

# LINUX TEMEL BİLGİLER

İsa Dumanođlu

Çukurova Üniversitesi

Fen-Edebiyat Fakóltesi Fizik Bölümü

# Linux Nedir?

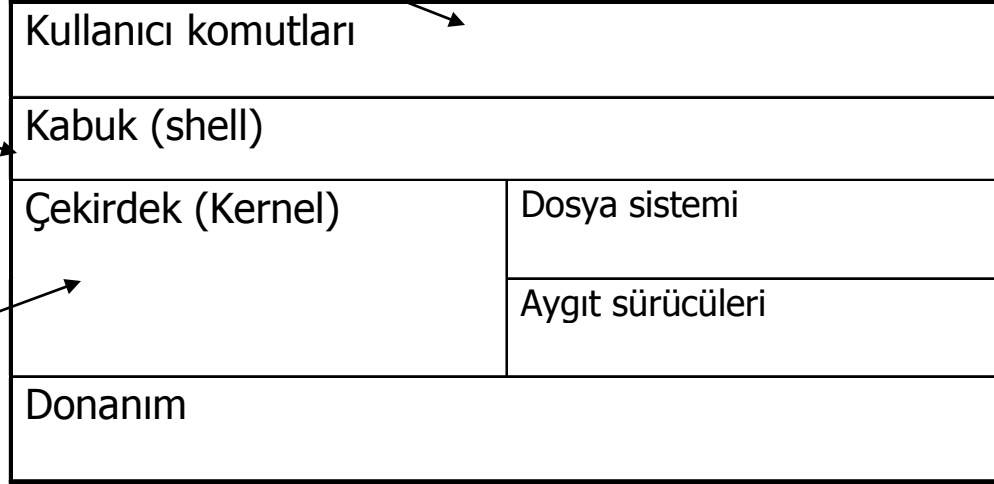
- 32/64-Bit Unix-benzeri İşletim Sistemi
  - sed, awk, and grep gibi unix araçları
  - C, C++, Fortran, Smalltalk, Ada gibi derleyiciler
  - Ssh (telnet), sftp(ftp), ping, traceroute **network araçları**
- Çok kullanıcı, çok görevli, çok işlemcili
- X Windows GUI
- Diğer işletim sistemleriyle birlikte bulunabilme
- Birçok platformda çalışabilme
- Kaynak kodlarını içermesi
- Bedava

# The Linux System

Kullanıcı komutları, çalıştırılabilir programları ve komut dizilerini (scripts) içerir.

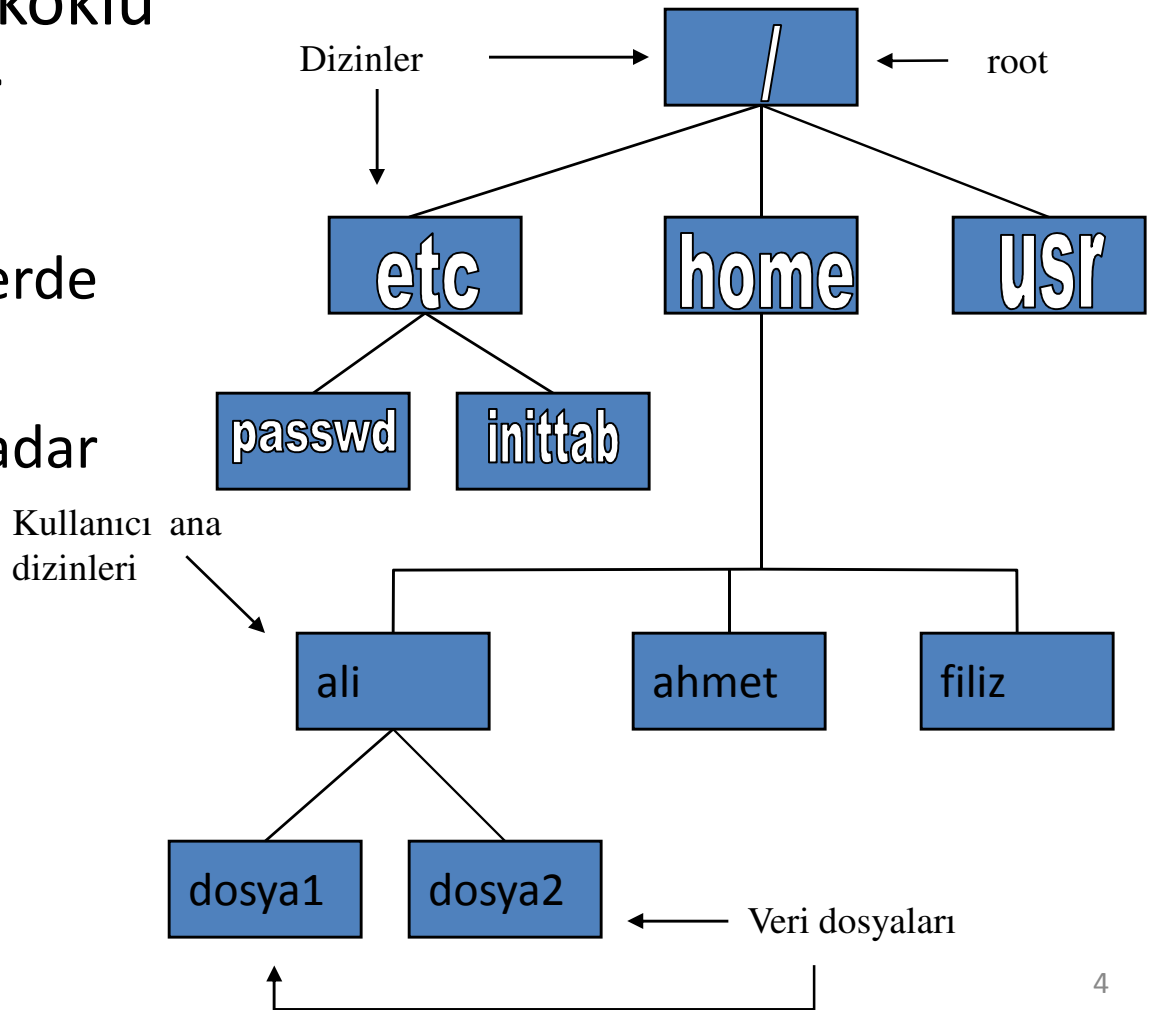
Kabuk kullanıcı komutlarını yorumlar. Komutları bulup çalıştırmakla sorumludur. Birkaç değişik kabuk vardır. Bash en popüler olanıdır.

Çekirdek sistemin geri kalan kısmı için donanım kaynaklarını idare eder.



# Linux'in Temel Dosya Yapısı

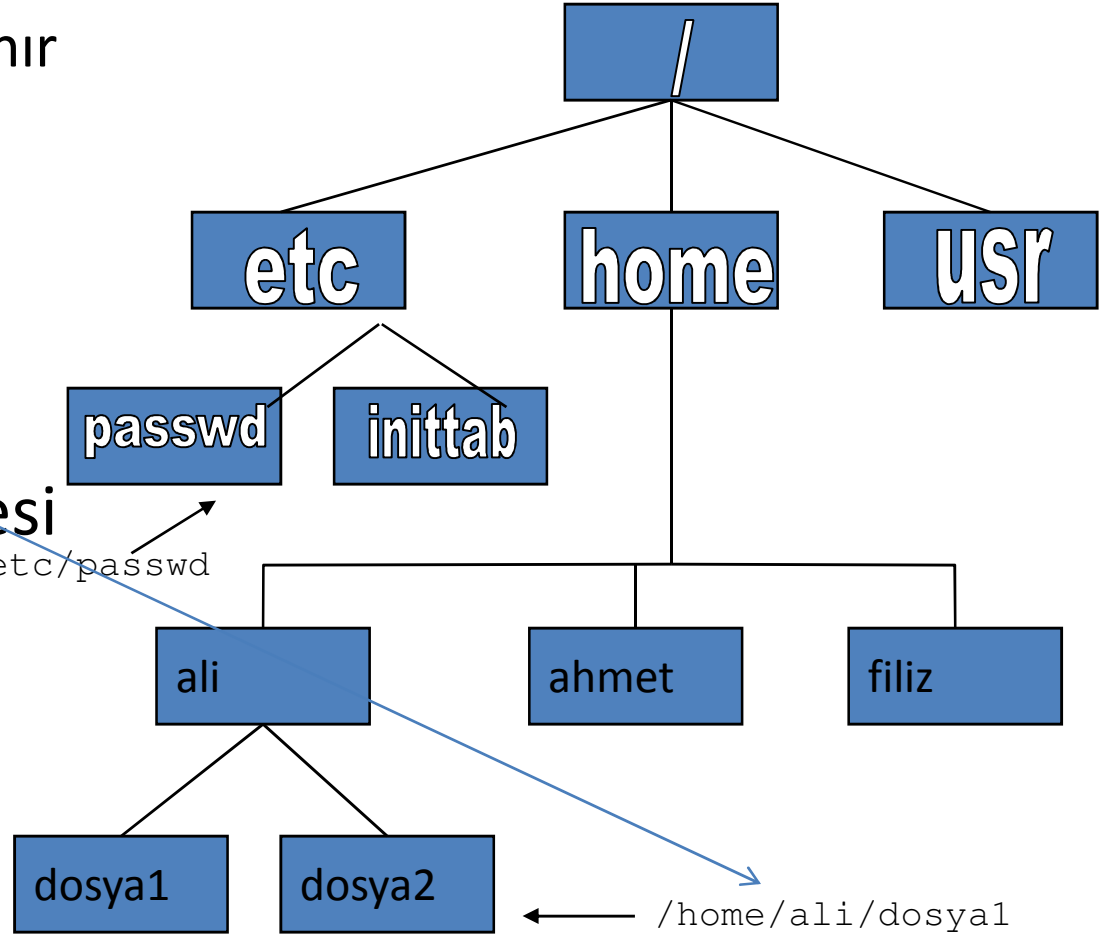
- Linux dosyaları tek-köklü hiyerarşik yapıda bir dosya sistemidir.
  - Veri dosyaları dizinlerde saklanır.
  - Dizinler istenildiği kadar alt dizin içerebilir.



# Dosyaların İsimlendirilmesi

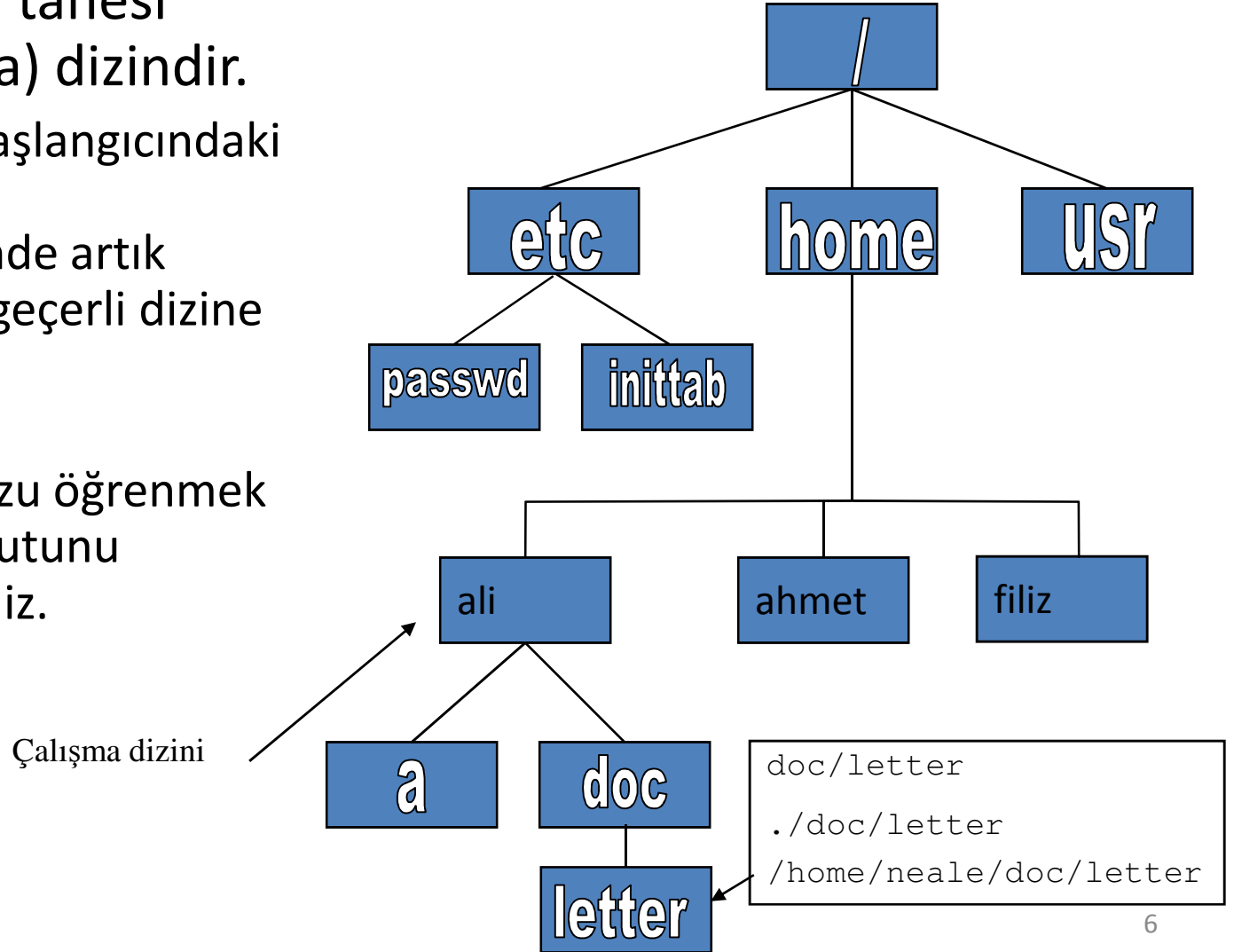
- - Root dizininden başlanır
  - Her dizin isimlendirilir

- *Buna dosya yolu (pathname) diyoruz. dosyanın adresi gibi de düşünülebilir.*



# Geçerli (çalışılan) Dizin (Current Directory)

- Dizinlerden bir tanesi geçerli (çalışma) dizindir.
  - dosya yolu başlangıcındaki / karakteri düşürüldüğünde artık dosya yolları geçerli dizine göredir.
  - Hangi dizinde bulunduğunuzu öğrenmek için `pwd` komutunu kullanabilirsiniz.



# Özel Dosya ve Dizinler

- `/home` – bütün kullanıcı dizinleri burada saklanır
- `/bin, /usr/bin` – sistem komutları bu dizinlerdedir.
- `/sbin, /usr/sbin` - sistem-yöneticisi tarafından kullanılan komutlar bu dizindedir.
- `/etc` - her çeşit konfigürasyon dosyaları
- `/var` - logs, spool gibi dizinler
- `/dev` - aygıt (device) dosyaları
- `/proc` – özel sistem dosyaları

## Sisteme Giriş Çıkış ve Kullanıcı Bilgi Komutları (ssh,login, logout, exit, passwd,whoami)

ssh [isa@lxplus.cern.ch](ssh://isa@lxplus.cern.ch)

Passwd:\*\*\*\*\*

**passwd** kullanıcı şifresini değiştirmek için kullanılır.

**exit** ve **logout** kullanıcı hesabından çıkmamızı sağlar.

**whoami** açık bir hesabın kullanıcılarının kim olduğunu öğrenmemizi sağlar.

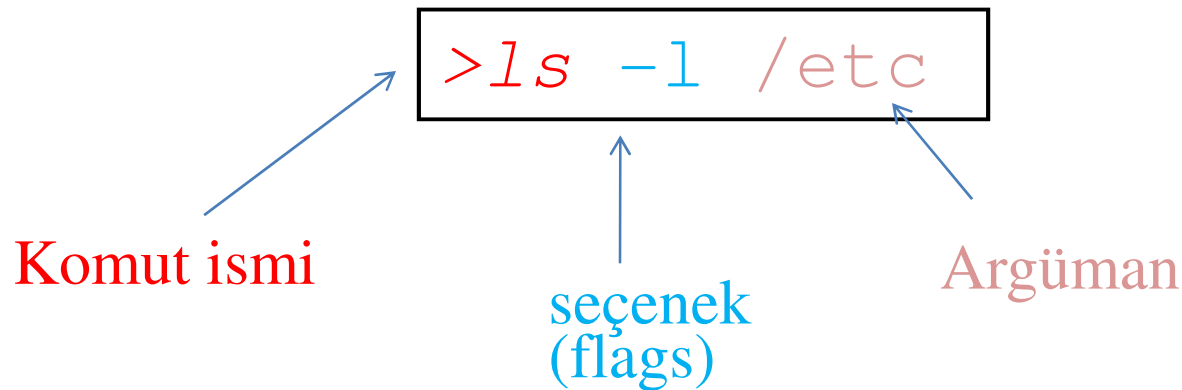


# Komut Seçenekleri

- Komut seçenekleri komutların belirli bir düzeye kadar kontrol edilmesini sağlar.
- Anlaşma:
  - genellikle bir eksi (-) işareti ve bir harf ile birlikte kullanılır (“-l”).
  - Arasına iki eksi işareti (-- ) ve bir kelime ile birlikte kullanılır (“--help”).
  - Bazen de komutun arkasından hiçbirşey gelmez.

# Linux Komut Yapısı

- Bir komutu çalıştırmak için ismi ve varsa seçenekleri ve argümanları yazılır.
- Genel komut yapısı aşağıdaki gibidir.



# Joker Karakterler

- **?** bir karakterin yerine geçer  
file?.root file1, file2, files,... ? İşaretinin bulunduğu yerde herhangi bir karakter olabilir.
- **\*** bir grup karakterin yerine geçer file1.\*  
file1.cc, file1.gz, file1.bmp burada \* karakterinin bulunduğu yere herhangi bir karakter grubu gelebilir.
- Bu iki karakter göreceğimiz komutların bir seferde birçok dosya üzerine uygulanmasını sağlar. Sadece dosya ve dizin isimleriyle birlikte kullanılır. Komutlarla kullanılmaz.
- **[abc]** içerisinde a veya b veya c olanlar.
- **[^abc]** içerisinde a ve b ve c olmayanlar
- **.** çalışılan dizini temsil eder.
- **..** çalışılan dizinden bir önceki dizini temsil eder.

# Dosyaları Listelemek (ls)

- `>ls` geçerli dizindeki dosyaları listeler
- `ls /home/isa` /home/isa dizinindeki dosyaları listeler
- `>ls -l` dosyaları listelerden tarih, boyut, vs gibi detayları da gösterir
- `>ls -lt` detayları dosyaların yaratıldığı tarihe göre listeler
- `>ls -ltr` tarih sırasına göre tersten listeler
- `>ls -a` gizli dosyaları da göster. .uzantı şeklindeki dosyalar gizli dosyalardır. Örneğin .tcshrc gibi.
- `>ls -l *.cc` uzantısı cc olan dosyaları detaylı olarak listeler.

# Dizin Yaratma (mkdir)

- *>mkdir* **dizin1** geçerli olan dizin içerisinde dizin1 diye bir dizin yaratır.
- *>mkdir* **/home/isa/dizin1** /home/isa dizininde dizin1 isminde bir dizin yaratır.

# Dizin Değiştirme (cd)

- `>cd dizin1` geçerli dizini dizin1 olarak değiştirir.
- `>cd /home/isa/analiz/` geçerli dizini /home/isa/analiz dizini olarak değiştirir.

# Dosya ve Dizin Kopyalama (cp)

- `>cp dosya1 dosya2 dosya1'in dosya2 adında bir kopyasını yaratır.`
- `>cp -r /home/isa/analiz /home/isa/isler/analiz2 analiz dizininin analiz2 olarak kopyasını yapar.`
- `>cp -rf isa/dizin1/* isa/dizin2/ dizin1 deki bütün dosyaları dizin2'ye kopyalar. dizin2'de dizin1'deki dosyalarla aynı isimli dosyalar varsa bunları sormadan dizin1'deki dosyalarla değiştirir. Sormasını istiyorsanız -f seçeneğini kaldırmalısınız.`

# Dosya ve Dizin silme (rm)

- `>rm dosya1` dosya1 isimli dosyayı siler
- `>rm -ri /home/isa/dizin1 dizin1` dizin1 isimli dizini siler ve onay için sorar
- `>rm -rf /home/isa/dizin1 dizin1` dizin1 isimli dizini ve bunun içerisinde bulunan diğer dizinleri de siler. Onay için sormaz.
- `>rmdir /isa/analiz` isa dizinindeki analiz dizinini siler. Bu işlemin gerçekleşebilmesi için analiz dizinin boş olması gerekir.



# Dosya veya Dizin Taşıma (mv)

- `>mv dosya1 dosya2` dosya1'i dosya2 olarak değiştirir. Dosyanın isminin değiştirilmesi demektir.
- `>mv -r /home/isa/analiz /home/isa/isler/analiz2`  
analiz dizinini isa dizininin altındaki isler dizinine taşır.
- `>mv -rf isa/dizin1 isa/dizin2` dizin1 isimli dizini dizin2 olarak taşır. Daha önceden dizin2 diye bir dizin varsa ve bunun içerisinde dizin1'deki dosyalarla aynı isimli dosyalar varsa bunları sormadan dizin1'deki dosyalarla değiştirir. Sormasını istiyorsanız `-f` seçeneğini kaldırmalısınız.

# Linux'te Yardım

- Linux'te yardımın (HELP) karşılığı man (manual)'dir.
  - içerisinde keyword içeren tüm komutları görmek için `>man -k <keyword>` komutunu yazınız.

Herhangi bir komut ile ilgili yardım almak içinse

```
>man <komut>
```

veya

```
>info komut
```

yazınız.

`>komut -help` şeklinde de komut hakkında yardım alınabilir.

- man komutu ekranda bilgileri sayfa sayfa getirir. Geri gitmek için **b**\_ileri gitmek için **f** tuşunu, ileri doğru incelemek için boşluk çıkmak içinse **q** tuşunu kullanınız.

# Dosyanın Yaratılma Tarihini Değiştirmek(touch)

- *>touch* dosya

dosya isimli dosyanın yaratılma tarihini bugünün tarihiyle değiştirir. Böyle bir dosya yoksa kendisi 0 bitlik bir dosya yaratır.

# Sistem ve Kullanıcılar Hakkında Bilgi Görüntüleme

(pwd, whoami,w,finger,hostname,logname)

- **>pwd** Hangi dizinde çalışıldığını(geçerli dizini) gösterir.
- **>whoami** çalışılmakta olan hesabın kime ait olduğunu gösterir.
- **>finger ahmet** ahmet isimli kullanıcı hakkında bilgi verir.
- **>w** sistemde çalışan kullanıcıları ve hangi uygulamaları çalıştırdıkları hakkında kısa bilgi görüntüler.
- **>hostname** login yapılan makinanın ana-makina ismini gösterir.
- **>logname login** yapılan kullanıcı ismini görüntüler.  
logname=whoami.

# Dosya İçeriği Görüntüleme (more, less, cat)

- Her üç komut benzerdir. Fakat bazı önemli ayrıcalıkları da vardır.
- `more/less/cat` bir dosyanın içeriğini görüntülemek için kullanılır.
- `>more/cat/less text1.txt` text1.txt isimli dosyanın içeriğini görüntüler.
- `less`'in `more`'dan önemli ayrıcalığı dosya görüntülenirken sayfayı ileri ve geriye hareket imkanı sağlamasıdır. `Ctrl-V` ileri `Ctrl-Y` geri.
- `cat` dosyaları birleştirmek için de kullanılabilir.
- `cat dosya1 dosya2 > dosya3`

# Dosya İçeriği Karşılaştırma (comm,diff, diff3)

- *>comm* dosya1 dosya2

dosya1 ve dosya2'nin birleşim kümesini gösterir.

- *>diff* dosya1 dosya2

dosya1 ve dosya2'nin farklılıklarını gösterir.

- *diff3* 3 dosyayı karşılaştırmak için kullanılır.

- *>diff3* dosya1 dosya2 dosya3, dosya1, dosya2 ve dosya3 arasındaki farklılıkları gösterir.

# Dosya İçeriği ve Bilgi Görüntüleme (head, tail, wc,file)

- `>wc tex1.txt` tex1.txt dosyasında bulunan satır, kelime ve karakterlerin sayısını gösterir.
- `>head ana.C` ana.C dosyanın başlangıcından birkaç satır gösterir.
- `>tail ana.C` ana.C dosyanın sonundan birkaç satır gösterir.
- `>file histo.cc`  
dosyanın tipi hakkında bilgi verir (c programı, text dosyası, ... gibi)

# Disk Kullanımı (du)

- **du** komutu disklerin nasıl kullanıldığını öğrenmek için kullanılır. Özellikle de sistem yöneticisi için önemlidir. Aşırı büyüyen disklerin belirlenmesi sistem performansı açısından önemlidir. Genellikle diskin kullanımı hakkında özet bilgiler edinmek için kullanışlıdır.

>du

- 8 *./xemacs*
- 12 *./kde/Autostart*
- 8 *./kde/share/config/session*
- 152 *./kde/share/config*
- 4 *./kde/share/servicetypes*
- 24 *./kde/share/mimelnk/application*
- 28 *./kde/share/mimelnk*
- 4 *./kde/share/applnk-redhat/System/MoreSystem*
- 8 *./kde/share/applnk-redhat/System*
- 4 *./kde/share/applnk-redhat/Internet/More Internet*
- 8 *./kde/share/applnk-redhat/Internet*
- 4 *./kde/share/applnk-redhat/Office*
- 4 *./kde/share/applnk-redhat/Accessories*
- 4 *./kde/share/applnk-redhat/Documentation*

Görünür boyut

Klasör ismi



# Disk Kullanımı (df)

- **df** komutu bir disk üzerindeki mevcut kapasite ve boş alanların (ayrılmış bölümlerin) miktarını öğrenmek amacıyla kullanılır.

- *>df*

Filesystem Mounted on	1K-blocks	Used	Available	Use%
/dev/sda2	25806300	7993732	16501664	33% /
/dev/sda1	256666	55567	187847	23% /boot
/dev/sda14	103351416	3735936	94365440	4% /tmp
/dev/sda12 /usr/vice/cache	2063504	1518972	439712	78%
/dev/sda3	10317860	802480	8991260	9% /var

# Disk kullanımı (quota)

- **quota** komutu kullanıcının disk alanının ne kadarlık bir kısmının kullanıldığını görüntüler. Bu komutu kullanmak için kullanıcı için kota konulmuş olmalıdır.
- *>quota*
- Disk quotas for user isa (uid 22461):
- Filesystem blocks quota limit grace files quota limit grace
- /dev/sda14 20 31005425 41340566 5 0 0

## Bilgi Komutları (top, date, xclock,cal)

- **>top** hangi sürecin sistemin kaynaklarını ne kadar kullandığını gösterir.
- **>date** tarihi ve saati öğrenmek/değiştirmek için kullanılır.
- **>date 012116352009** tarihi 21/01/2009 ve saati 16:35 olarak kurar.
- **>xclock** saat programıdır. Bazen X-window uygulamalarının çalışıp çalışmadığını test etmek için de yararlıdır.
- **>cal** takvimi göstermek için kullanılır.

# Terminal Ekranını Silme ve X-terminal Açma (clear, reset, xterm)

- *>clear* ekranı temizler ve komut satırını da ekranın en üst kısmına taşır.
- *>reset* ekranı hem temizler hem de yeniden başlatır.
- *>xterm &* yeni bir X-terminal açar.

# Komut ve Dosyaların Konumlarının Belirlenmesi

## (which,whereis,locate,slocate)

- **whereis** bir komutun hangi kopyalarının olduğunu öğrenmek için kullanılır.  
>**whereis** ls
- **which** bir komutun dosya yolunu (hangi dizinde bulunduğunu) gösterir.  
>**which** cp
- **locate/slocate** sistemde bulunan dosyaları bulmayı sağlar. Locate veritabanı güncellendikten sonraki dosyaları bulabilir. Eğer bir dosya güncellemeden sonra yaratılırsa locate bu dosyayı bulamayabilir. Sistem açıldığında veri-tabanı (database) otomatik güncellenir. Güncelleme yapılmak istenirse **updatedb (root yetkili olmak gerekebilir)** komutu kullanılabilir.
- >**locate** dosya1

# Güvenli Bağlantı ve Dosya Transferi (ssh, telnet, sftp,ftp,scp,rcp)

- `scp SourceFile user@host:directory/TargetFile`
- `scp user@host:directory/SourceFile TargetFile`
- `scp isa@lxplus.cern.ch:mysetup.csh .`
- `lxplus.cern.ch` makinasındaki `isa` isimli kullanıcının `public` dizininden `mysetep.csh` dosyasını geçerli dizine kopyalar.
- `scp hello.c isa@lxplus.cern.ch:public`
- `hello c` isimli dosyayı `lxplus.cern.ch` makinasındaki `isa` isimli kullanıcının `public` dizinine kopyalar.
- Her iki durumda da `isa` isimli kullanıcının şifresini sorar.
  
- `sftp isa@lxplus.cern.ch` dosya transferi yapmak için kullanılır.
- Sifre girildikten sonra `ls`, `rm`, `cd`, `chown` gibi bazı kabuk komutları da kullanılabilir.
- `get dosya1` `dosya1`'i `uzak makinadan(lxplus)` çalışılan makinanın geçerli dizinine getirir.
- `put dosya2` `çalışılan makinadaki` geçerli dizinde bulunan `dosya2`'yi `uzak makinaya` koyar.
- `mget *.csh` uzantısı `csh` olan bütün dosyaları getirir.
- `mput *.cc` uzantısı `.cc` olan bütün dosyaları `uzak makinaya` koyar.
- `ls *` uzak makinadaki `bütün dosyaları` görüntüler.
- `exit` işlemler bittikten sonra `sftp`'den çıkmak için kullanılır.
- `telnet`, `rcp` ve `ftp`, `ssh`, `rcp` ve `scp`'nin kodlama olmadan kullanılan çeşitleridir. Güvenli kabul edilmedikleri için yerlerini güvenli karşılıklarına bırakmışlardır.

# Sistemle İlgili Bilgi Görüntüleme (echo, free, history)

- **echo** ekrana mesaj yazdırmak veya bir çevresel değişkenin içeriğini yazdırmak için kullanılır.
- **free** hafızanın hangi oranlarda kullanıldığını gösterir.
- **history** komut geçmişini(daha önce kullanılan komutları) gösterir.

# Süreçleri Ön/Arka Planda Çalıştırma (fg,bg,&)

- **&** bir sürecin arka-planda(background) çalışmasını sağlar. Terminali blok edecek bir süreç bu şekilde çalıştırılırsa terminal blok olmaz. Örneğin :
- **>emacs text1.txt &** gibi.
- **>fg** arka-planda çalışan bir süreci ön-plana getirir.
- **>bg** ön-plana gelen bir süreci tekrar arka-plana gönderir.
- **Emacs** çalışırken **Ctrl-Z** yapıp sonra da **bg** yazmayı deneyin.



# Dosyalarda Karekter Katarı Arama (grep)

- `>grep -i "print" program.c`

program.c içerisinde print karekter dizisi var mi diye bakar. `-i` küçük büyük harf ayırımı yapma anlamındadır.

- `>grep "cout" *.C`

C uzantılı dosyalar içerisinde cout karekter dizisi var mi diye bakar.

# Dosya Arama (find)

- `>find <dizin ismi> -name < aranılan>`
- `> find /home/isa/ -name dosya.txt -print`  
isa dizini altında dosya.txt dosyasını arar.
- `> find . -name "*.html" -print`  
Geçerli dizinden itibaren arama yapar.
- `> find / -name dosya.txt -print`
- Tüm dizinlerde dosya.txt dosyasını arar.

# Çalışan Süreçleri Görüntülemek (ps)

- `>ps`  
çalışan süreçleri görüntüler.
- `>ps aux | more`  
çalışan bütün süreçleri görüntüler.
- `>ps -au` süreçleri görüntülerken kullanıcı bilgisini de gösterir.
- `> ps -au | grep isa`
- `>ps -au | grep emacs`

# Süreç

& işareti sürecin arka planda çalışmasını sağlar.

```
> sleep 10h &
```

```
[1] 6718
```

```
> ps -af
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	6718	6692	0	14:49	ttyp0	00:00:00	sleep 10h

İş Numarası

süreç ID (ID)

Ana Proses ID'si

Komut adı

# Çalışan Bir Süreci Öldürmek (kill)

- **kill** çalışan bir süreci durdurur. Bir program donarsa veya bir program gereksiz yere sistemin kaynaklarını kullanıyorsa ve normal yolla durdurulamıyor ise bu süreci **kill** ile öldürmek gerekir.
- Bir root kullanıcısı istediği her süreci **kill** komutuyla öldürebilir. Diğer kullanıcılar sadece kendilerine ait süreçleri öldürebilir.
- Bir süreci öldürmek için sürecin ID'sini bilmek gerekir. Sürecin ID'sini öğrenmek için **ps** komutundan yararlanılır.
- **>kill 205**
- 205 PID numaralı süreci öldürür.
- **>kill -9 205**
- Bu komutta 205 PID numaralı süreci öldürür. Bir önceki komuta kıyasla işlemi gerçekleştirmek de daha karardır.
- **root olarak bir süreci öldürürken çok dikkatli olmak gerekir. root'un her süreci öldürme yetkisi olduğu için yanlışlıkla sistemi çalıştıran süreç öldürülürse sistemin çökmesine yol açabilir.**
- Bir süreç bu şekilde durdurulamıyorsa tek yol sistemi yeniden başlatmaktır.

# Dosya Sıkıştırma (gzip, gunzip, bzip2)

- **gzip** dosyaları sıkıştırır
- **>gzip** dosya.\*
- dosya.\* isimli dosyaları dosya.\*.gz olarak sıkıştırır.
- **>gunzip** dosya.\*.gz
- sıkıştırılan dosyaları tekrar açar(eski haline dönüştürür).
- **bzip2** gzip'e oranla daha çok sıkıştırma sağlar.
- **>bzip2** dosya.\*
- Dosya.\*.bz2 olarak sıkıştırır.
- **>bzip2 -d** dosya.\*.bz2
- Sıkıştırılmış dosyaları açar.

# Arşiv Komutları

- `tar` – arşivleri işler
  - Arşiv dosya ve dizin setinden oluşan tek bir dosyadır. Arşiv dizinlerin orjinal hiyerarşik yapısını korur.
  - `>tar -czf analiz.tar /home/isa/analiz`  
/home/isa/analiz dizinindeki dosyaları analiz.tar isimli tek bir dosya olarak arşivler. **z** seçeneği dosyaların arşivlenirken sıkıştırılmasını sağlar.
  - `>tar -xzf analiz.tar`  
analiz.tar isimli dosyayı arşivden alıp açar.

# Yönlendirme (piping)

- `ls -ltr | more`  
`ls` komutunun çıktısı bir ekranda görüntülenemeyecek kadar büyükse çıktısı `more`'a yönlendirilerek sayfa sayfa görüntülenebilmesi sağlanır.
- `>ps -au | grep emacs`
- `ps`'in bulunduğu süreçler `grep` komutuna yönlendirilir `grep` de bunların içerisinde `emacs` geçen süreçleri bulur.
- `find ./ -name "*.c" | xargs grep -i "histo1"`
- Uzantısı `.C` olan dosyaları arıyoruz. Bulduğumuz dosya isimlerini `xargs` komutuna yönlendiriyoruz. `xargs` bu dosya isimlerini alıp `"grep -i histo1 <dosya1>...<dosyan>"` şeklinde komut satırı yapar. Bu komutta dosyaların içerisinde "histo1" geçen yerleri bulup ekranda gösterir. `-i` seçeneği bu aramanın büyük-küçük harf ayırımı yapılmaksızın gerçekleştirilmesini sağlar.



# Lakap, Diğer Ad (alias, unalias)

- `alias` - komutları kısaltmak veya daha hatırlanabilir isimler vermek için kullanılır.
- `>alias erase= 'rm'`
- `>alias grep= 'grep -I'`
- `unalias` alias ile tanımlanan lakapları iptal eder.
- `>unalias erase`
- `>unalias grep`

# Sisteme Veri Aygıtlarının Bağlanması (mount)

- **>mount**
- Sisteme mount edilmiş tüm kütük sistemlerini ve bunların özelliklerini göstermek için kullanılır.
- Sisteme disk, cd-rom , disket sürücü veya flash-disk'i bağlar.

```
>mount -t vfat /dev/hda5 /mnt/winC
```

hda5 nodunu sisteme bağlar./mnt dizininde winC diye bir dizin olmalıdır.

```
>mount /dev/cdrom /mnt/cdrom
```

cdrom'u sisteme bağlar

```
>mount /dev/fd0 /mnt/floppy
```

Disket sürücüyü sisteme bağlar.

```
>mount /dev/sda1 /mnt/usb
```

Usb diski sisteme bağlar.

# Sembolik Link (ln)

- **ln** İki dosya ya da dizin arasında sembolik bağ kurar.
- **\$ ln -s /home/yedek /tmp/yedek**
- Böylece, gerçekte /home altında yer alan yedek dizini, aynı zamanda **/tmp** altında da varmış gibi davranacaktır.

# -Düğümler ( nodes )

- /dev/hda <<Birinci disk
- /dev/hdb <<ikinci disk
- /dev/hda1 <<Birinci diskin birinci bölümü
- /dev/hdc5 <<üçüncü diskin beşinci bölümü
- /dev/sda <<Birinci SCSI disk
- /dev/sdb2 <<İkinci SCSI diskin ikinci bölümü
- /dev/cdrom <<CD sürücü
- /mnt/fd0 <<Disket sürücü

# Sistemden Veri Aygıtlarını Ayırmak (umount)

• **umount** : Sisteme bağlı olan sürücüleri ayırmak için kullanılır.

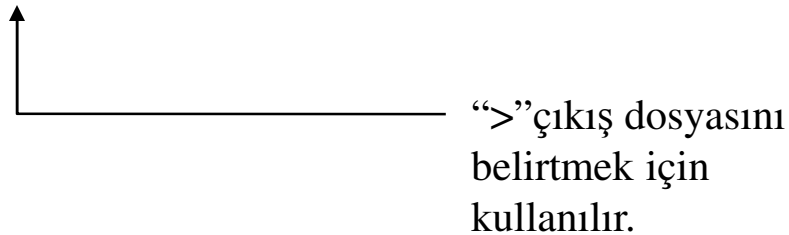
```
>umount /mnt/hda5
```

```
>umount /dev/cdrom
```

```
>umount /dev/fd0
```

# Çıkışı Yönlendirmek

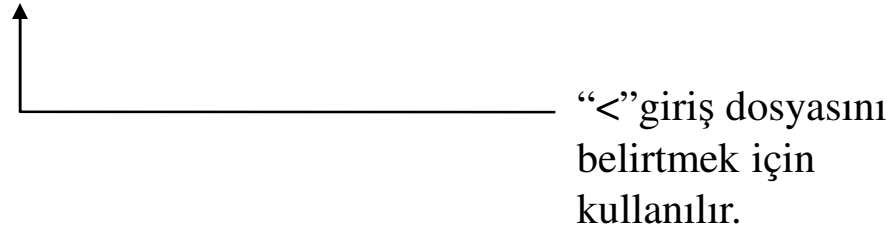
- Bir komutun çıktısı bir dosyaya yönlendirilebilir.
- ls -l >output



- ls -l >>output
- Çıktıyı var olan dosyaya ekler.

# Giriş Yönlendirmek

- Bir komutun girişi aşağıdaki gibi bir dosyadan alınabilir.
- wc <input



# Çevresel Değişkenler

- Çevresel değişkenler bir programı ve kabuk fonksiyonlarını kontrol eden global ayarlardır. Global kabuk değişkenleri olarak da adlandırılırlar.
- Çevresel değişkenlere değer atama:
  - VAR=/home/fred/doc
  - export TERM=ansi
  - SYSTEMNAME=`uname -n`



# Bazı Önemli Çevresel Değişkenler

- **HOME**
  - Ana dizininiz (login yaptığınızda başlangıç olarak gelen dizin, genellikle “~” olarak kısaltılır)
- **TERM**
  - Çalıştırdığınız terminalin tipi (örneğin vt100, xterm, ve ansi gibi)
- **PWD**
  - Geçerli dizin
- **PATH**
  - Bir komutun aranacağı dizinlerin listesi

# PATH Çevresel Değişkeni

- Komutların nerede bulunacağını kontrol eder
  - **PATH** bir dizi dizinin : işaretiyle ayrıldığı dosya yollarıdır.  
Örneğin :  
`PATH=/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:/home/scully/bin`  
şeklinde.
  - Eğer komut / karakterini içermezse kabuk komutu **PATH** ile verilen dizinlerde arar ve ilk bulduğu yerde komutu çalıştırır.

```
>echo $PATH
```

```
>echo $HOSTNAME
```

# Çevresel Değişkenlerin Ayarlanması (set, env, setenv, export)

- *set ve export* bu şekilde boş olarak kullanıldığında tanımlanmış çevre değişkeninin sahip olduğu değeri gösterir.
- *>set*
- *>export*
- *>export PATH=\$PATH:/root/bin ....*  
PATH değişkenine /root/bin dizinini de ekler.  
Böylece bu dizinde bulunan komutlar bir terminal ekranında yazılarak çalıştırılabilirler.
- *>setenv* tcsh kabuğunda *export*'la eşdeğerdir.

# Kullanıcı Değiştirme

- `su <kullanıcı-adı>`
  - Kullanıcı hesaplarını değiştirmek için kullanılır. Bu komut yazıldığında sistem bu kullanıcıya ait şifreyi sorar. Süreç tamamlandığında yeni kullanıcı olarak login yapmış olursunuz. Bu hesaptan `exit` komutu ile eski hesaba dönülebilir.
- `su`

kullanıcı adı olmadan kullanıldığında root olarak login yapılmasını sağlar. Şifre girildiğinde de root olarak login yapılmış olur.

Root'un kendisi kullanıcı hesaplarının birinden diğerine şifresiz login yapabilir.

# Kural !!!

- Zorunluluk olmadığı sürece root olarak login yapmayınız.
- root sistemin süper-kullanıcısıdır.
- Sistem ile ilgili herşeyi yapma hakkına sahiptir.
- Fakat bu güç bilinçsiz kullanıldığında tehlike demektir.
  - Doğal olarak herşeye erişim hakkı vardır.
  - Dikkatsiz kullanım sistemi bozabilir.
- root kurulum sırasında tanımlanmak zorunda olan tek kullanıcıdır.
  - Kurulumdan sonra yapacağınız ilk iş root şifresini değiştirmektir.
  - İkinci iş günlük kullanımlar için normal kullanıcı hesapları açmaktır.

# Sistemin Kapatılması/Yeniden Başlatılması (reboot, shutdown, halt)

- **reboot** ve **halt** sistemin yeniden başlatılmasını sağlar. Bu komutları konsoldan çalıştırmak için root yetkisinde olmanız gerekebilir.
- *>reboot*
- *>halt*
- **shutdown** sistemi kapatmaya yarar
- *>shutdown -h now* sistemi kapatır.
- *>shutdown -r now* sistemi yeniden başlatır.

# Kullanıcılar ve Gruplar

- Kullanıcılar **UID** denilen bir kullanıcı tanımlı numarası ile tanımlıdır. UID 0 ile 4 294 967 295 (X'FFFFFFFF') arasında bir tamsayıdır. **UID=0** olan kullanıcılar **süper-kullanıcı** hakkına sahiptirler.
- Kullanıcılar gruplar içerisine yerleştirilmiştir. Bunlar da **GID** denilen grup tanımlı numarası ile tanımlıdır. **GID** numaraları da 0 ile 4 294 967 295 arasında bir tamsayıdır.
- Kullanıcı kopyalarından sakınmak için sistemin kullanıcılara **UID** vermesini sağlayınız.
- Kullanıcı **UID** ve **GID** bilgilerinizi öğrenmek için **id** komutunu kullanabilirsiniz.

```
>id  
uid=22461(isa) gid=1399(zh) groups=1399(zh),1095065472
```

# Kullanıcılar ve Gruplar

- Gruplar fonksiyonel yerler ve sorumlulukları tanımlar.
- Kullanıcı topluluklarının dosya paylaşmasını sağlarlar.
- Bir kullanıcı birkaç gruba ait olabilir.
- Hangi gruplara ait olduğunuzu öğrenmek için **groups** komutunu kullanabiliriz.



# Dosya Kullanma İzinleri

- Her dosyaya
  - Bir kişi sahiptir.
  - Bir gruba aittir.
  - sahibi, grup üyeleri ve diğerleri için kesin bir erişim izni vardır.
  - Başlangıç izinleri **umask** ile belirlenmiştir.

# Dosya Kullanım Hakları

- Her kullanıcıya ait 3 bilgi tanımlıdır :
  - uid (login name), gid (login grup) ve üyelik "groups" listesi:
    - *uid* kim olduğunuzu tanımlar (isim ve numara).
    - *gid* ise ait olduğunuz başlangıç login grubunuzdur.
    - üyelik grup listesi grup izinlerine göre erişebileceğiniz dosya gruplarıdır.

# Dosya Erişim Hakları

- Linux 3 çeşit erişim hakkı sağlar :
  - **Okuma** – okuma izni olan kullanıcılar dosyayı okuyabilir veya bu bir dizin ise dizindeki dosyaların listesine bakabilir.
  - **Yazma** – yazma izni olan kullanıcılar bir dosyaya yazabilir veya bu bir dizinse oraya dosya koyabilir.
  - **Çalıştırma** – çalıştırma hakkı olan kullanıcılar bir dosyayı çalıştırabilir veya bu bir dizinse bu dizindeki dosyaya bakabilir.

# Dosya Kullanım Hakları

- `>ls -l` komutu ile dosyaların kullanım haklarına bakılabilir.

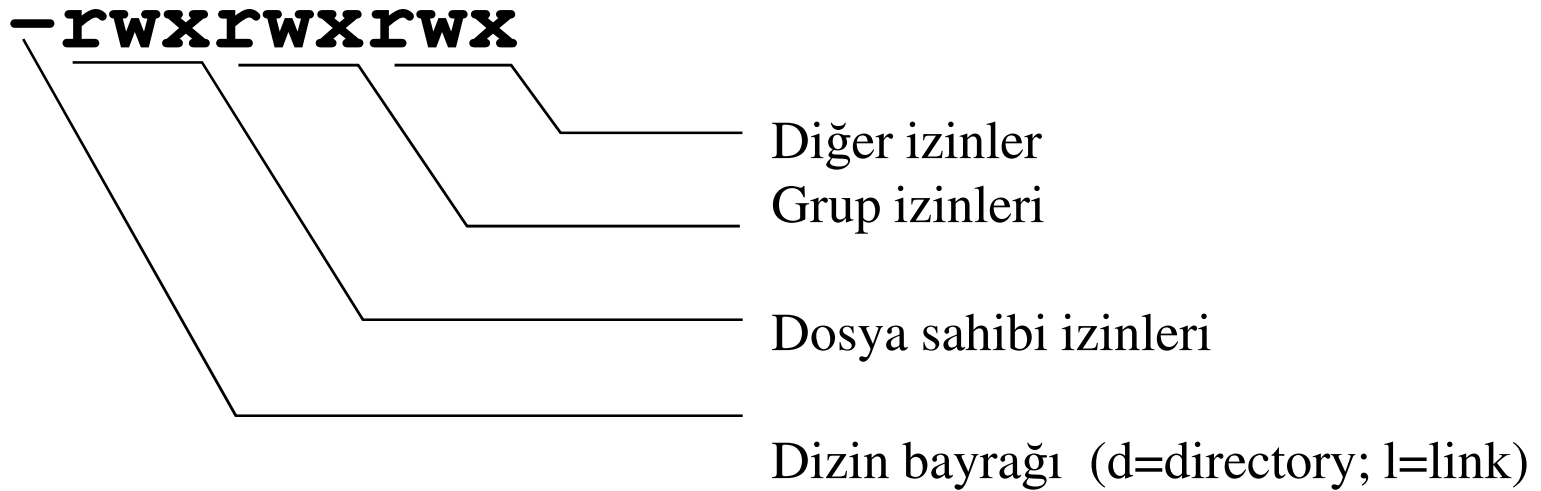
```
-rwxrwxr-x  1 rvdheij  rvdheij    5224 Dec 30 03:22 hello
-rw-rw-r--  1 rvdheij  rvdheij     221 Dec 30 03:59 hello.c
-rw-rw-r--  1 rvdheij  rvdheij    1514 Dec 30 03:59 hello.s
drwxrwxr-x  7 rvdheij  rvdheij    1024 Dec 31 14:52 posixuft
```

Kullanım  
hakları

Dosya  
sahibi

Grup

# Dosya İzinlerinin Gösterimi



# Dosya Erişim Haklarını Değiştirmek (chmod)

Dosya erişim hakları 8'lik rakamlarla kodlanmıştır.

<b>400</b>	<b>sahibi okur</b>			
<b>200</b>	<b>sahibi yazar</b>			
<b>100</b>	<b>sahibi çalıştırır</b>			
<b>040</b>	<b>grubu okur</b>	<b>0</b>	izin yok	<b>4</b> sadece okuma
<b>020</b>	<b>grubu yazar</b>	<b>1</b>	çalıştırma	<b>5</b> okuma ve çalıştırma
<b>010</b>	<b>grubu çalıştırır</b>	<b>2</b>	sadece yazma	<b>6</b> okuma ve yazma
<b>004</b>	<b>diğerleri okur</b>	<b>3</b>	yazma ve çalıştırma	<b>7</b> okuma, yazma ve çalıştırma
<b>002</b>	<b>diğerleri yazar</b>			
<b>001</b>	<b>diğerleri çalıştırır</b>			

Daha mutlak

```
chmod 755 dosya # sahibi=rwx grup üyesi=r-x diğerleri=r-x
chmod 500 dosya1 # sahibi=r-x grup üyesi=--- diğerleri=---
chmod 644 dosya2 # sahibi=rw- grup üyesi=r-- diğerleri=r--
chmod +x dosya # herkes için çalıştırma hakkı eklemek
chmod o-r dosya # diğerleri için okuma iznini kaldırma
chmod a+w dosya # herkes için yazma izni eklemek
```

Hatırlaması kolay

# Dosyaların Kullanıcılarının/Gruplarının Değiştirilmesi (chown,chgrp)

- **chown**: Dosyanın sahibini ve grubunu değiştirir.
- Bir dosyanın sahibini ve grubunu değiştirme yetkisi sadece root kullanıcısına aittir. Aşağıdaki komutları gerçekleştirebilmek için root olarak login yapmanız gerekir.
- **chown** komutundan sonra kullanıcı ve grup isimlerinin arasına . (nokta) konularak dosyanın grubu da değiştirilebilir.
- **>chown isa.users dosya1**

**Yukarıdaki komut yardımıyla dosya1 adlı dosyanın kullanıcısı isa, grubu users olarak değiştirilir.**

- **>chown isa bilgi.dat**
- Bilgi.dat dosyanın sahibini isa isimli kullanıcı olarak ayarlar.
- **>chgrp users bilgi.dat**
- bilgi.dat dosyasının grup sahibi olarak users verildi.

# Editörler-1

- Editörlerle bir program veya bir metin yazarken sık sık kayıt yapmalısınız. Böylece bir elektrik kesintisi olduğunda veya sistem bir şekilde bloke olduğunda emekleriniz boşa gitmemiş olur.
- Değişik editör seçenekleri mevcuttur.
- Fare tuşları ile menüden seçmek yerine kısa-yol tuşlarını kullanmaya alışmak X-window seçenekleri olmadığı zaman sizi editör-özürlü yapmaktan kurtarır.
- Kullanıcılar kullandıkları editörler için fanatiktirler.
- Genelde `>editor dosya-adi` şeklinde kullanılır. Fakat önce editörü çalıştırıp dosya daha sonra da açılabilir.
  - `vi` standart UNIX editörü.
  - `pico/nano` basit gösterge uyumlu metin editörü.
  - `xedit` X-windows metin editörü.
  - `the` XEDIT benzeri editör.
  - `nedit` X-window Motif metin editörü.
  - `emacs` X-window temelli güçlü bir editördür. Fakat Linux'in ön tanımlı editörü olmadığı için ayrıca kurmak gerekebilir.



# Editörler-2

## (pico/nano)

- pico/nano dosya-adi.
- ^ işareti pico/nano editör menüsünde Ctrl (kontrol) tuşu anlamındadır.
- **Ctrl-G** yardım.
- **Ctrl-C** seçilen bir komutun iptali.
- **Ctrl-R** dosya açma.
- **Ctrl-O** dosya kaydetme(yazılacak dosyanın ismini sorar).
- **Ctrl-V** bir sonraki sayfa.
- **Ctrl-Y** bir önceki sayfa.
- **Ctrl-F** imleci(cursor) bir karakter ileriye hareket etmek.
- **Ctrl-B** imleci bir karakter geriye hareket ettirmek.
- **Ctrl-P** imleci bir satır yukarıya hareket etmek.
- **Ctrl-N** imleci bir satır aşağıya hareket ettirmek.
- **Ctrl-K** satır kes.
- **Ctrl-U** satır yapıştır.
- **Ctrl-W** karakter arama.
- **Ctrl-X** dosyadan çıkış(dosyada yaptığınız değişiklikleri kaydetmediyseniz çıkarkan kaydetmeniz için sizi uyarır. N kaydetmeden çıkar. Y kaydedip çıkar.
- **Ctrl-G** ile burada anlatılmayan seçenekleri de inceleyebilirsiniz.

# Editörler-3

## (emacs)

- Emacs'in kendisi başlı başına bir ders konusudur. Fakat biz burada kısaca değineceğiz.
- Linux'in ön-tanımlı (default) editörü olmadığı için her linux'te bulunmayabilir.
- Emacs grafik metin editörüdür.
- X-windows olarak çalıştırılabilir ve oradan açılan penceredeki seçeneklere tıklayıp işlemleri gerçekleştirebilirsiniz. Fakat kısa-yol tuşlarını öğrenmek ileride çok işinize yarayabilir.
- Kısa-yol tuşları hem daha hızlıdır hem de X-window seçekleri olmadığı durumlarda işinizi çok kolaylaştırır.
- >emacs dosya-adı1 dosya-adı2 ... şeklinde kullanılır.
- X-window imkanı olmayan durumlarda da >emacs -nw dosya-adı şeklinde kullanılabilir.
- C ve M harfleri emacs'ta Ctrl ve Alt tuşları anlamındadır.
- Ctrl-X eliniz kontrol tuşuna basılıyken X tuşuna basın anlamındadır.
- Ctrl-X Ctrl-S Yukarıda olduğu gibi önce kontrol-X sonra da kontrol-S tuşuna basın. Elinizi kontrol tuşundan kaldırmadan da önce X sonra S tuşuna basabilirsiniz.
- Ctrl-X h Eliniz kotrol tuşuna basılıyken X tuşuna basın sonra da tuşları bırakıp h tuşuna basın anlamındadır.

# Editörler-4

## (emacs)

- **Ctrl-G** seçilen seçeneğin iptali (**Ctrl-C** yerine).
- **Dosya İşlemleri**
- **Ctrl-X-Ctrl-f** dosya açma
- **Ctrl-X-Ctrl-S** dosya kaydetme
- **Ctrl-X-Ctrl W** dosyayı farklı isimle kaydetme
- **Ctrl-X-i** açık olan dosyaya ikinci bir dosyayı ilave etmek.
- **Ctrl-X-d** dizin açma
- **Desen Arama-Değiştirme**
- **Ctrl-S** ileriye doğru karakter arama
- **Ctrl-R** geriye doğru karakter arama
- **Alt-%** bul-değiştir.
- **Silme-Kesme-Kopyalama**
- **Ctrl-K** göstergenin bulunduğu noktandan satır sonuna kadar olan karakterleri keser.
- **Ctrl-\_** Geri al.
- **Ctrl-Boşluk** karakter seçme modu
- **Alt-M** Ctrl-boşluk ile seçilen yerden itibaren Alt-M yazılan yere kadar seçer.
- **Ctrl-w** benzer şekilde kesmek için kullanılır.
- **Alt-w** Ctrl-w ile benzerdir.
- **Ctrl-Y** yapıştır.

# Editörler-5 (Emacs)

- **Pencere İşlemleri**
- **Ctrl-x o** İmleci diğer pencereye geçirir (küçük o harfi)
- **Ctrl-x 2** Var olan pencereyi yatay olarak ikiye böler
- **Ctrl-x 3** Var olan pencereyi dikey olarak ikiye böler
- **Ctrl-x 1** Sadece imlecin bulunduğu arabelleği gösterir
- **Ctrl-x 0** İmlecin bulunduğu pencereyi yok eder (sıfır)
- **Yardım İşlemleri**
- **Ctrl-X k** kısayol tuşlarının ne işe yaradığını gösterir.
- **Ctrl-h Ctrl-h** yardım hakkında yardım
- **Ctrl-h Ctrl-f** sıkça sorulan sorular.
- **Kabuk Komutları**
- **Alt-M-!** Kabuk komutları yazmak için komut emacs'tan çıkmadan komut satırı açar.
- **Emacs'tan çıkış**
- **Ctrl-X-Ctrl-C** emacs'tan çıkış.

# Basit Bir Örnek Program (Hello.c)

```
#include <iostream.h>
Main()
{
    std::cout << "selam dünya" << std::endl;
    return 0;
}
```

```
g++ hello.c -o hello.exe
```

# Kaynaklar

- <http://www.linux.org.tr>
- <http://gnu-tr.sourceforge.net>
- <http://www.debian.org.tr>
- <http://www.uludag.org.tr>
- <http://www.gnome.org.tr>
- <http://www.kde.org.tr>
- <http://ileriseviye.org>
- <http://www.enderunix.org>
- <http://www.linux34.com>
- <http://people.core.gen.tr/~aycan.irican/web/publish/NasilBirEmacs.html>