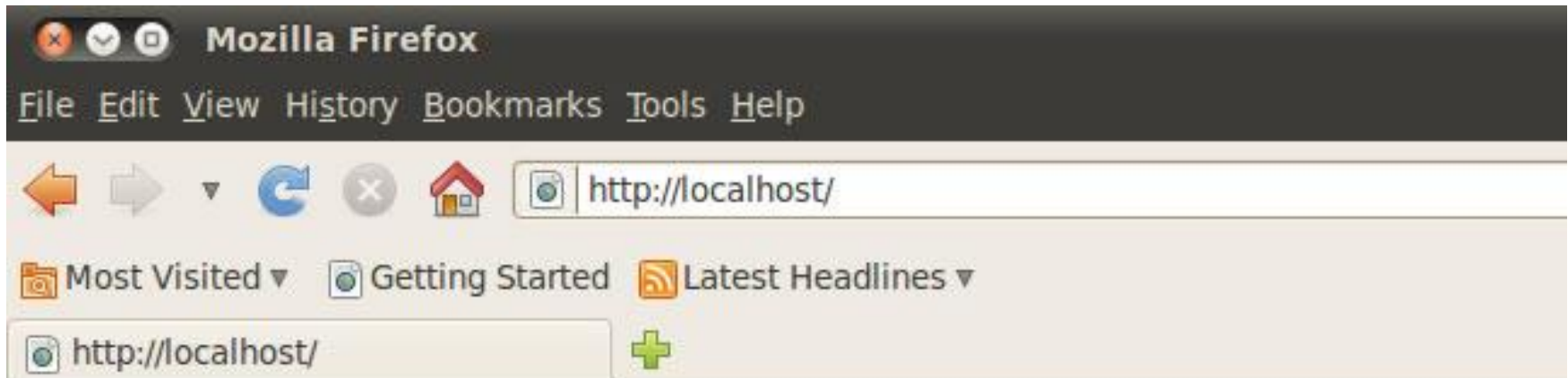
LAMP (Linux Apache MySQL PHP), Linux'ta PHP ve MySQL uygulamalarını geliştirmek için kurulması gereken paket yazılımı.

Bu paketleri ayrı ayrı kurabileceğimiz gibi, toplu halde kurmak için;

```
# sudo apt-get install lamp-server^
```



# /var/www/index.html



## It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

# Ubuntu' da PHPMyAdmin Kurulumu

```
# sudo apt-get install phpmyadmin
```

Kurulum tamamlandıktan sonra terminal ekranından,

```
# sudo pico /etc/apache2/apache2.conf
```

dosyasını açıp sayfanın sonuna aşağıdaki satırı ekleyin.

```
Include /etc/phpmyadmin/apache.conf
```

Ve son olarak Apache'yi yeniden başlatın.

```
# sudo /etc/init.d/apache2 restart
```

## Terminal

Package configuration

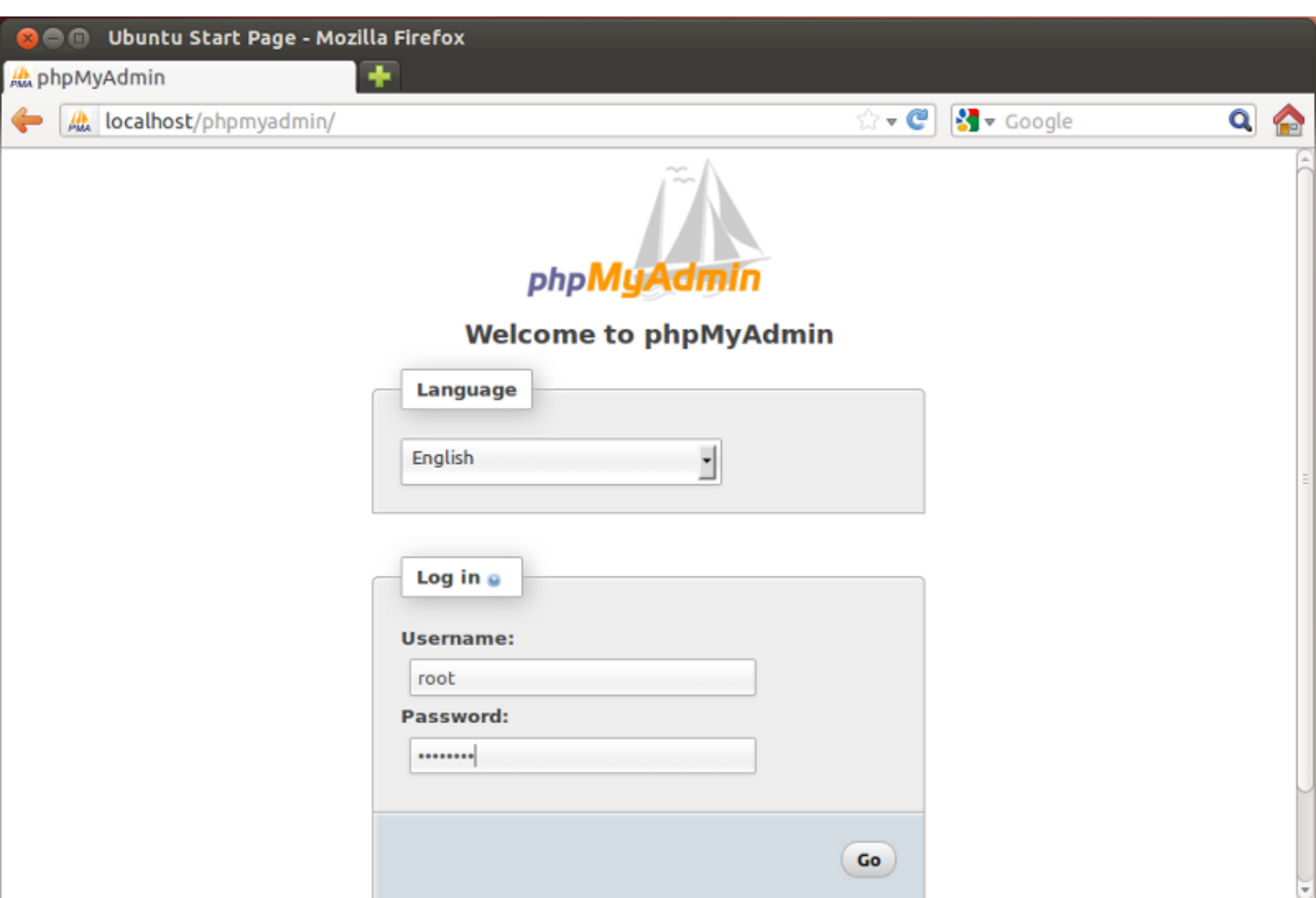
### Configuring phpmyadmin

Please provide a password for phpmyadmin to register with the database server. If left blank, a random password will be generated.

MySQL application password for phpmyadmin:

<Ok>

<Cancel>



phpMyAdmin



information\_schema  
mysql  
performance\_schema  
phpmyadmin  
test

localhost

Databases SQL Status Processes Privileges Export More

## General Settings

Change password

MySQL connection collation : utf8\_general\_ci

## Appearance Settings

Language : English

Theme / Style: pmahomme

Font size: 82%

More settings

## MySQL

- Server: Localhost via UNIX socket
- Server version: 5.5.22-0ubuntu1
- Protocol version: 10
- User: root@localhost
- MySQL charset: UTF-8 Unicode (utf8)

## Web server

- Apache/2.2.22 (Ubuntu)
- MySQL client version: 5.5.22
- PHP extension: mysqli

## phpMyAdmin

- Version information: 3.4.10.1deb1
- [Documentation](#)
- [Wiki](#)
- [Official Homepage](#)
- [Contribute](#)
- [Get support](#)
- [List of changes](#)

# KABUK PROGRAMLAMA

(shell programming- scripting)

# Kabuk Programlamaya Giriş

---

Her kabuğun kendine özgü programlama dili yapısı vardır. Bash kabuğu ise güçlü programlama özellikleriyle karmaşık programların rahatça yazılmasına izin verir. Mantıksal operatörler, döngüler , değişkenler ve modern programlama dillerinde bulunan pek çok özellik bash kabuğunda da vardır ve işleyiş tarzları da hemen hemen aynıdır.

Genellikle, bir programı oluşturacak olan komutlar bir dosyaya yazılırlar ve ardından bu dosya çalıştırılır. Herhangi bir editör yardımıyla yazılan program, daha sonra kabuk altında çalıştırılır. Bir kabuk programı diğerlerini çalıştırabilir. Bu düzende kabuk programlarını daha karmaşık komutların biraraya gelmiş ve yapılaşmış haline benzetebiliriz.

Bash'in en büyük dezavantajı, derlenerek çalıştırılan dillere göre (C, C++ gibi) daha yavaş olması, sistem kaynaklarını biraz daha fazla tüketmesidir.

## Kabuk Programları

Kabuk programları, bir veya birden fazla Linux komutunu tutan dosyalardır. Bu dosya yaratıldıktan sonra doğrudan dosyanın ismi girilerek veya dosya isminden önce '.' karakteri getirerek çalıştırılabilir. Bir kabuk programı, çalıştırma bitini 1 yapmak suretiyle "çalıştırılabilir" hale getirilir. chmod komutu yardımıyla bir programı çalıştırılabilir yapmak için ,

```
$ chmod +x komut-ismi
```

yazılabilir. Bundan sonra programın ismi yazılıp enter tuşuna basıldığı zaman bir program Linux komutuymuş gibi çalışacaktır.

```
$ cat calistir
echo -n "Tarih : "
date
$ chmod +x calistir
$ calistir
Tarih : Sun Dec  8 07:11:51 EET 1996
```

Yukarıdaki örnekte "calistir" isimli iki satırlık bir kabuk programının önce içeriği ekrana yazıldı, ardından çalıştırılacak duruma getirildi ve çalıştırıldı.

Kabuk programları yazarken dosyanın işlevini ve her satırdaki komutun veya komut kümesinin ne amaçla kullanıldığını gösteren açıklama satırları kullanmak işe yarar. Bir açıklama eklemek için satır başına (veya boş satıra) # işareti eklenir ve ardından istenilen cümle girilir. # işaretinden sonraki tüm satır kabuk tarafından gözardı edilir. Aşağıdaki programda komut öncesinde yeralan açıklama satırı, komut hakkında bilgi veriyor.

```
# gunzip komutu dosya acmak için kullanılır.  
gunzip sistem.gz
```

Yorum satırı, komutun sonuna da eklenebilir.

```
ps -aux          # sistem surecleri hakkında ayrıntili bilgi..
```

Bir kabuk altında çalışırken başka bir kabuk için yazılmış bir programı çalıştırmak mümkündür. Örneğin tcsh altındasınız ve daha evvel bash kullanarak yazdığınız bir programı çalıştırmak istiyorsunuz. Önce bash yazarak kabuk değiştirmeli, ardından programı çalıştırmalı, ve tekrar tcsh'a dönmelisiniz. Tüm bunları otomatik olarak yaptırabilirsiniz. Programın en başına #! karakterini, ardından programın çalışacağı kabuğun patikasını yazın. Örneğin #!/bin/bash komutunu programın en üstüne eklerseniz bu program bash kabuğu altında çalışacaktır.

## Değişkenlerin Kullanımı

Bir değişkene değer atandığı anda sistem tarafından tanınır. Değişkenler alfabetik veya nümerik karakterlerden oluşabilirler fakat bir değişken sayısal bir değer ile başlayamaz. Bunların dışında değişken isminin içinde "\_" karakteri de bulunabilir. Bir değişkene değer ataması "=" işareti yardımıyla yapılır.

```
$ mesaj="aksama yemege geliyorum"
```

İçeriği olan bir değişkene başına "\$" işareti konularak ulaşılır. Aşağıda, echo komutu yardımıyla bir değişkenin içeriği ekrana basılıyor.

```
$ echo $mesaj  
aksama yemege geliyorum  
$ echo yarin $mesaj  
yarin aksama yemege geliyorum
```

Aynı mesajı değişken kullanmadan da görüntüleyebiliriz.

```
$ echo "Aksama yemege geliyorum"  
Aksama yemege geliyorum
```

## Giriş/Çıkış İşlemleri

Bir kabuk programı çalışırken kullanıcıdan klavye yardımıyla bilgi girmesi sağlanabilir. Bu tür işlemler için tanımlanan read komutu klavyeyi okur ve aldığı bilgiyi bir değişkene atar. Aşağıdaki komutları içeren program yardımıyla klavyeden okunan değer ekrana yazılıyor. **echo** komutundan sonra birden fazla değişken grubu veya hem değişken, hem de dizi kullanılabilir.



```
echo Bir sayi giriniz..
read sayi
echo Girilen sayi : $sayi
```

Bazı durumlarda girilen değer özel karakterleri içerebilir. Bu durumda istenmeyen bazı sonuçların doğması kaçınılmaz olur. Aşağıdaki örneği bir dosya içine yazın ve dosyayı çalıştırdıktan sonra "\*" tuşuna basın.

```
echo Bir karakter giriniz
read a
echo Girdiginiz karakter : $a
```

**echo** komutundan gelecek bir yıldız işareti, bulunduğunuz dizindeki tüm dosyaları listeleyecektir.

## Aritmetik İşlemler

bash kabuğunda matematiksel işlemlere büyük sınırlamalar getirilmiştir. Tamsayı değişkeni dışında matematiksel değişken kullanmak için bu işlemler için geliştirilmiş ve kolaylıklar sağlayan **awk** veya **bc** kullanabilirsiniz.

Aritmetik işlemler için **eval** komutunu veya bash kabuğu altında yerleşik (builtin) komut olan **let** komutunu kullanabilirsiniz. Aşağıda **let** komutunun kullanımı görülüyor.

```
$ let "degisken=aritmetik islem"
```

Bu örnekte iki sayı çarpılıp çıkan sonuç başka bir değişkene yazılıyor.

```
$ let "carpim=2*7"
$ echo $carpim
```

Aritmetik değişken tanımlamanın diğer bir yolu da **typeset** komutu kullanmaktır.

```
$ typeset -i sonuc      (sonuc degiskeni bir dogal sayi icerecek)
$ a=100 ; b=56          (iki komutu ayirmak için ; kullanilabilir)
$ sonuc=a*b
$ echo $sonuc
5600
```

Normal olarak bash, kesirli ve noktalı işlemleri yapamaz. Bunun için **bc** kullanabilirsiniz. **Bc**, çok yüksek duyarlılığa sahip bir hesap makinasıdır.

```
$ a=3.749
$ b=22.34
$ echo "$a*$b" | bc
83.752
```

## ***if-else Kalıbı ve Kontrol İşlemleri***

Hemen her programlama dilinde olan `if` kalıbı bir Linux komutunun çalışmasını kontrol (`test`) eder. `if` komutu yerleşik bir komuttur. `if` komutunun ardından gelen Linux komutu çalıştırılır ve komutun çıkış durumu (`exit status`) gözönüne alınarak ardından gelen `then` deyimiyle birlikte devamı işletilir. Genellikle komutun iki türlü çıkış durumu olacağından `else` komutunun ardından gelen komut zinciri, diğer çıkış durumunda çalıştırılır. Her `if`, bir `fi` komutuyla bitmelidir. Aşağıda `if-then-else` komutunun örnek sözdizimi görülüyor.

```
if linux komutu
then
    komut1
    komut2
    ...
else
    komut1
    komut2
    ...
fi
```

`if` komutu genellikle kendine `test` komutu ile birlikte kullanım bulur. Bu komut yardımıyla mantıksal işlemler yapılabilir, sayılar ve hatta diziler karşılaştırılabilir. Anahtar sözcük olan `test`'ten sonra opsiyonlar ve/veya karşılaştırılacak olan değerler yazılır. Her opsiyon bir mantıksal işleme karşılık gelir. Örneğin `-lt` opsiyonu ilk girilen aritmetik değişkenin ikinci değerden küçük olup olmadığını denetler. Benzer şekilde `=` opsiyonu da iki karakter kümesinin eşitliğini kontrol eder. Aşağıda `test` komutunun örnek kullanımı yeralıyor.

```
$ test 5 -eq 3
$ a="linux"
$ test $a="linux"
```

komutun işletilmesinin ardından kabuğa bir değer döndürülür. Bu değer komut başarılı olarak işletilmişse 0, değilse 1'dir. Son çalıştırılan tüm Linux komutlarının çıkış değeri `$?` değişkeninde tutulur. `test` komutunun çıkış değeri de bu yolla öğrenilebilir.

```
$ sayi=4
$ test $sayi -eq 4
$ echo $?
0
$ test $sayi -lt 2
$ echo $?
1
```

test komutu yerine parantezler de kullanılabilir. Yukarıdaki iki örnek, parantez kullanılarak şu şekilde yazılabilir:

```
$ [ $sayi -eq 4 ]
$ [ $sayi -lt 12 ]
```

Dikkat edilmesi gereken bir nokta, köşeli parantez kullanırken araya boşlukların eklenmesidir. Parantezler başlı başına bir komut olarak görüldüklerinden sağında ve solunda en az bir boşluk bırakılmalıdır. test komutunda sıkça kullanılan diğer seçenekler şunlardır:

Aritmetik karşılaştırma	
-gt	büyük
-lt	küçük
-ge	büyük eşit
-le	küçük eşit
-eq	eşit
-ne	eşit değil
Dizisel karşılaştırma	
-z	boş dizi
-n	tanımlı dizi
=	eşit diziler
!=	farklı diziler
Dosya karşılaştırması	
-f	dosya var
-s	dosya boş değil
-r	dosya okunabilir
-w	dosyaya yazılabilir
-x	çalıştırılabilir dosya
-h	sembolik bağlantı
-c	karakter aygıt
-b	blok aygıt
Mantıksal karşılaştırma	
-a	VE
-o	VEYA
!	DEĞİL

if komutunun test ile birlikte kullanılabildiğini daha önce belirtmiştik. Aşağıda bununla ilgili küçük bir örnek yeralıyor.

```
#!/bin/bash
echo "0 ile 20 arasında bir sayı seçin"
read sec
if [ $sec -lt 10 ]
then
    echo "Secilen sayı tek basamaklı"
else
    echo "Secilen sayı çift basamaklı"
fi
```

Her if komutu bir fi ile son bulmalıdır.

### **case Kalıbı**

Birkaç alternatif arasından seçim yapmak için kullanılan bir komut olan case, bir eşleştirme gördüğü anda belirli bir komut kümesini işleme sokar. case yapısı case komutu ile başlar, eşleştirilecek olan anahtar sözcük yazılır ve seçenekler alt alta, her seçeneğe ait olan komutlarla birlikte belirtilir. Tüm yapı esac komutu ile son bulur.

```
case anahtar-sozcuk in
    secenek1)
        komutlar
        ;;
    secenek2)
        komutlar
        ;;
    *)
        komutlar
        ;;
esac
```

Seçenekler arasında özel karakterler (\*, [, ], ? gibi) kullanılabilir. Hiçbir eşleme yapılmadığı zaman \*) seçeneği değerlendirilecek ve buna bağlı olan komutlar işletilecektir. \* kullanımı isteğe bağlıdır. Aşağıda case komutuna ilişkin kısa bir örnek veriliyor.

```
#!/bin/bash

clear
echo "1. ekrani temizle"
echo "2. sistemdekileri görüntüle"
echo "3. dizindeki dosyalari goster"

echo -n "Secenegi giriniz : "
read secenek

case $secenek in
```

```

1)
  clear
  ;;
2)
  w
  ;;
3)
  ls -al
  ;;
*)
  echo Hatali secenek
esac

```

## Döngüler

Diğer hemen tüm programlama dillerinin en büyük gücü olan döngü işlemlerine kabuk altında da izin veriliyor. Burada programcı tarafından en çok kullanılan 2 döngü tipi anlatılacaktır: `while` ve `for`. `while` komutu her döngüde bir denetleme mekanizmasını harekete geçirirken `for` döngüsü bir listenin elemanlarını sırayla seçer.

### *while-do Döngüsü*

Döngü bloğu `while` anahtar kelimesiyle başlar, ardından gelen koşul sağlandığı sürece döngü işletilir. Önce koşulun sağlanıp sağlanmadığına bakılır. Döngüden çıkabilmek için mutlaka döngü içindeki koşul ifadesinin değerini yanlış yapacak bir durum oluşmalıdır, aksi halde sonsuz döngü oluşur.

```

while kosul ifadesi
do
    komutlar
done

```

`if` komutuyla birlikte kullanılan `test` komutu, `while` döngüsünde koşul ifadesi olarak da yer alabilir. Aşağıda 1'den 100'e kadar sayan ve ekrana basan bir döngü görülüyor.

```

#!/bin/bash
deger=0
while [ $deger -lt 100 ]
do
    deger=$((deger+1))
    echo $deger
done

```

Yukarıda kullanılan (( ve )) karakterleri arasına matematiksel bir işlem getirilebilir. Bu özellik `bash` kabuğuna özgüdür.

### *for-do döngüsü*

Bir liste dahilindeki tüm değerlere sırayla erişimi sağlar. `for` komutundan sonra yeralan liste sırayla kullanılır ve herbirisi için döngü çalıştırılır. Listenin sonuna gelindiğinde ise döngüden çıkarılır.

```
for degisken1 in deger1 deger2 ... degerX
do
    komutlar
done
```

Aşağıdaki örnek bu döngüyü kullanarak ekrana bir dizi kelime yazıyor. Döngü boyunca akasya, elma ve visne kelimeleri "agac" değişkenine kopyalanıyor ve her döngüde bu değişkenin içerdiği bilgiler ekrana yazılıyor.

```
for agac in akasya elma visne
do
    echo $agac
done
```

for-do döngüsü, dosya isimleri üzerinde yapılan işlemlerde de büyük kolaylıklar sağlar. Bunun için özel karakterlerden yararlanmak da olasıdır. Örnek olarak \* karakteri o anki çalışma dizini içindeki tüm dosyaları seçer.

```
for a in * ; do
    file $a
done
```

## Örnek Kabuk Programı

Sistem görevlisinin en çok kullandığım komutlardan birisi

```
ps -aux grep -i xxx
```

komutudur. Bu satır yardımıyla çalışan xxx isimli program hakkında daha detaylı bilgi elde edilebilir. ps komutu detaylı bir süreç listesini ekrana verirken çıktı doğrudan grep komutuna yönlendirilir ve sadece istediğimiz bilgi ekranda görünür. Fakat her zaman aynı uzun satırı tekrar tekrar yazmaktansa bu satırı bir dosyaya gönderip, dosya adını komut satırından çalıştırmak zamandan tasarruf sağlayacaktır. Biz de öyle yapalım ve aşağıdaki satırları ``goster" isimli dosyaya yazalım.

```
#!/bin/bash
if [ $# = 1 ]
then
    ps -ax | grep -i $1
else
    ps -ax
fi
```

Ardından dosyayı PATH değişkeninin işaret ettiği dizinlerden

```
/usr/local/bin
```

altına yerleştirip çalıştırılabilir olması için

```
chmod +x /usr/local/bin/goster
```

komutunu uygulayın. Kendi yazdığınız dosyaları

```
/usr/local
```

veya

```
~/bin
```

dizini altına kopyalamanız bunların derli toplu olarak tek bir dizinde her an erişilebilir şekilde durmaları açısından önem taşır.

Dosyaya biraz daha yakından bakalım. İlk satır, bu dosyanın /bin/bash programı tarafından çalıştırılacağını gösterir. İkinci satır yardımıyla komut satırı üzerinde kaç tane opsiyon olduğu bulunur. \$# çevresel değişkeni her biri TAB veya boşluk karakteri ile ayrılmış komut satırı opsiyonları sayısını verir. Aşağıdaki komutta toplam 3 opsiyon vardır.

```
$ ls --8bit -F -b
```

\$# komutuna benzer şekilde \$1, \$2, \$3 ... değişkenleri de opsiyonları verir. Yukarıdaki komutta \$1, \$2, \$2 değişkenleri sırasıyla

```
--8bit
-F
-b
```

değerlerini alırlar. Komut satırından çalıştırılan komut ise \$0 değişkenine atılır. Yukarıdaki örnekte \$0 değişkeni ls değerini tutacaktır.

goster dosyasına tek opsiyon yollayacağız. Bu opsiyon da hakkında detaylı bilgi alacağımız süreç olacaktır. Dosya içinde \$1 değişkenini ps komutuna yerleştirmek için :

```
ps -ax | grep -i $1;
```

yazılır. Bundan sonra komut satırında, örneğin:

```
$ goster bash
```

girilirse \$1 değişkeni bash e eşit olacak ve bu da

```
ps -ax grep -i bash
```

eşdeğer komutunu çalıştıracaktır. Komut olarak sadece goster girilirse \$# değişkeni 0'a eşit olacağından dosyada yeralan ve hiç bir filtreleme yapmayan

```
ps -ax
```

komutu çalışır.

