

LINUX SİSTEM YÖNETİMİ





BEN KİMİM?

Mehmet Gürol ÇAY

MetaRock siber güvenlik ekibi üyesi
GTÜ Bilgisayar Mühendisliği öğrencisi

mgurolcay.wordpress.com
gurolcay123@gmail.com

Neler Göreceğiz?

1. Linux Dağıtımları
2. Komut Satırı
3. Text İşlemleri
4. Kullanıcı Yönetimi
5. İşlemler
6. İzinler
7. Dosya İşlemleri
8. Biraz Kripto

UFAK BİR GİRİŞ

Ken Thompson ve Dennis Ritchie, 1969 yılında Bell Laboratuvar'ında UNIX işletim sistemini geliştirdi. Daha sonra C programlama dilini geliştirdiler.

Daha sonralarında Richard Stallman GNU(GNU is Not UNIX) projesi üzerinde çalışmaya başladı. GNU çekirdeğinin adı Hurd.

Bu sırada UNIX benzeri işletim sistemi olan MINIX'de geliştirildi.

1991'de ise Linus Torvalds bugünkü Linux çekirdeğini geliştirmeye başladı.



Soldan sağa: Ken Thompson , Dennis Ritchie



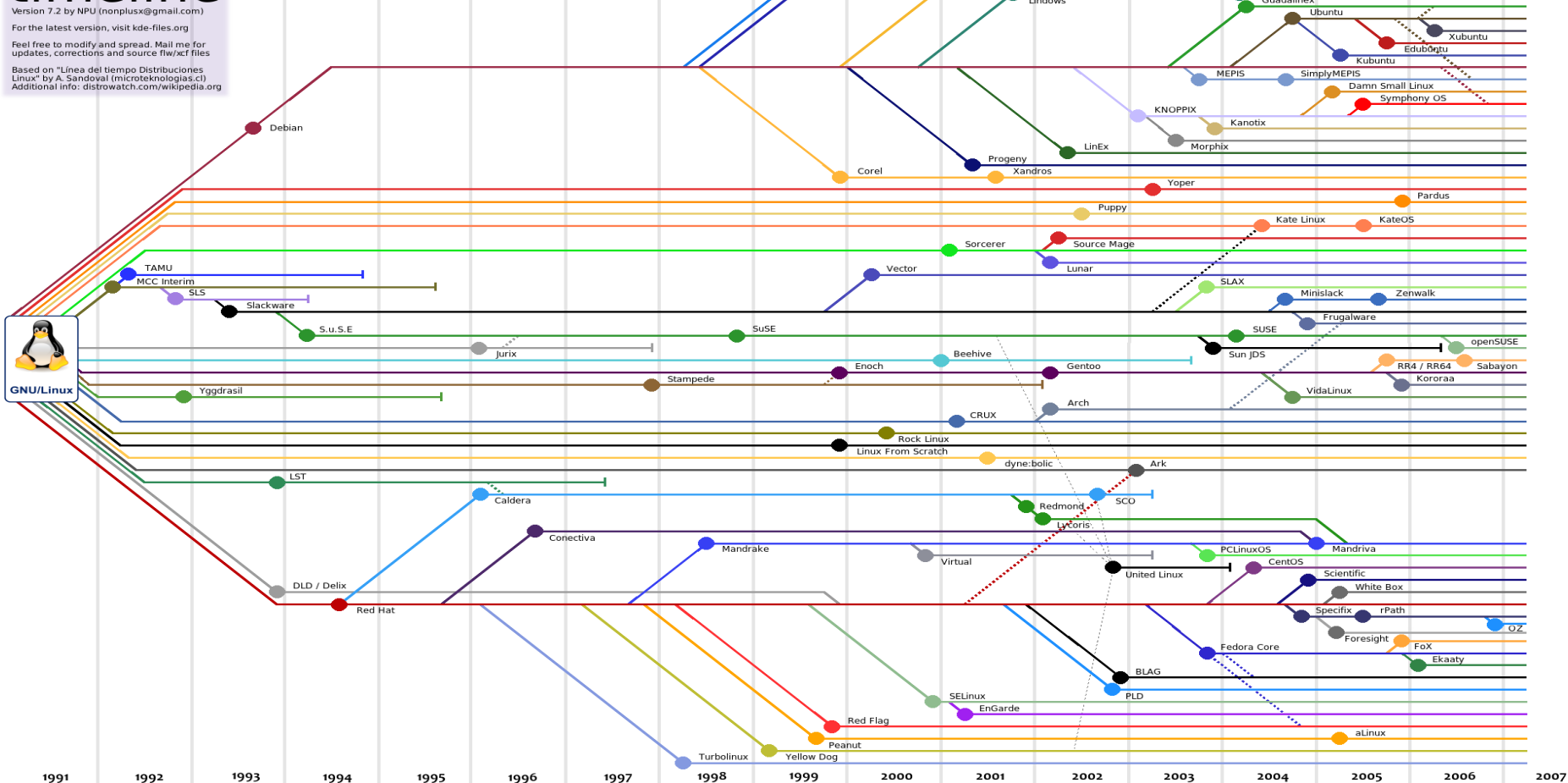
Linus Torvalds

1.

LINUX DAĞITIMLARI



Version 7.2 by NPU (nonplusx@gmail.com)
For the latest version, visit kde-files.org
Feel free to modify and spread. Mail me for updates, corrections and source flw@kde-files.org files
Based on "Línea del tiempo Distribuciones Linux" by A. Sandoval (microtecnologias.cl)
Additional info: distrowatch.com/wikipedia.org



Belli başlı GNU/Linux Dağıtımları



► **Debian:** 20 yılı aşkın süredir geliştirilen bir sistemdir. Kendi paket sistemini kullanır. Mükemmel stabil çalışan bir sistemdir.



► **Red Hat Enterprise Linux:** RHEL olarak anılır. Debian'dan farklı paket sistemini kullanır: RPM. Sunucu olarak kullanılır.



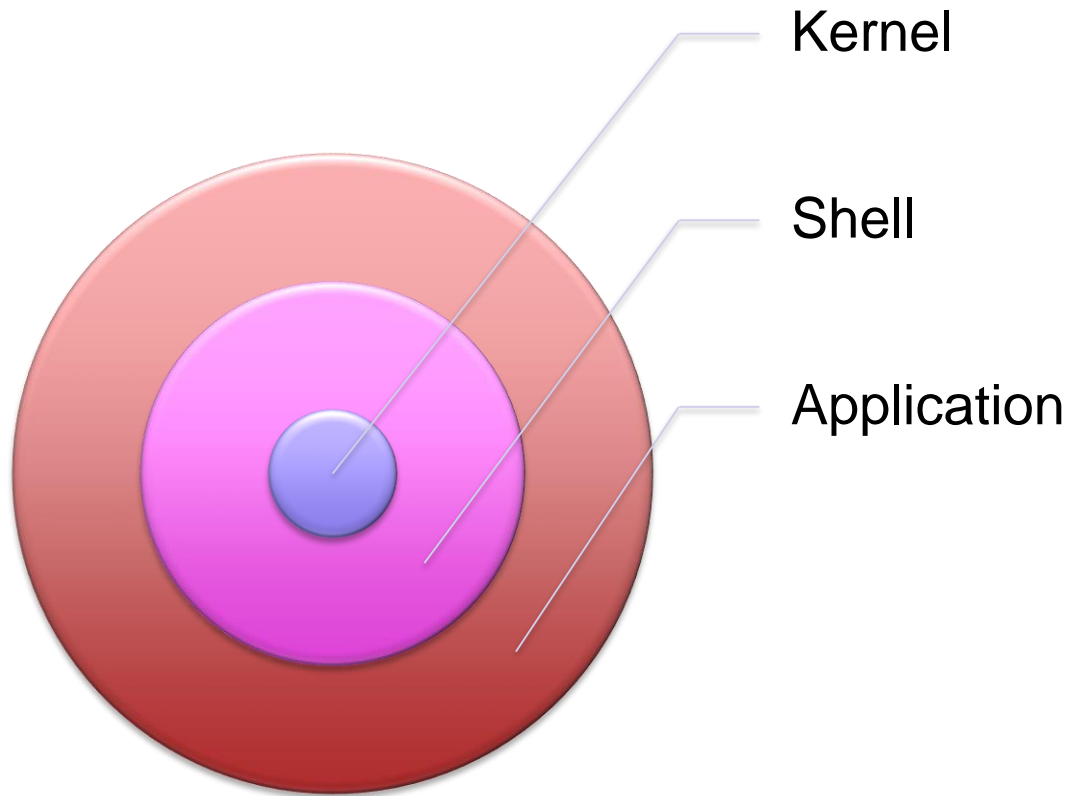
► **Ubuntu:** Debian tabanlı bir sistemdir. Debian paket yönetim sistemini kullanır. Başlangıç için iyi bir tercihtir.



► **Fedora:** Red Hat'ın bir kolu diyebiliriz. Red Hat paket yönetim sistemini kullanır.



► **Arch Linux:** Tamamen topluluklar tarafından geliştirilir. Debian'a benzer ancak güncellemeleri hemen alır. Yani test ve stabil olmayan sürümleri yoktur. Kendi paket yöneticisini kullanır.



2.

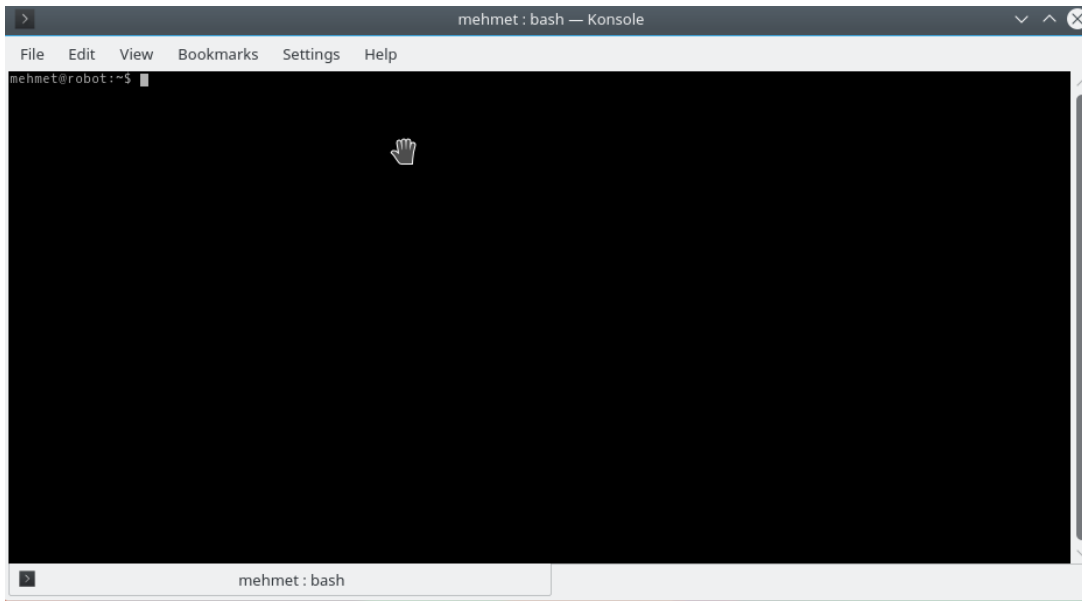
KOMUT SATIRI





Shell Nedir?

- ▶ Shell (Kabuk) kısaca özetleyecek olursak klavyeden komutları alıp işletim sistemine gönderen bir programdır.
- ▶ “Terminal” veya “Console” adındaki programlar ile Shell’ı çalıştırmak mümkün.



► username@hostname:current_dir \$

► Her shell kendi simgesi vardır. Biz bash kullandığımız için simgesi “\$” işaretidir.

Diğer Shell Türleri

► Bash(eskiden sh) Kabuğu:

► Bash, ilk Unix sistemlerde sh,ksh ve bazı csh özellikleri için geliştirilmiştir. Çoğu Linux sistemlerinde varsayılandır.

► Tcsh (eskiden csh) Kabuğu:

► C kabuğunun(csh) açık kaynak versiyonudur. Çoğu Berkeley Unix sistemlerinde varsayılandır.

► Ksh Kabuğu:

► Sh kabuğunun atasıdır.

► Zsh Kabuğu:

► Sh kabuğunun klonlarından birisidir. POSIX uyumludur. Yazım denetimi ve komut düzenlemeye farklı yaklaşımları gibi özellikleri vardır. İlk Mac OS X sistemlerinde varsayılandı. Şu an Bash.



Bir Kaç **Shell** Komutu Öğrenelim

- ▶ Linux'da herşey bir dosyadır. Bu dosyalar bir hiyerarşiye sahiptir.
- ▶ İlk dizin “root” olarak isimlendirdiğimiz dizindir ve “/” simgesi ile gösterilir.



▶ \$ whoami



▶ \$ echo “hello world”



▶ \$ date

- ▶ `$ pwd`

- ▶ Bu komut çalıştığınız dizinin tam adresini verir

- ▶ `$ cd`

- ▶ Bulunduğunuz dizin yolunu değiştirmenizi sağlar

- ▶ `$ touch`

- ▶ Dosya oluşturmanızı sağlar. Bu komut ile dosyanın oluşturulma tarihi ve değiştirilme tarihi değiştirilebilir.

- ▶ \$ touch

- ▶ Dosya oluşturmanızı sağlar. Bu komut ile dosyanın oluşturulma tarihi ve değiştirilme tarihi değiştirilebilir.

- ▶ \$ file

- ▶ Linux'ta dosyaların tipini öğrenmek için kullanılır.

► \$ ls -la

- Bulunduğunuz dizin altındaki dizinleri dosyaları listeler
- “-la” komutu, “-l” listeleme, “-a” gizli dosyaları gösterme

```
mehmet@ubuntu: ~  
mehmet@ubuntu:~$ ls -l  
total 44  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:24 Desktop  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:20 Documents  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Downloads  
-rw-r--r-- 1 mehmet mehmet 8980 Oct  3 14:13 examples.desktop  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Music  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Pictures  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Public  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Templates  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:17 Videos  
mehmet@ubuntu:~$
```



```
mehmet@ubuntu: ~  
mehmet@ubuntu:~$ ls -l  
total 44  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:24 Desktop  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:20 Documents  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Downloads  
-rw-r--r-- 1 mehmet mehmet 8980 Oct 3 14:13 examples.desktop  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Music  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Pictures  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Public  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Templates  
drwxr-xr-x 2 mehmet mehmet 4096 Oct 3 14:17 Videos
```

Dosya/dizin izinleri	Link sayısı	Dosya sahibi	Dosya grubu	Dosya boyutu	Dosya değiştirilme tarihi	Dosya isimleri
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:24	Desktop
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:20	Documents
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Downloads
-rw-r--r--	1	mehmet	mehmet	8980	Oct 3 14:13	examples.desktop
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Music
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Pictures
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Public
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Templates
drwxr-xr-x	2	mehmet	mehmet	4096	Oct 3 14:17	Videos

- İlk sütun dosya/dizin durumu:
 - ▷ “d” olursa bir dizin
 - ▷ “c” olursa character device
 - ▷ “b” block device
 - ▷ “l” bir dosyayı işaret eden link

- Dosya izinlerinde ilk üçlü grup root yetkilerini, sonraki üçlü grup kullanıcı yetkilerini, son üçlü grupta diğer kullanıcıların yetkilerini gösterir



- ▶ Linux işletim sistemlerinde dosya tipinin ne olduğunu söyleyen bir uzantıya ihtiyaç yoktur.

```
$ cat myfile newfile
```

- ▶ Dosya içeriğini gösterir. İki farklı dosya ismi *cat* komutuna verilirse dosyaların birleşimi gösterilir.

```
$ cat myfile > newfile
```


- ▶ *myfile* içeriği *newfile* dosyasına yapıştırılır.

```
$ cat myfile >> newfile
```

- ▶ *myfile* içeriği *newfile* dosyasına eklenir.




- ▶ Terminalde bir önceki komuta erişmek için yukarı aşağı yön tuşları kullanılabilir.



► \$ less /home/blabla/text1

► Bu komut ile çok uzun dosyaların içeriğini terminal ekranına sığacak kadar gösterir.

► **q:** çıkış **g:** dosya başı **G:** dosya sonu **/search:** arama **h:** yardım



► \$ history

► Terminalde girdiğiniz komut geçmişini görüntüler.

► \$ clear

► Terminal sayfasını yeniler.

Dosya İşlemleri

- ▶ `$ cp mycoolfiler /home/username/Documents/`

- ▶ Komutu ile “mycoolfiler” dosyasını “home/username/Documents/” alanı kopyalamış oluruz.

- ▶ “-r” komutu verirsek kopyalama bir dizin için yapılıyor ise özyinemeli olarak uygulanır

- ▶ “-i” komutu ile aynı dosya isminin üstüne yazılması sağlanır

- ▶ `$ mv mycooolfile /home/username/Documents/`

- ▶ Komutu ile “mycooolfile” dosyasını “home/username/Documents/” alanı taşımış oluruz.
- ▶ “-r” komutu verirse kopyalama bir dizin için yapılıyor ise özyinemeli olarak uygulanır
- ▶ “-i” komutu ile aynı dosya isminin üstüne yazılması sağlanır

- ▶ \$ mkdir books

- ▶ Dosyalarımızı koyabileceğimiz bir dizin oluşturmak için kullanılır.

- ▶ “-p” komutu ile alt dizinleride oluşturmak mümkündür.

- ▶ \$ mkdir books/cemalsureya/

▶ \$ rm file1

- ▶ komutu ile *file1* dosyasını silebiliriz.
- ▷ “-f” komutu ile silmeye zorlayabilir
- ▷ “-r” komutu ile bir dizini özyinemeli olarak silebiliriz.

▶ \$ rmdir books

- ▶ `$ find /home -name blabla.jpg`

- ▶ bir dosyayı aramak istediğinizde kullanabileceğiniz bir komut. Bulunan dosyanın arattığınız yolla(path) birlikte verir.

- ▶ “-type d” komutu ile bir dizin arayabilirsiniz.

- ▶ `$ man ls`

- ▶ bir komutun ne olduğunu nasıl çalıştığını anlatan komuttur. Örneğimizde `ls` komutumuzun nasıl çalıştığını verecektir.

- ▶ `$ whatis ls`

- ▶ sonrasında verilen komutun kısaca tanımlamasını verir

- ▶ `$ alias foobar='ls -la'`

- ▶ Bazen çok uzun komutları tekrar tekrar kullanmak zorunda kalırız. Bunu çözebilmek için *alias* komutu kullanılır. Yukarıdaki örneğimizde “*ls -la*” komutu yerine *foobar* komutu çağırılması yeterlidir.

- ▶ `$ ~/.bashrc`

- ▶ Peki bilgisayarımızı yeniden başlatmak zorunda kaldık. Bu komutu verdiğimizde yeniden başlatmadan sonrada “*foobar*” komutu geçerli olacaktır.

- ▶ `$ unalias foobar`

- ▶ Yaptığımız atamayı kaldırmak için

- ▶ `$ exit`

3.

TEXT İŞLEMLERİ



Neler Göreceğiz?

► Bu bölümde komut satırından basit bir şekilde nasıl dosyaların içeriğini değiştiririz, nasıl görüntüleriz, bir komutun çıktısını başka bir çıktıya nasıl veririz gibi sorulara yanıt bulacağız.

Stout-Stdin-Stderr

- ▶ **Stdin:** standard input dosyasıdır ve varsayılan olarak okuma amacıyla klavyeye yönlendirilmiştir.
- ▶ **Stdout:** standard output dosyasıdır ve varsayılan olarak yazma amaçlı ekrana yönlendirilmiştir.
- ▶ **Stderr:** programlarda oluşacak hata mesajlarının ekrana yazılması için kullanılan standard error dosyasıdır. Stdout'tan farkı buffer kullanılmadan hatanın doğrudan ekrana gönderilmesidir.

stdout

- ▶ `$ echo Hello World > peanuts.txt`

- ▶ Dersimizin ilk kısmında bu komutu kullanmıştık. Hatırladığınız gibi ekrana herhangi bir şey yazdırmak istediğimizde “echo” komutu bize yardımcı oluyordu.

- ▶ Burada “echo” komutu klavyeden bir *input* alıyor ve bunu *I/O stream*’e vererek çıktıyı döndürüyor.

stdout

► >

► Bu bir *redirection* operatörüdür ve standart *output*'a giden verinin yönünü değiştirir. Böylece *echo* komutunun almış olduğu *input*, *I/O stream*'a yönlendirilmek yerine dosyaya yazılır. Bu *redirect* komutu dosyada bir şey varsa üstüne yazılmasına neden olur.

stdout

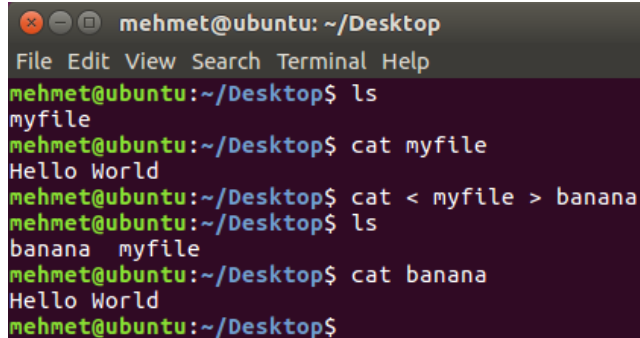
- ▶ `$ echo Hello World >> peanuts.txt`

- ▶ *Redirection* komutunu iki kere verirsek örnekte görüldüğü gibi, dosyanın sonuna ekleme yaparak dosya içeriğinin güncellenmesini sağlar.

stdin

- ▶ `$ cat < myfile.txt > banana.txt`

- ▶ `cat` komutu bize dosya içeriğini veriyordu. Burada `peanuts.txt` dosyasının içeriğini `banana.txt` dosyasına yönlendirmektedir.

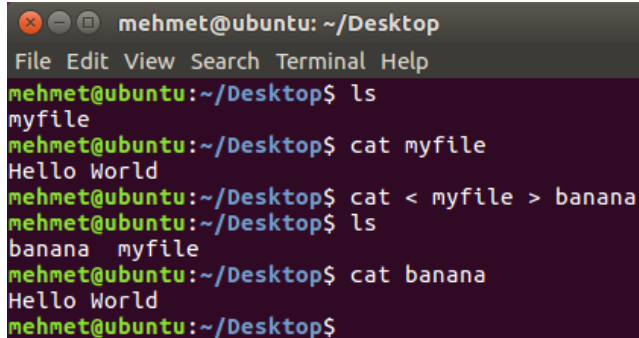


```
mehmet@ubuntu: ~/Desktop
File Edit View Search Terminal Help
mehmet@ubuntu:~/Desktop$ ls
myfile
mehmet@ubuntu:~/Desktop$ cat myfile
Hello World
mehmet@ubuntu:~/Desktop$ cat < myfile > banana
mehmet@ubuntu:~/Desktop$ ls
banana  myfile
mehmet@ubuntu:~/Desktop$ cat banana
Hello World
mehmet@ubuntu:~/Desktop$
```

stdin

- ▶ `$ cat < myfile.txt > banana.txt`

- ▶ `cat` komutu bize dosya içeriğini veriyordu. Burada `peanuts.txt` dosyasının içeriğini `banana.txt` dosyasına *stdin redirection* komutu ile yönlendirmektedir.

A terminal window titled 'mehmet@ubuntu: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
mehmet@ubuntu:~/Desktop$ ls
myfile
mehmet@ubuntu:~/Desktop$ cat myfile
Hello World
mehmet@ubuntu:~/Desktop$ cat < myfile > banana
mehmet@ubuntu:~/Desktop$ ls
banana  myfile
mehmet@ubuntu:~/Desktop$ cat banana
Hello World
mehmet@ubuntu:~/Desktop$
```

stderr

- ▶ `$ ls /deneme/dizin > peanuts`

- ▶ Yukarıdaki komutu girdiğimizde `/deneme/dizin` yolu olmadığı için `ls` komutu bir hata döndürecektir. Bu kısımda hatanın `peanuts` dosyasına yazılmasını bekleyebilirsiniz. Ancak böyle bir şey söz konusu değildir.

- ▶ Linux üzerinde *file descriptors* vardır. Bunlar `stdin`, `stdout` ve `stderr`'yi tanımlamak için kullanılmaktadır ve sırasıyla 0,1 ve 2 değerlerini alırlar.

stderr

- ▶ `$ ls /deneme/dizin 2> peanuts`

- ▶ 0 *stdin*, 1 *stdout*, 2 *stderr*

- ▶ Eğer komutumuz yukarıdaki gibi olsaydı *ls* komutunun döndürdüğü hata dosyaya kaydedilecekti.

- ▶ `$ ls /deneme/dizin > peanuts 2>&1`

- ▶ Komutu ile *stderr*'nin *stdout*'a işaret etmesini sağlamış olduk. Böylece hatamız hem dosyada hem de ekranda görülebilecek

pipe and tee

- ▶ `$ ls -la /etc | less`

▶ *pipe* bir komutun çıktısının başka bir komuta verilmesini sağlar. Yukarıda vermiş olduğumuz örnekte `ls -la /etc` klasörünün çıktısı çok uzun olacaktır. Bu komutun çıktısını `less` komutuna “|” ile vererek okumayı kolaylaştırmıştır.

- ▶ `$ ls | tee banana`

▶ iki farklı *stream* de çıktımı görmek istediğimde bu komutu kullanabilirim.

env (Environment)

- ▶ `$ echo $HOME`

▶ Yukarıdaki komut size ev klasörünün yolunu verecektir. Bu bilgiler Linux içerisinde zaten oluşturulmuş kaynaktan gelmektedir.

- ▶ `$ env`

▶ Bu komut ile tüm *environment*'ları görmek mümkündür.

env (Environment)

```
mehmet@ubuntu: ~/Desktop
File Edit View Search Terminal Help
mehmet@ubuntu:~/Desktop$ env
XDG_VTNR=7
XDG_SESSION_ID=c2
CLUTTER_IM_MODULE=xim
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/mehmet
SESSION=ubuntu
GPG_AGENT_INFO=/home/mehmet/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
SHELL=/bin/bash
XDG_MENU_PREFIX=gnome-
VTE_VERSION=4205
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
WINDOWID=56623114
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/5989
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=mehmet
```

- Burada bulunan *environment*'leri başına “\$” koyarak ekrana basabiliriz.

cut

- ▶ `$ cut -c 5 sample.txt`

- ▶ *sample.txt* içerisindeki yazının beşinci karakterini kesmektedir. Ancak orjinal yazı bozulmaz sadece istenilen yer bize gösterilir.

- ▶ `$ cut -f 1 -d “;” sample.txt`

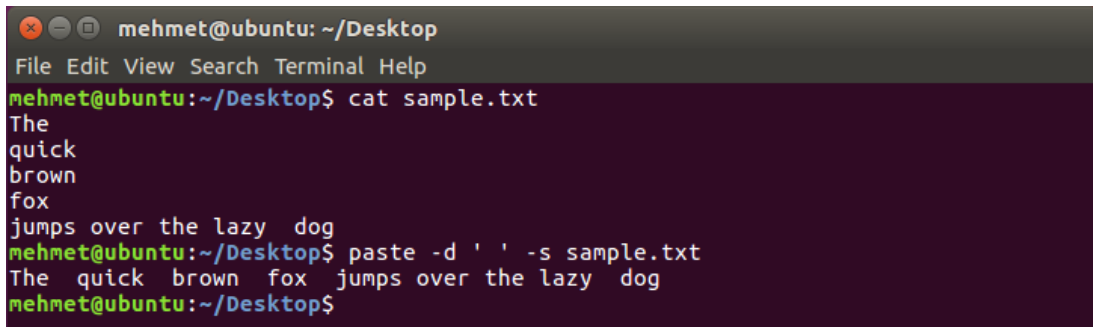
- ▶ “-f” ekiyle hangi field’ta çalışacağını belirler.
- ▶ “-t” ekiyle de ayırıcının ne olduğunu belirtebilirsiniz böylece ayırıcımıza kadarki cümle veya kelimeleri terminal ekranında gösterir.

paste

- ▶ `$ paste -s sample.txt`

▶ Bu komut, cat ile benzerlik göstermektedir. Örneğin; satır satır olan dosya içeriğinin tek bir satırda gösterilmesini sağlar. Ayrıca bu boşlukların ne şekilde gösterilebileceğini düzenleyebiliriz.

- ▶ `$ paste -d ' ' -s sample.txt`

A terminal window titled 'mehmet@ubuntu: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'mehmet@ubuntu:~/Desktop\$'. The first command is 'cat sample.txt', which outputs 'The quick brown fox jumps over the lazy dog' on five lines. The second command is 'paste -d ' ' -s sample.txt', which outputs 'The quick brown fox jumps over the lazy dog' on a single line with spaces between words. The prompt returns to 'mehmet@ubuntu:~/Desktop\$' after each command.

```
mehmet@ubuntu: ~/Desktop
File Edit View Search Terminal Help
mehmet@ubuntu:~/Desktop$ cat sample.txt
The
quick
brown
fox
jumps over the lazy dog
mehmet@ubuntu:~/Desktop$ paste -d ' ' -s sample.txt
The quick brown fox jumps over the lazy dog
mehmet@ubuntu:~/Desktop$
```

tail

- ▶ `$ tail /var/log/syslog`

- ▶ *head* komutu ile benzer çalışmaktadır. Farkı ise tail son 10 satırı göstermektedir.

- ▶ `$ tail -n 15 /var/log/syslog`

- ▶ Son 15 satırını göstermesini sağlar.

- ▶ `$ tail -f /var/log/syslog`

- ▶ Syslog dosyası devamlı değişen bir dosyadır. Bu dosyayı devamlı izlemek istiyorsak “-f” komutu kullanabiliriz.

Tr (Translate)

- ▶ \$ tr a-z A-Z

- ▶ Bu komut bize küçük harflerin büyük harflere çevrilmesini sağlar.

uniq (Unique)

- ▶ `$ uniq sample.txt`

▶ *sample.txt* dosyası içerisinde birden fazla aynı geçen kelimeleri birer defa gösterecek şekilde düzenler.

- ▶ `$ uniq -c sample.txt`

▶ *sample.txt* dosyasında bir kelimedenden kaç adet olduğunu yanına yazarak gösterir.

- ▶ `$ uniq -u sample.txt`

uniq (Unique)

- ▶ `$ uniq -u sample.txt`

- ▶ *sample.txt* dosyasında bir defa kullanılmış kelimeyi bulur

- ▶ `$ uniq -d sample.txt`

- ▶ *sample.txt* dosyasında iki kere geçen kelimeleri bulur

uniq (Unique)

```
mehmet@ubuntu: ~/Desktop
File Edit View Search Terminal Help
mehmet@ubuntu:~/Desktop$ cat sample.txt
brown
The

brown
fox
jumps
jumps
fox
brown
mehmet@ubuntu:~/Desktop$ uniq sample.txt
brown
The

brown
fox
jumps
fox
brown
mehmet@ubuntu:~/Desktop$ uniq -c sample.txt
 1 brown
 1 The
 1
 1 brown
 1 fox
 2 jumps
 1 fox
 1 brown
```

grep

- ▶ `$ grep fox sample.txt`

- ▶ Text işlemede, arama yapmada en çok kullanılan komuttur. Genellikle başka bir komutun çıktısı *pipe* ile bu komuta aktarılarak sonuç içerisinde arama yapılır.

- ▶ `$ env | grep -i User`

- ▶ `$ ls /somedir | grep '.txt$'`

Meta
karakter

Kapsam

Örnek

Sonuç

^	Satırı başını belirtir.	^silgi	Satır başlarında “silgi” geçen bütün satırları getirir.
\$	Satır sonlarını belirtir.	Kalem\$	Satır sonları “kalem” ile biten satırları getirir.
.	Herhangi bir karakter ile eşleşir.	If...e	“if” ile başlayan sonrasında herhangi iki karakter gelen ve sonu “e” ile biten ifadeleri getirir.
*	Kendinden önceki karakterden o veya daha fazlasını belirtir	Kalem*	“kalem” ile başlayan herşey ile eşleşir.
[]	Köşeli parantez içindeli karakterlerden sadece biriyle eşleşir	[a-z]*	Küçük alfabetik karakterlerle eşleşir veya hiçbir şeyle eşleşmez
		[Kk]alem	Kalem veya kalem içeren satırları getirir.

Text Editörler

- ▶ Vim (Vi Improved)
- ▶ Emacs
- ▶ Nano

3.

KULLANICI
YÖNETİMİ



Kullanıcılar ve Gruplar

- Her işletim sisteminde kullanıcı ve gruplar vardır. Her kullanıcının yalnızca kendi erişimi ve yetkileri vardır. Bir işlem çalışırken ona sahip olan kullanıcının yetkileriyle çalışır. Dosya erişimi, sahipliği ve yetkiler birbirleriyle bağlantılıdır.
- Her kullanıcının kendi ev dizini ve kendi özel dosyalarını tuttuğu bir alan vardır. Bu dosyalar genellikle */home/username/* altında tutulur.

- İşletim sistemi kullanıcıları yönetilmek için onlara farklı **UID** (user ID)'ler verir. İşletim sistemi grupları da yönetebilmek için de **GID** (groups ID)'ler tanımlar.
- Linux'da iki tane çok önemli kullanıcı vardır: **root** ve **superuser**.
- **root**: Bu kullanıcı sistemde her türlü yetkiye sahiptir ve her türlü dosyaya erişebilir. Her türlü işlemi(process) başlatıp bitirebilir.
- **superuser**: **root** olarak işlem yapmak çok tehlikeli olabileceği için **sudo** komutu ile **superuser** erişimi elde edilebilir.

- ▶ **superuser** ile **root** yetkisi kazanırız.
- ▶ Aşağıdaki komutu deneyelim:

```
▶ $ cat /etc/shadow
```

```
▶ $ su
```

▶ **sudo** komutu gibidir. Eğer herhangi bir kullanıcı adı belirlenmemişse bu komut bize root yetkileri olan bir shell verecektir.

- ▶ Bize **root** yetkileri kazandırdığı için bu komutu kullanmak tehlikelidir. Bu yüzden **sudo** komutu kullanmak daha yerinde olacaktır.
- ▶ Peki sistem bizim bir komutu **superuser** olarak çalıştırdığımızı nerden anlıyor?

- ▶ **/etc/sudoers**

- ▶ Bu dosyayı **visudo** komutu ile değiştirmek mümkün.

/etc/passwd

- Bu dosya içerisinde hangi kullanıcı hangi **UID** ile eşleştirilmiş görülebilir.

```
mehmet@ubuntu:~/Desktop$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

- Dosya içeriği yukarıdaki görseldeki gibidir. Her bir satır bir kullanıcıyı temsil etmektedir.
- Sırasıyla inceleyecek olursak:
 - ▷ Kullanıcı adı
 - ▷ Parola (parola gerçekten bu dosya içerisinde tutulmamaktadır.)


```
mehmet@ubuntu:~/Desktop$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

- ▶ Kullanıcı ID(User ID)
- ▶ Grup ID
- ▶ Kullanıcı hakkında yorum: genellikle gerçek ismi veya telefon numarası gibi.
- ▶ Kullanıcı ev klasör yolu
- ▶ Kullanıcının kullandığı shell: genellikle bash'dir.

/etc/passwd

- ▶ Kullanıcı yetkilendirmeleri hakkında bilgiler saklar.
- ▶ Erişim için **superuser** olunması gereklidir.

```
mehmet@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep mehmet  
mehmet:$1$WaDFz7$.gHP7HSkcr9nZNVaxDwte1:17077:0:99999:7:::
```

- ▶ Kullanıcı adı
- ▶ Şifrelenmiş parola
- ▶ Parolanın değiştirildiği tarih: 1 Ocak 1970'ten itibaren gün olarak ifade edilir.
- ▶ Parolanın en az değiştirilme süresi
- ▶ Parolanın süresi.
- ▶ Parola süresi dolduktan sonra geçerli olduğu süre
- ▶ Gelecek kullanıcı yenilikleri ayrılmış alan

/etc/shadow

```
[root@indishell ~]#  
[root@indishell ~]# head -n 1 /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

it shows user shell name , when user will get successfull login, he will get " bash " shell

this is the user home directory , where user will be dropped by OS after login

this is gcosse field which contain additional info for user

this field shows the group id number of the user tho which group user belongs

This field represent user id number of user.

this x represnet that user password will be checked from shadow file . if we delete x from here, user will be allowed to login without password

this is the username of the user

/etc/shadow

1. **Kullanıcı adı:** kullanıcıların giriş yaparken kullandıkları isim. dizin yolundaki isimle aynı olma ihtimali yüksektir.
2. **Şifre:** passwd de şifreler gözükmmez. Onun yerine x yazar. Asıl şifreler /etc/shadow da kriptolanmış olarak bulunur. Eğer bu bölüm boş bırakılmışsa şifresiz olarak giriş yapılabilir. Eğer * varsa kullanıcı disable olmuştur.
3. **User id:** kullanıcı id . 0 ' ı root kullanır. Genel olarak 100 e kadar olanlar sistemin kendi sanal kullanıcılarına aittir. Mesela daemon,mail,wealizer vs vs... Bunlar normal kullanıcılar değildir. Normal kullanıcılar genellikle 100 den büyük numaraları alırlar. Bu numaralandırma 65000 e kadar gider
4. **Grup id:** /etc/group buradan gruplara ulaşabiliyoruz. Grup id hangi kullanıcının hangi gruba ait olduğunu bize bildirir. Genelde user id ile aynı olur
5. **User id info:** kullanıcı hakkında daha detaylı bilgileri bulabileceğimiz bölüm. Mesela kullanıcının telefon numarası ,tam ismi vs yazabilmektedir.
6. **Dizin yolu:** burası kullanıcının giriş yaptığı zaman bulunacağı bölüm. Burada kullanıcıya ait programlar , dosyalar vs bulunur.
7. **Shell:** Shell kullanıcının girdiği komutları işlemeyi ve sonuçlarını göstermeyi sağlayan programdır.Burada genelde /bash ya da /bin/bash yazar. Eğer /bin/false gibi bişey yazıyorsa kullanıcı girişi engellenmiş demektir

/etc/shadow

- ▶ Bugünkü dağıtımlar sadece bu **shadow** dosyasına bakarak kullanıcı yetkilendirme yapmamaktadır.
- ▶ PAM (Pluggable Authentication Modules) gibi mekanizmalar kullanılabilmektedir.

Kullanıcı Yönetim Araçları

► Kullanıcı eklemek için:

```
► $ sudo useradd bob
```

► Kullanıcı kaldırmak için:

```
► $ sudo userdel bob
```

► Kullanıcı şifresi değiştirmek için:

```
► $ passwd bob
```

4.

İZİNLER



Dosya İzinleri

► Daha önce `ls` komutunu gördüğümüzde kısaca bahsetmiştik.

```
mehmet@ubuntu: ~  
mehmet@ubuntu:~$ ls -l  
total 44  
drwxr-xr-x 2 mehmet mehmet 4096 Oct  3 14:24 Desktop
```

► Üç parttan oluştuğunu hatırlayalım. **d** harfi bize dosya tipinin ne olduğunu söylüyordu. Burda sıklıkla **-** ve **d** harflerini göreceksiniz. **-** işareti dosyanın genel dosya tipi olduğunu ifade eder.

► d | rwx | r-x | r-x

► İlk grubun root izinleri, ikinci grubun kullanıcı izinlerini ve son grubun da diğer izinlerini ifade ettiğini hatırlayalım.

► Her bir karakterin bir anlamı vardır:

► r : okunabilir (readable)

► w : yazılabilir (writable)

► x : çalıştırılabilir (executable)

► - : o iznin olmadığını simgeler.

İzinleri Değiştirme

► Dosyalarımızın izinlerini root, kullanıcı veya diğerleri için değiştirebilmemiz mümkün. Bunun için:

```
► $ chmod u+x myfile
```

► Kullanıcı için *myfile* dosyasına çalıştırabilir yetkisi vermiş olduk. “+” yerine “-” kullanmış olsaydık yetkiyi kaldırmış olurduk.

► rwx | r-x | r-x

► 111 | 101 | 101

► Dosyamıza sayısal olarak da yetki verebiliriz. Yukarıda vermiş olduğum örnekte **1** yetkinin verildiğini **0** ise yetkinin verilmediğini ifade etmektedir.

► \$ `chmod 755 myfile`

► Kullanıcı için: $1 * 2^0 + 1 * 2^1 + 1 * 2^2 = 7$ okuma yazma ve çalıştırma,

► Grup için: $1 * 2^0 + 0 * 2^1 + 1 * 2^2 = 5$ okuma ve çalıştırma,

► Diğer için: $1 * 2^0 + 0 * 2^1 + 1 * 2^2 = 5$ okuma ve yazma yetkilerini vermiş oluruz.

Dosya Sahibi Değiştirme

► Sistemimizde bir çok kullanıcı olabilir. Bu yüzden dosya sahiplerini de değiştirebilmemiz gereklidir.

► `$ sudo chown yusuf myfile`

► Komutu *myfile* dosyasının sahibini *yusuf* olarak değiştirecektir.

► `$ sudo chgrp calisma myfile`

► Komutu *myfile* dosyasının grubunu *calisma* olarak değiştirecektir.

- ▶ `$ sudo chown patty:whales myfile`

- ▶ Dosyaya sahip olan kullanıcıyı değiştirirken aynı zamanda grubunuda değiştirmek istiyorsak kullanıcı adından sonar iki nokta koyup grup ismini yazmamız yeterli.

setuid

► \$passwd

► Bu komut ile şifremizi değiştirebiliriz.

```
mehmet@ubuntu:~/Desktop$ ls -l /etc/passwd  
-rw-r--r-- 1 root root 2255 Oct  4 00:13 /etc/passwd
```

► Burada passwd komutunun yetkileri var. Komuta sahip olan root grubu da root. Okuma ve yazma yetkisi de sadece root tanımlanmış. Peki nasıl çalışıyor bu komut?

```
mehmet@ubuntu:~/Desktop$ ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 54256 Mar 29 2016 /usr/bin/passwd
```

- Yeni bir yetki daha var! **s** bu yetkilendirme SUID (set user ID) yetkilendirmesidir. Bir dosyaya böyle bir yetki verildiği zaman, o programı/komutu kim çalıştırırsa dosyanın sahibinin yetkileri ile çalıştırır.
- Peki bu yetkiyi dosyamıza nasıl vereceğiz?

► \$ sudo chmod u+s myfile

► Veya sayısal olarak

► \$ sudo chmod 4755 myfile

Setgid (Set Group ID)

► **setuid**'de olduğu gibi SGID yetkisi de vermek mümkün. Bunun için:

► \$ sudo chmod g+s myfile

► \$ sudo chmod 2555 myfile

```
mehmet@ubuntu:~/Desktop$ ls -l /usr/bin/wall  
-rwxr-sr-x 1 root tty 27368 May 27 02:31 /usr/bin/wall
```


Sticky Bit

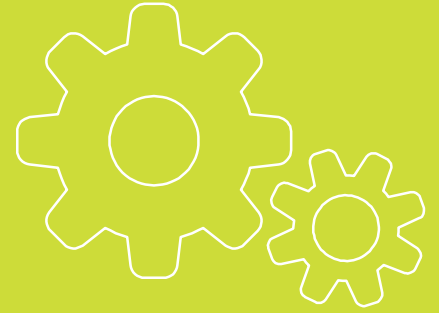
► Aynı klasör üzerinde erişme, yazma hakkı olan kullanıcıların, klasörü ve alt dosyalarını silmelerini engellemek için kullanılır. Bir dosyayı ortak kullanıma açtığımızda diğer kullanıcıların bu dosyayı herkes kullanabilir ancak sadece klasörün sahibi silebilir.

```
► $ sudo chmod +t mydir
```

```
► $ sudo chmod 1755 mydir
```

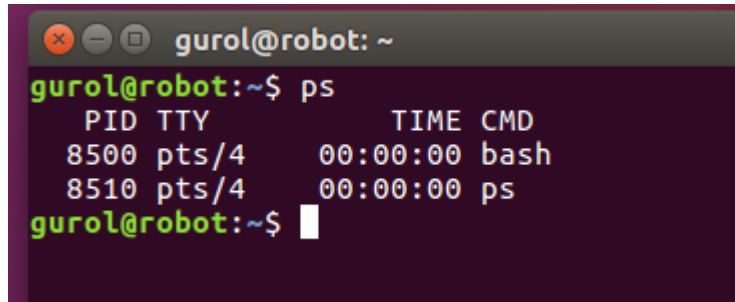
5.

İŞLEMLER



ps (processes)

- Bilgisayarınızda çalışan programlar.
- Kernek tarafından yönetilir ve her işlem bir process ID(PID) olarak ifade edilen ID'lere sahiptir.



```
gurol@robot: ~  
gurol@robot:~$ ps  
  PID TTY          TIME CMD  
 8500 pts/4        00:00:00 bash  
 8510 pts/4        00:00:00 ps  
gurol@robot:~$
```

The image shows a terminal window with a dark background. The title bar at the top reads 'gurol@robot: ~'. The prompt 'gurol@robot:~\$' is followed by the command 'ps'. The output is a table with four columns: PID, TTY, TIME, and CMD. It lists two processes: PID 8500 (bash) and PID 8510 (ps). The prompt 'gurol@robot:~\$' is followed by a cursor.

ps (processes)

- ▶ PID: Process ID
- ▶ TTY: İşlemle ilgili terminal kontrolücüsü
- ▶ Time: İşlemciyi toplam kullandığı süre
- ▶ CMD: Çalıştırılabilir komutun ismi



▶ \$ ps aux



▶ \$ top



▶ CTRL+ALT+F1 ve CTRL+ALT+F7



TTY1



PTS

İşlem Detayları

- ▶ Daha öncesinde bir process'in bir programa benzediğinden bahsetmiştik. Tam olarak, sistem programın çalışması için hafıza, işlemci ve I/O kaynak ayrımı yapar. Process aslında bir programın parçasıdır.
- ▶ Kernel process'lerden sorumludur. Biz programı çalıştırdığımız zaman kernel kodu hafızaya yükler ve onun kaynakları nasıl kullanacağına karar verir.

İşlemleri Sonlandırma

- Bir program çıkmak için `_exit` sistem komutunu çağırabilir. Bir işlem biterken bir durum ile biter ve bu durum 0(sıfır) ise işlemin başarılı olarak bittiği anlamına gelir.
- **Orphan Processes:** Ana process, çocuk(child) processten önce öldüğünde, kernel child process'in bir çağrı beklediği için gitmediğini bilir. Bu durum process'ı orphan durumuna sokar ve bunları init altına koyar. Init sonunda bu orphans işlemler için sistem çağrısı yapacaktır.

- ▶ **Zombie Processes:** Zombie işlemler öldürülemez, teknik olarak ölüdürler. Bu yüzden herhangi bir sistem sinyali ile onları öldürmek mümkün değildir.
- ▷ Ana işlem bir sistem çağrısı yaptığı zaman zombie işlem kaybolur. Bu “reaping” olarak adlandırılır.
- ▷ Zombi işlemler, işlem tablosu(process table) uzayında yer turacaktır ve diğer işlemlerin çalışmasını engelleyecektir.

Sinyaller

► Programları veya işlemleri sonlandırabilmek için sisteme bazı komutlar verilebilir. CTRL+Z ve CTRL+C işlemi öldürmek, bekletmek veya uykuya almak için kullanılabilir.

SIGHUP veya 1	Kapatmak, sonlandırmak (hangup)
SIGINT veya 2	Kesmek, ara vermek (interrupt)
SIGKILL veya 9	Öldürmek
SIGSEGV veya 11	Segmentation Fault (segment hatası)
SIGTERM veya 15	Yazılım sonlandırma
SIGSTOP veya STOP	Durdurma

kill

- ▶ \$ kill 12445

- ▶ Kill komutu işlemi sonlandırmak için kullanılan komuttur. Kill komutundan sonra istediğimiz sinyal kodunu verebiliriz.

- ▶ \$ kill -9 12445

/proc

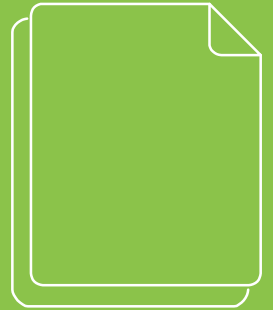
► Linux'ta herşeyin bir dosya olduğunu söylemiştik. Process'ler bile bir özel dosyadır aslında.

```
► $ ls /proc
```

```
► $ cat /proc/PID/status
```

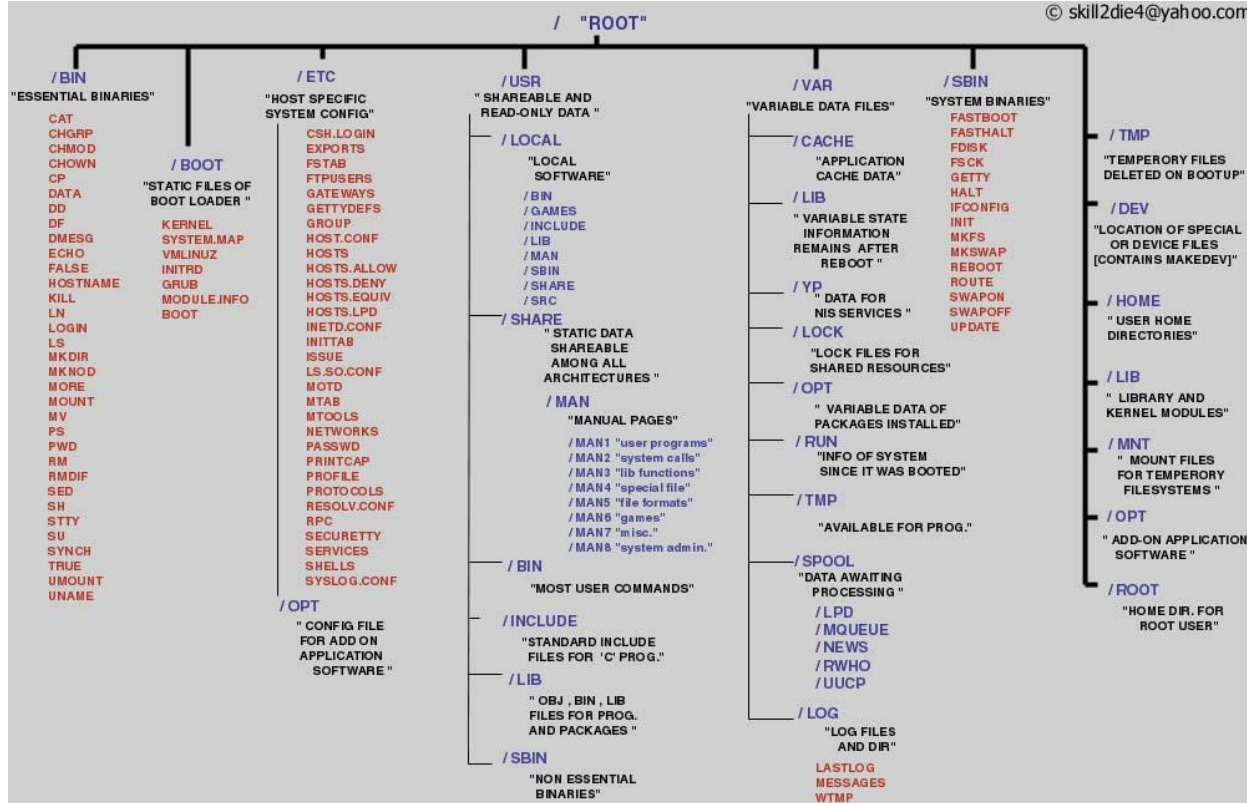
6.

DOSYA SİSTEMLERİ



Dosya Sistemi Hiyerarşisi

© skill2die4@yahoo.com



Dosya Sistemi Tipleri

► **Günlükleme(Journaling):** Bu özellik öntanımlı olarak gelir. Herhangi bir güç kaybından kaynaklı veya sistem kararsızlığından kaynaklı veri kayıplarını engellemek için yapılan işlemlerin logları ana sistem dışında bir yerde tutularak very kaybı önlenir.

En Bilindik Dosya Sistemi Tipleri

- ▶ **Ext4** : GNU/Linux sistemlerin hemen hemen hepsi bu dosya sistemini kullanmaktadır. Ext2 ve ext3 olan eski versiyonları ile uyumludur. 1 Exabyte ve 16 terabyte boyutlarını destekler.
- ▶ **Btrfs** : “Better or Butter FS” Linux için yeni bir dosya sistemidir. Snapshot özelliği ile gelir. Arttırılmış yedekleme ve daha yüksek performansı vardır. Henüz stabil ve uyumlu değildir.

- ▶ **XFS** : Yüksek performanslı dosya sistemidir. Medya sunucuları gibi yüksek veriyi depolayan sistemler için uygundur.
- ▶ **NTFS & FAT** : Windows dosya sistemidir.
- ▶ **NFS+** : Macintosh dosya sistemidir.

► \$ df -T

► Bu komut ile dosya sistemimizin kullanımını ve diğer disk bilgilerini görüntüleyebiliriz.

```
mehmet@ubuntu:~/Desktop$ df -T
df: /mnt/hgfs: Protocol error
Filesystem      Type      1K-blocks    Used Available Use% Mounted on
udev            devtmpfs   1995172        0   1995172    0% /dev
tmpfs           tmpfs      402892    11512   391380    3% /run
/dev/sda1       ext4      37024320 9351832 25768720   27% /
tmpfs           tmpfs      2014440     212   2014228    1% /dev/shm
tmpfs           tmpfs       5120         4     5116    1% /run/lock
tmpfs           tmpfs      2014440        0   2014440    0% /sys/fs/cgroup
tmpfs           tmpfs      402892      84     402808    1% /run/user/1000
```


Disk Anatomisi

- ▶ Hard diskler bölünebilir parçalardır. Örnek vermek gerekirse, /dev/sda1 ve /dev/sda2, /dev/sda tam diskinin iki parçasıdır. Diskleri bölümlemek sisteme yeni bir disk takmaktan daha kolaydır. Bu bölümleme işlemi verilerimizi birbirinden ayırmaya yardımcı olur.
- ▶ **Partition Table:** Her disk kendi bölüm tablosuna sahiptir. Bu sisteme diskin nasıl bölümlendirildiğini, hangi bölümün nerede başlayıp nerede bittiğini söyler. İki tip bölüm tablosu vardır: Master Boot Record (MBR) ve GUID Partition Table (GPT)

► Diskler verilerimizi organize etmek için bölümlerden oluşmuştur. Bu bölümler birbirleriyle çakışmayan bölümlerdir. Diskte herhangi bir şekilde biçim verilmemiş yer varsa bu ver *free* alandır.

► **MBR:**

- ▷ Geleneksel bölümlleme tablosudur,
- ▷ MBR'de en fazla 4 ana mantıksal bölüm oluşturulabilir,
- ▷ 2 terabytelık diskler kadar destekler.

► **GBT:**

- ▷Yeni disk bölümleme standardı ile gelir.
- ▷Sadece bir bölüm tipi vardır.
- ▷Her bölüm genel bir eşsiz ID'ye sahiptir.
- ▷Çoğunlukla UEFI tabanlı sistemlerde kullanılır.

► **Dosya Sistemi Yapısı:**

- ▷*Boot block*: Dosya sisteminin ilk bir kaç sektörüdür. İşletim sistemi tarafından bilgisayarı başlatabilmek için ihtiyaç duyulan bilgiler mevcuttur.
- ▷*Super block*: *boot block*'tan sonar tek blok olarak bulunur. Dosya sistemi hakkında dosya indeksi mantıksal blok boyutu ve dosya sistemi boyutu gibi bilgiler içerir

- *Inode table*: Dosyalarımızı yöneten bir veritabanı olarak düşünülebilir. Dosyalarımızın indekslendiği yerdir. Herbir dosya veya dizin eşsiz kodlara sahiptir.
- *Data blocks*: Gerçekten sahip olduğumuz dosyalar ve dizinler bulunur.
- Aşağıdaki görüntü MBR bölümleme tablosu örneğidir:

```
mehmet@ubuntu:~/Desktop$ sudo parted -l
[sudo] password for mehmet:
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number   Start    End      Size    Type    File system  Flags
  1       1049kB   38.7GB   38.7GB   primary ext4          boot
  2       38.7GB   42.9GB   4293MB   extended
  5       38.7GB   42.9GB   4293MB   logical  linux-swap(v1)
```

Disk Bölümleme

- ▶ Dosya sistemimizi bölümlmek için aşağıdaki araçları kullanabiliriz:
 - ▷ *fdisk*: komut satırından çalışır, sadece MBR destekler
 - ▷ *parted*: komut satırından çalışır GBT ve MBR destekler
 - ▷ *gparted*: *parted* aracının GUI versiyonudur
 - ▷ *gdisk*: komut satırından çalışır, sadece GBT destekler

Diski bağlamak(mount) ve ayırmak(umount)

- ▶ Dosya içeriğini görmeden önce disklerin/disk bölümlerinin sisteme bağlanması gereklidir.
- ▶ Disk bölümünün sisteme bağlanması bir dizine bağlanması ile gerçekleşir.

```
▶ $ sudo mount -t ext4 /dev/sdb2 /mydrive
```

- ▶ Böylece *sdb2* bölümü */mydrive* dizini ile erişilebilir hale geldi.

```
▶ $ sudo umount /mydrive
```

- ▶ Komutu ile bağladığımız diski/bölümü ayırabiliriz.

► Herbir diskin/bölümün evrensel eşsiz ID'sini görebilirsiniz:

► \$ sudo blkid

```
mehmet@ubuntu:~/Desktop$ sudo blkid
[sudo] password for mehmet:
/dev/sda5: UUID="75707b2a-9516-4066-a9a2-244837e02800" TYPE="swap"
/dev/sda1: UUID="3dd0a518-2a6e-4083-8cd2-5a64a41c66dc" TYPE="ext4"
```

► Disklerin/bölümlerin UUID'leri eşsiz olduğu için bu kodlarını kullanarak da bağlamamız mümkün:

► \$ sudo mount UUID=7570.....8000 /mydrive

/etc/fstab

► Bilgisayarımızı her açtığımızda sistemimize disklerimizin otomatik bağlanmasını istiyorsak diskimizi **/etc/fstab** dosyasına direk değiştirerek eklememiz gerekmektedir.

```
meahmet@ubuntu:~/Desktop$ cat /etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=3dd0a518-2a6e-4083-8cd2-5a64a41c66dc / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=75707b2a-9516-4066-a9a2-244837e02800 none swap sw 0 0
```

- UUID: Diskimizin eşsiz ID'si
- Mount point: bağlı olduğu dizin
- Type: disk/bölüm tipi
- Options: diğer bağlama seçenekleri
- Dump: dump aracı tarafından kullanılır
- Pass: disk kontrolü yapılıp yapılmayacağını belirler

swap

► Swap alanı işletim sistemi tarafından sanal hafıza(memory) olarak kullanılan alandır. Bu düşük hafızalı sistemlerin çakılmasını önlemek için kullanılır.

► Swap alanı eklemek için:

```
► $ mkswap /dev/sdb2  
► $ swapon /dev/sdb2
```

► Swap alanını silmek için:

```
► $ swapoff /dev/sdb2
```

► Swap alanı genellikle sahip olduğunuz RAM miktarının iki katı şeklinde olur.

Disk Kullanımı

► Diskimizin ne kadar kullanıldığını görmek için:

► \$ df -h

► komutu vermemiz yeterlidir. **-h** eki ile **df** komutu ile daha anlaşılır çıktı üretir.

Dosya Sistemi Bakımı

► Bazen dosya sistemimiz bozulmalara uğrayabilir.

Bu durumu düzeltmek için

► `$ sudo fsck /dev/sda`

► komutunu vermemiz yeterlidir.

TEŞEKKÜRLER!

KAYNAKLAR

- ▶ <http://www.linuxnix.com/suid-set-suid-linuxunix/>
- ▶ <https://linuxjourney.com/lesson/setuid-set-user-id>
- ▶ <http://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/06/linux-dosya-sistemleri>