

Dokuz Eylül Üniversitesi - Sistem Yönetimi Eğitimi

Yazan : Nuri Çilengir - nuriilengir@protonmail.com

İçindekiler

1. BSD Hakkında
2. FreeBSD Kurulumu
3. Kullanıcı Yönetimi ve Yetkiler
4. Temel Komutlar
5. Betik Yazımına Giriş
6. Process Yönetimi
7. Dizin Yapısı

BSD Hakkında

1.1 BSD Nedir?

BSD "Berkeley Software Distribution" 'in kısaltılmışıdır. Bu isim California Üniversitesi, Berkeley 'in kaynak kodu dağıtımı olan AT&T 'nin UNIX® 'i için bir eklentiler zinciridir. Bir çok özgür işletim sistemi projesi, 4.BSD-Lite olarak bilinen kaynak kodu dağıtımını temel kabul eder. Ek olarak bunlar, özellikle GNU projesi olmak üzere diğer birçok özgür yazılım projesini de kapsar. BSD 'in içeriği:

- BSD çekirdeği işlem zamanlama(task scheduling), hafıza yönetimi(memory management), SMP(Symmetric Multi-Processing), aygıt sürücüler ve diğerlerini kapsar.
- C kütüphanesi, sistem için API temellidir
- *BSD C kütüphanesi GNU projesi tabanlı dedildir, Berkeley kodu temellidir.*
- Kabuk, dosya araçları, derleyiciler(loader) ve linkerler gibi araçlar içerir.
- *Bazı araçlar dışında bir çoğu projesinden sağlanmıştır.*
- X Windows sistemi grafik ekran arabirimidir.

1.2 BSD Gerçek bir UNIX mi?

BSD işletim sistemi bir kopya değildir, fakat temelde UNIX'in atası sayılan AT&T'nin Research UNIX'inin açık kaynak kodundan oluşur.

1.3 BSD'nin Sahibi Kimdir?

Hiç kimse ya da hiçbir kuruluş BSD'nin sahibi değildir. BSD bütün dünya üzerinde teknik olarak üst düzeyde ve uzman birçok insan tarafından açık kaynaklı ve özgür bir şekilde geliştirilmektedir(bkz: [BSD LİSANSI](#)).

1.3 BSD'nin Hangi Sürümleri Vardır?

FreeBSD : "The Power to Serve"(Sunum Gücü). FreeBSD yüksek performans ve son kullanıcılar için kullanım kolaylığı sunar. Genellikle ISS firmaları için favori bir sürümdür. Ayrıca, en çok kullanıcıya sahip sürümdür. Dolayısıyla hata, problemler ve geliştirme birçok kaynak ve forum sunmaktadır.

NetBSD: "Of course it runs NetBSD"(Elbette NetBSD çalışır.). NetBSD azami seviyede taşınabilirlik hedefler. Ek olarak sade bir dizayna sahiptir. NetBSD el terminallerinden büyük sunuculara kadar her yerde çalışabilmektedir. Embedded cihazlar için çokça tercih edilmektedir. Özel olarak intel olmayan donanımlar için en iyi seçimdir.

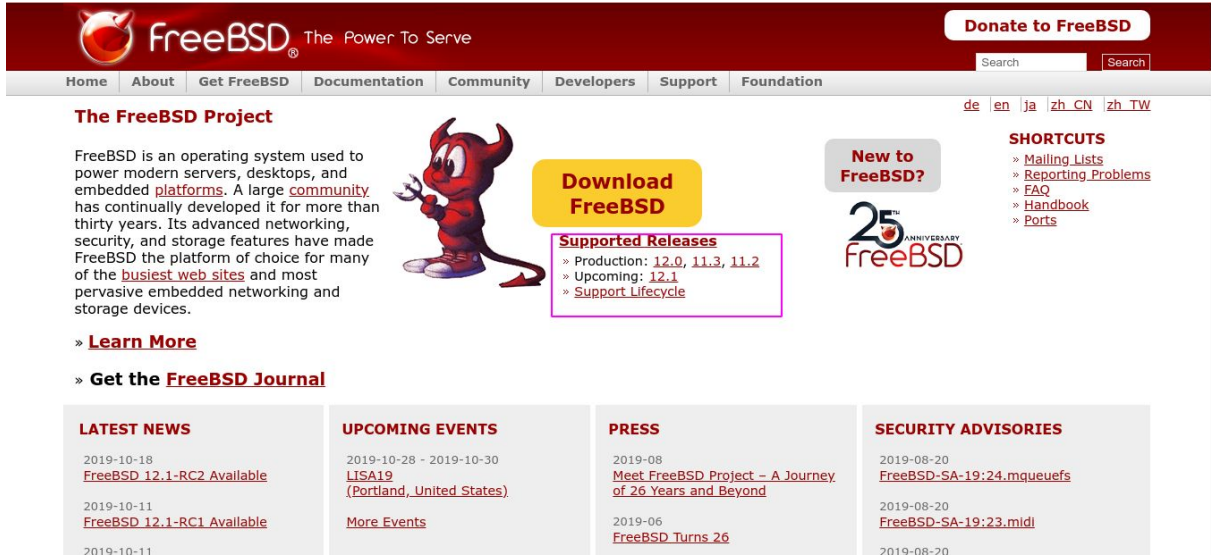
OpenBSD: "Only one remote hole in the default install, in more than 7 years"(7 yıldan fazla süredir sadece bir tane dışarıdan etkili güvenlik açığı). OpenBSD güvenlik ve Kod güzelliği hedefler. Devlet organizasyonlarında borsa ve bankalar gibi güvenliğin ön plana çıktığı merkezler için, çok sıkı kod denetimleri ve testleri gerçekleştirir ve bunu açık kaynak geliştirme modeline uygun olarak yapar.

Bunların haricinde özgür lisanslara sahip olmayan iki farklı BSD çeşidi de bulunmaktadır. Bunlar BSD/OS ve Apple Mac OS X'dir.

2. BSD Kurulumu

2.1 BSD Edinmek

<https://www.freebsd.org> sayfasına giderek. Ekte işaretli desteklenen STABLE sürümleri kontrol edebilirsiniz.



Daha sonra Download BSD sekmesinden eğer 64 bit mimari kullanıyorsanız amd64, 32 bit mimari kullanıyor iseniz de i386 mimarisini seçiyoruz.

FreeBSD 12.0-RELEASE

Installer Images	Virtual Machine Images	SD Card Images	Documentation
amd64 i386 powerpc powerpc64 sparc64 aarch64	README amd64 i386 aarch64	RPI-B BANANAPI BEAGLEBONE CUBIEBOARD CUBIEBOARD2 CUBOX-HUMMINGBOARD GENERICSD RPI2 PANDABOARD WANDBOARD PINE64 PINE64-LTS RPI3	Released: December, 2018 Release Notes Readme Hardware Compatibility List Installation Instructions Errata Signed Checksums

Burada kullanım şeklinize göre istediğimiz image'yi seçiyoruz.

Index of /ftp/releases/amd64/amd64/ISO-IMAGES/12.0/

File Name	File Size	Date
Parent directory/	-	-
CHECKSUM.SHA256-FreeBSD-12.0-RELEASE-amd64	1171	2018-Dec-07 08:10
CHECKSUM.SHA512-FreeBSD-12.0-RELEASE-amd64	1811	2018-Dec-07 08:09
FreeBSD-12.0-RELEASE-amd64-bootonly.iso	346167296	2018-Dec-07 05:09
FreeBSD-12.0-RELEASE-amd64-bootonly.iso.xz	77782060	2018-Dec-07 05:09
FreeBSD-12.0-RELEASE-amd64-disc1.iso	892467200	2018-Dec-07 05:08
FreeBSD-12.0-RELEASE-amd64-disc1.iso.xz	623944576	2018-Dec-07 05:08
FreeBSD-12.0-RELEASE-amd64-dvd1.iso	3884677120	2018-Dec-07 05:18
FreeBSD-12.0-RELEASE-amd64-dvd1.iso.xz	2888051180	2018-Dec-07 05:18
FreeBSD-12.0-RELEASE-amd64-memstick.img	975905280	2018-Dec-07 05:20
FreeBSD-12.0-RELEASE-amd64-memstick.img.xz	627656096	2018-Dec-07 05:20
FreeBSD-12.0-RELEASE-amd64-mini-memstick.img	385851904	2018-Dec-07 05:21
FreeBSD-12.0-RELEASE-amd64-mini-memstick.img.xz	81653720	2018-Dec-07 05:21

Not : Kurulum VM üzerinde gerçekleştirilecektir fakat arzu ederseniz diskte ayırdığınız herhangi bir boş alana aynı talimatları uygulayarak kurulum işlemi gerçekleştirebilirsiniz.

Not 2: Eğer fiziksel bir disk üzerine kurulum işlemi yapacaksanız ve diskinizde ayrıca bir windows barındırmak istiyorsanız önce windows kurulumunu daha sonra BSD kurulumu yapmanızı öneririm.

2.2 BSD Kurulumu

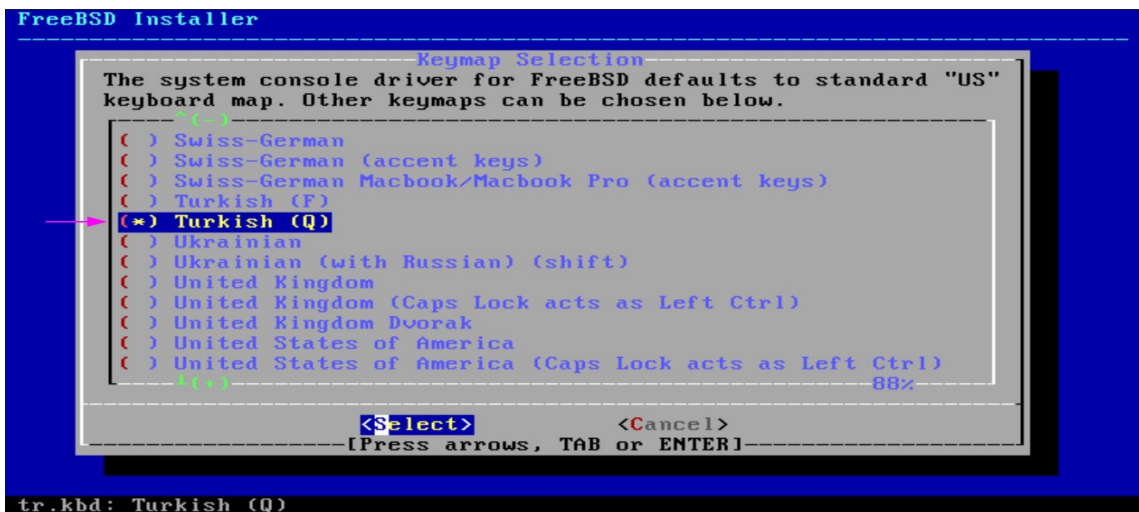
İlk olarak BSD boot menüsü bizi karşılıyor. Seçeneklerin ne olduğuna daha ileride değineceğiz. Bu yüzden, şimdilik ENTER'e basıp multi user olarak sistemimizi boot ediyoruz.



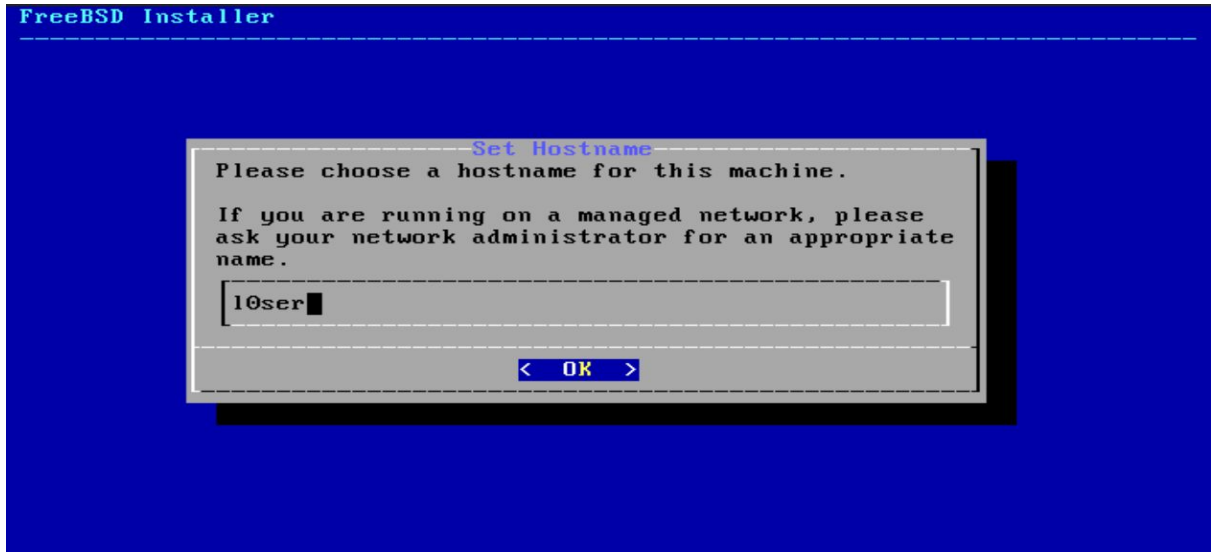
Sistem boot edildikten sonra "FreeBSD installer" ekranı karşımıza çıkıyor. Install ile devam ederek kurulumu başlıyoruz.



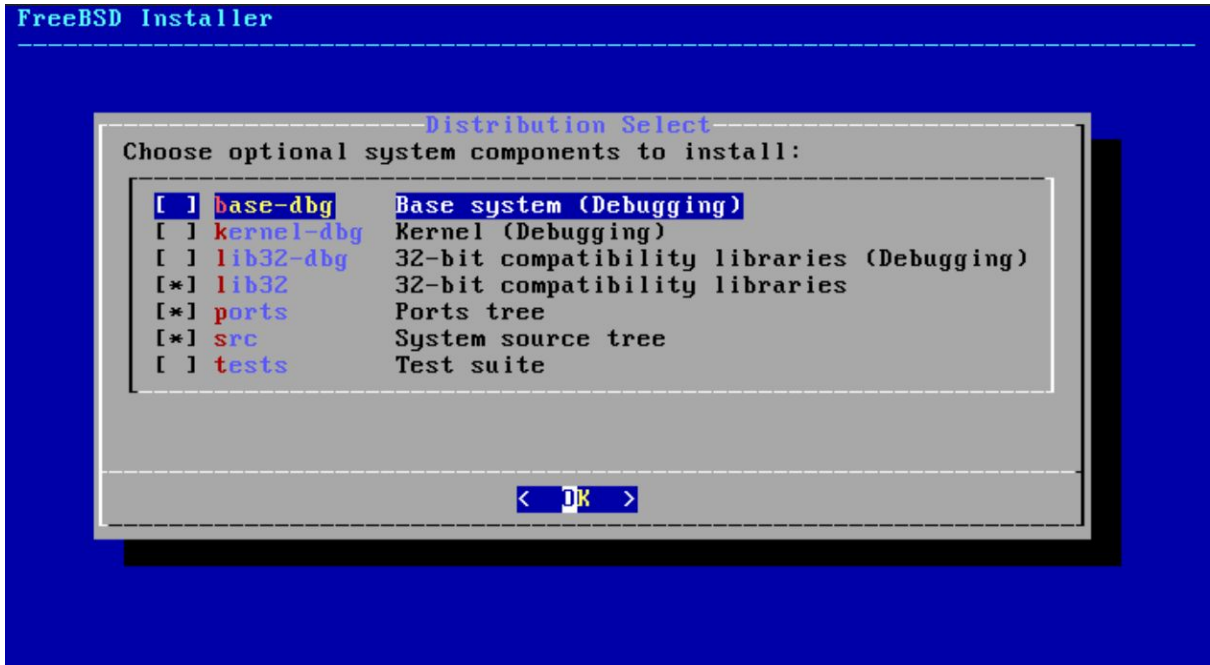
1. Aşama: Klavye düzenini ayarlıyoruz.



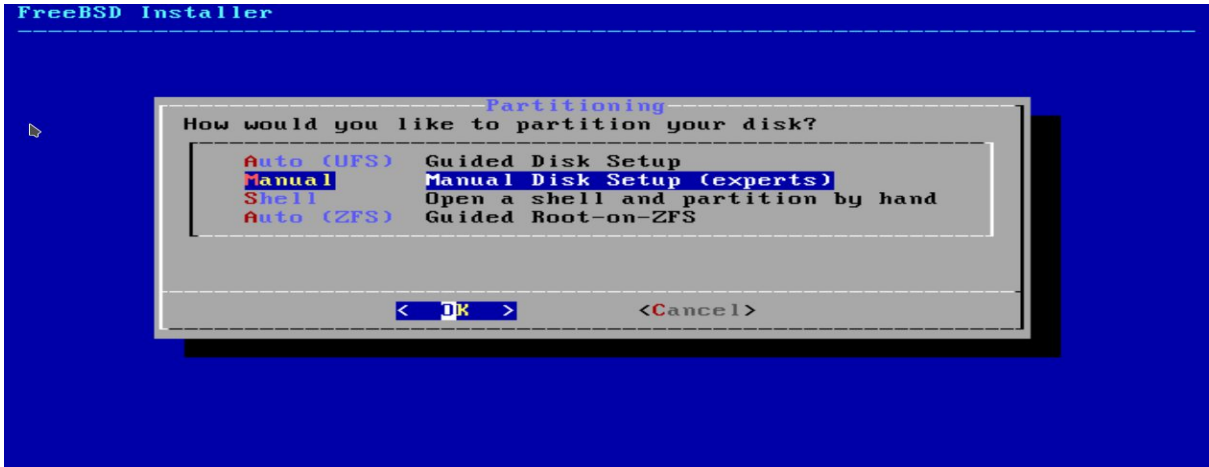
2. Aşama : Sisteme bir "Hostname" (Bilgisayar Adı) atıyoruz.



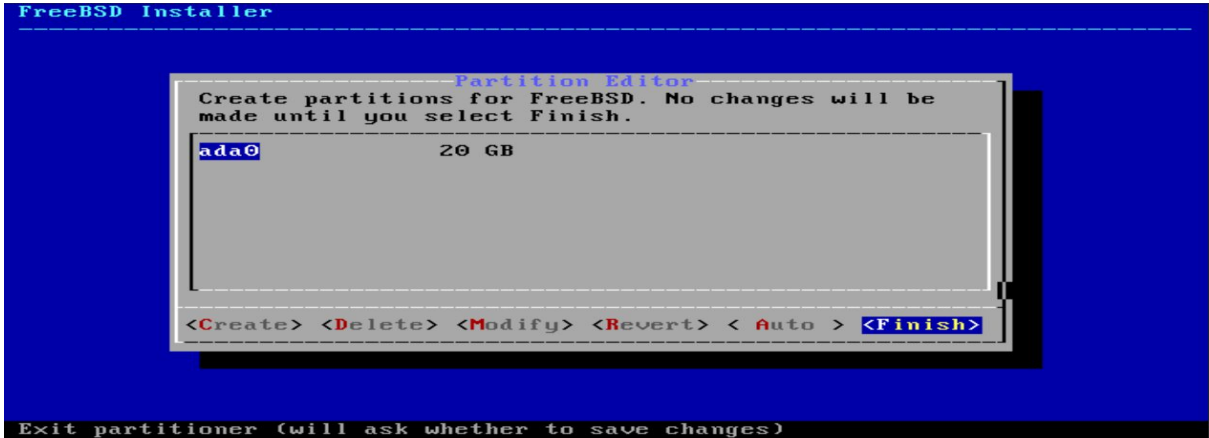
3. Aşama: Yüklenecek isteğe bağlı bileşenleri seçiyoruz. lib32, 32 bit uygulamaları FreeBSD'nin 64 bit sürümünde çalıştırmak için uyumlulukları içerir. ports, FreeBSD Ports Collection, üçüncü taraf yazılım paketlerinin indirilmesini, derlenmesini ve kurulmasını otomatikleştiren bir dosya koleksiyonudur. src, Hem çekirdek hem de kullanıcı alanı için FreeBSD kaynak kodunun tamamını içerir. Uygulamaların çoğunluğu için gerekli olmamakla birlikte, Ports Koleksiyonundan aygıt sürücülerini (driverlar), çekirdek modülleri veya bazı uygulamalar oluşturmak gerekebilir.



4. Aşama : Disk bölümlenme adımı geliyor. Disk'i manuel olarak düzenleyeceğiz.



5. Aşama : GPT genellikle amd64 bilgisayarlar için en uygun seçimdir. GPT ile uyumlu olmayan daha eski bilgisayarların MBR kullanmaları gerekir. Diğer bölüm şemaları genellikle yaygın olmayan veya daha eski bilgisayarlar için kullanılır. Bu yüzden bölümlendirme şemamızı GPT olarak seçiyoruz. Referans kurulum bilgileri, VM disk boyutu 20 GB ve Memory'i 4 GB olarak ayarlandı. Dolayısıyla bölümlendirme yaparken kendi şartlarınızı göz önünde bulundurarak bölümlendirme yapmalısınız.



Exit partitioner (will ask whether to save changes)



Bootable on most x86 systems

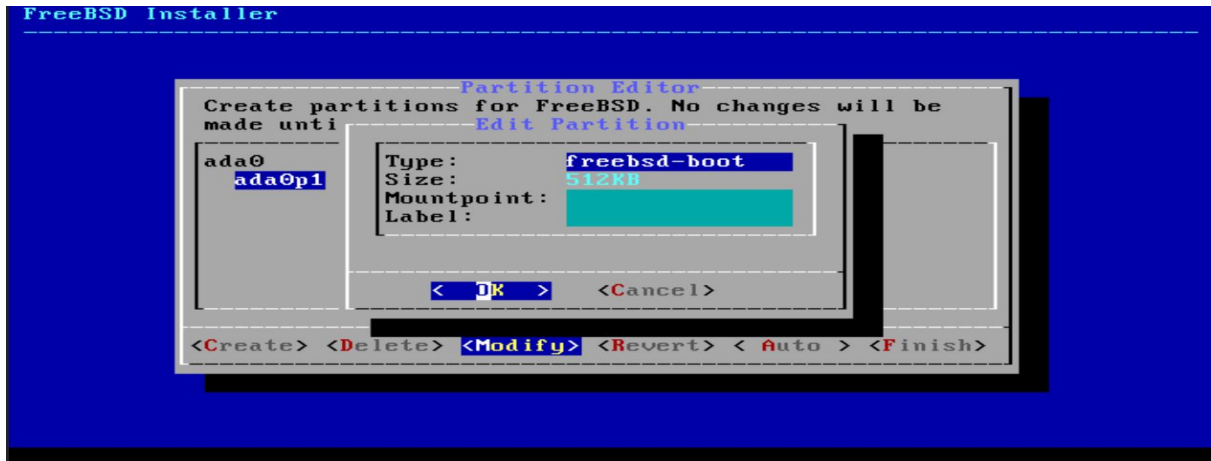
Standart bir FreeBSD GPT kurulumu en az üç bölüm kullanır:

- freebsd-boot - FreeBSD önyükleme kodunu tutar.
- freebsd-ufs - Bir FreeBSD UFS dosya sistemidir.
- freebsd-swap - FreeBSD takas alanıdır.
-

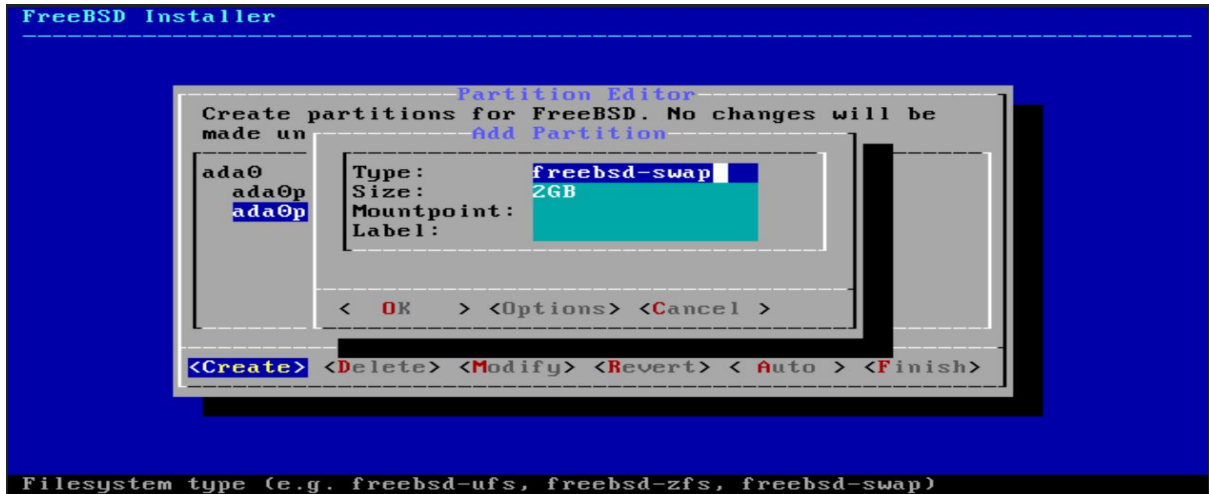
Biz sistemimizde birden fazla dosya sistemi belirlemek istiyoruz. Her işlemi / altında yapmak bazen tavsiye etmediğim zaman zaman sıkıntı çıkabilmektedir. Bu yüzden, dosya yollarımızı malloc değil kendi belirlediğimiz space'e bağlıyoruz.

Partition Type	Size	Mountpoint	Label
freebsd-boot	512K		
freebsd-ufs	2G	/	rootfs
freebsd-swap	4G		swap
freebsd-ufs	2G	/var	varfs
freebsd-ufs	1G	/tmp	tmpfs
freebsd-ufs	tercihen diskin geri kalanı	/usr	usrfs

a. freebsd-boot alanını oluşturuyoruz.



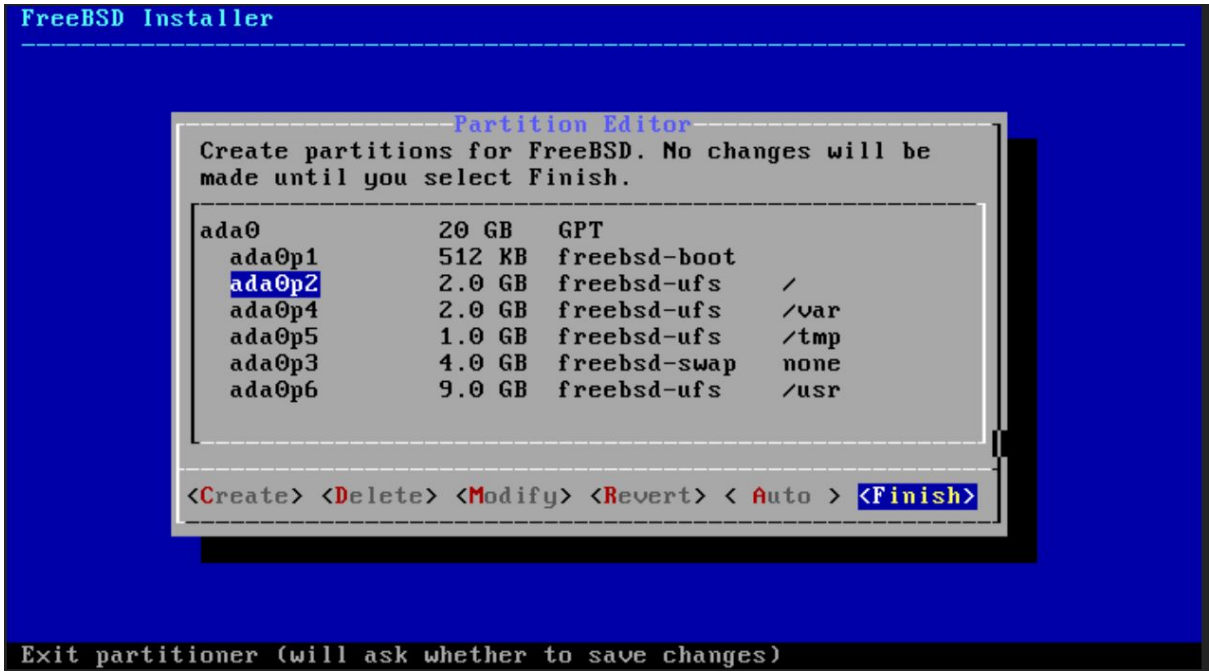
b. swap alanını oluşturuyoruz.



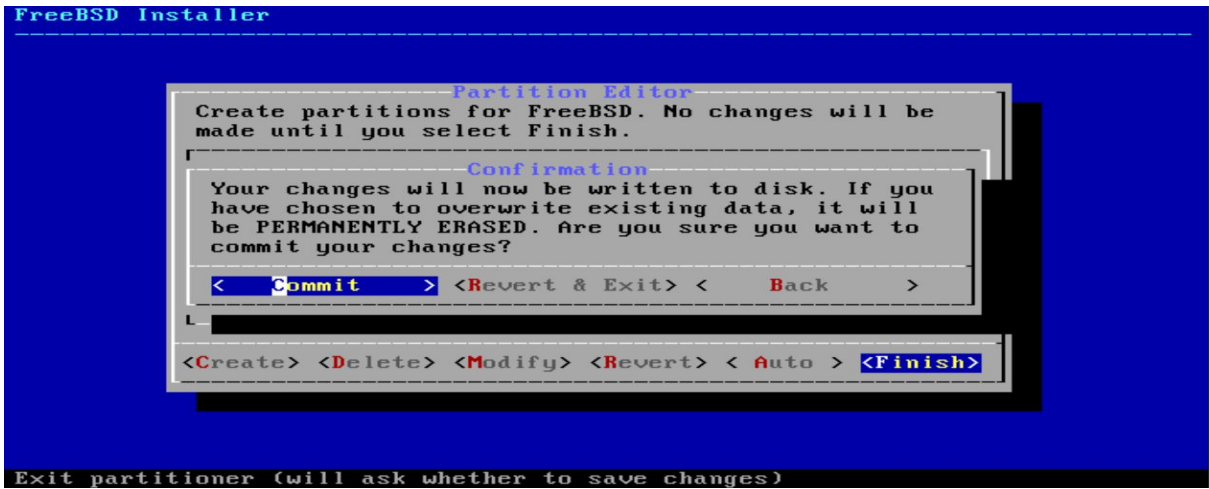
c. root alanını oluşturuyoruz.



d. Ardından diğer bölümlerimizi oluşturuyor finish diyerek bölümlendirme işlemini tamamlıyoruz.



e. Sistem yüklemesini başlatmak için, [Commit]'ı seçiyoruz.



Kurulum başladıktan sonra sistem ilk olarak, yükleyici seçilen diskleri biçimlendirir ve bölümleri başlatır. Daha sonra, bir bootonly medya durumunda, seçilen bileşenleri indirir. Daha sonra, yükleme sırasında bozulmadıklarından veya kurulum medyasından yanlış okumadıklarından emin olmak için dağıtım dosyalarının bütünlüğü doğrulanır. En sonunda ise doğrulanan dağıtım dosyaları diske çıkartılır.

```
FreeBSD Installer
-----
Archive Extraction
Extracting distribution files...
base.txz... | [ 30% ]
kernel.txz  [ Pending ]
lib32.txz   [ Pending ]
ports.txz   [ Pending ]
src.txz     [ Pending ]

Overall Progress:
[ 7% ]

7367 files read @ 2,455.0 files/sec. [1/4 busy/wait]
```

```
FreeBSD Installer
-----
Archive Extraction
Extracting distribution files...
base.txz      [ 50% ]
kernel.txz    [ 50% ]
lib32.txz     [ 50% ]
ports.txz...  \ [ 40% ]
src.txz       [ Pending ]

Overall Progress:
[ 69% ]

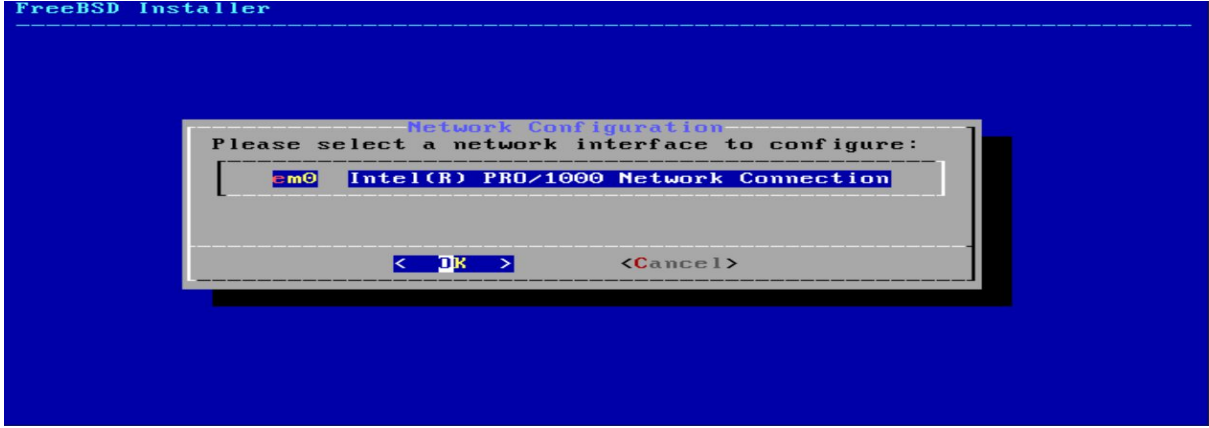
97871 files read @ 1,812.0 files/sec. [1/1 busy/wait]
```

Temel sistem kurulumu başarılı bir şekilde tamamladık !!

6. Aşama : Root kullanıcı parolasının belirlenmesi. Bu bölümde Root kullanıcımız için bir parola oluşturuyoruz.

```
FreeBSD Installer
=====
Please select a password for the system management account (root):
Typed characters will not be visible.
Changing local password for root
New Password:█
```

7. Aşama : Ağ yapılandırması. Bu bölümde yapılandırmak istediğimiz ağ arayüzlerinin listelerini görmekteyiz. Yapılandırmak istediğimiz arayüzü seçerek devam ediyoruz.



Eğer cihazınız yükleme esnasında internete çıkmıyorsa wifiniz üzerinden ağı taracaktır ve sizden bağlanmak istediğiniz ağa ait bilgileri girmenizi isteyecektir.

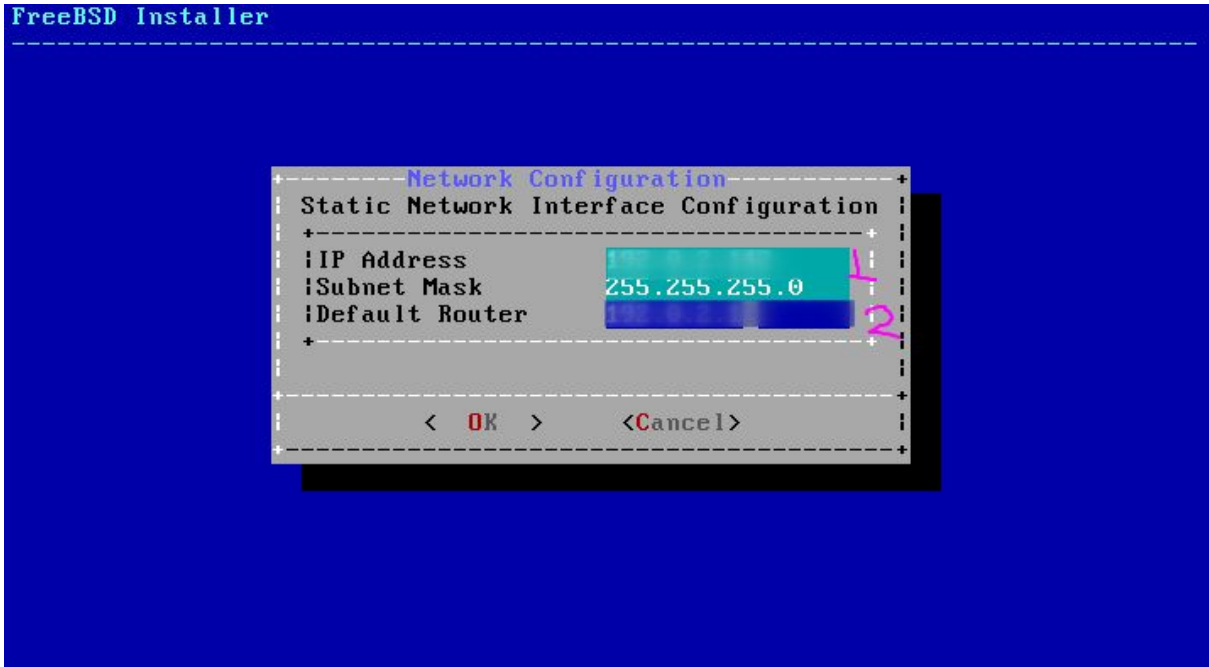
- a. IPv4 adresinin Ethernet veya kablosuz arabirimde yapılandırılıp yapılandırılmayacağını seçiyoruz. IPv4 yapılandırmasının iki yöntemi vardır. İlk seçenekte, otomatik olarak IP adresini DHCP üzerinden alır. Diğerinde ise, adres bilgilerinin statik bir yapılandırma olarak manuel olarak girilmesi gerekir.



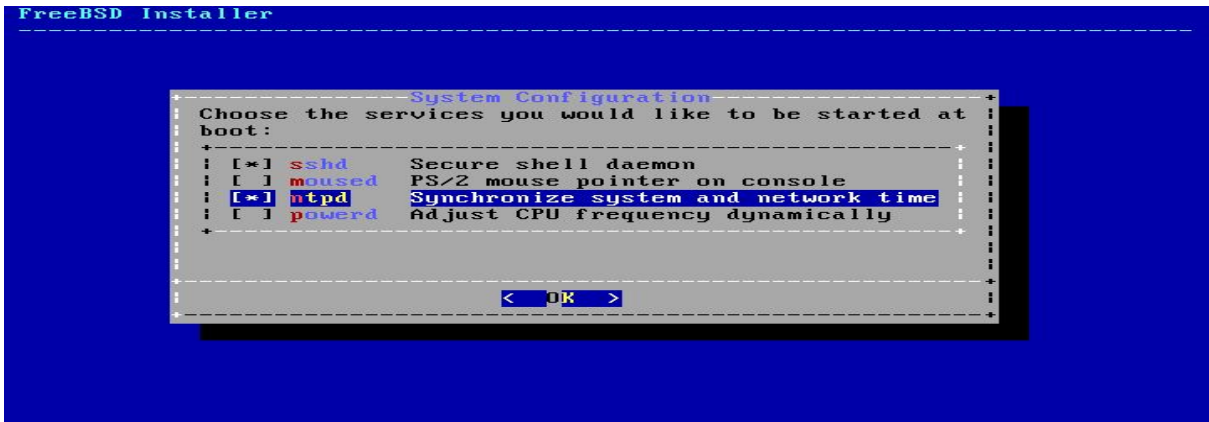
- b. Bir DHCP sunucusu varsa, ağ arayüzünü otomatik olarak yapılandırmak için sonraki menüden [Yes] seçin.



- c. (1) IP Address alınina internete çıkış adresinizi girmelisiniz. Default router ise, Ağın gateway'inizin IP adresi olmalıdır (örn: ev ortamında modem adresi).

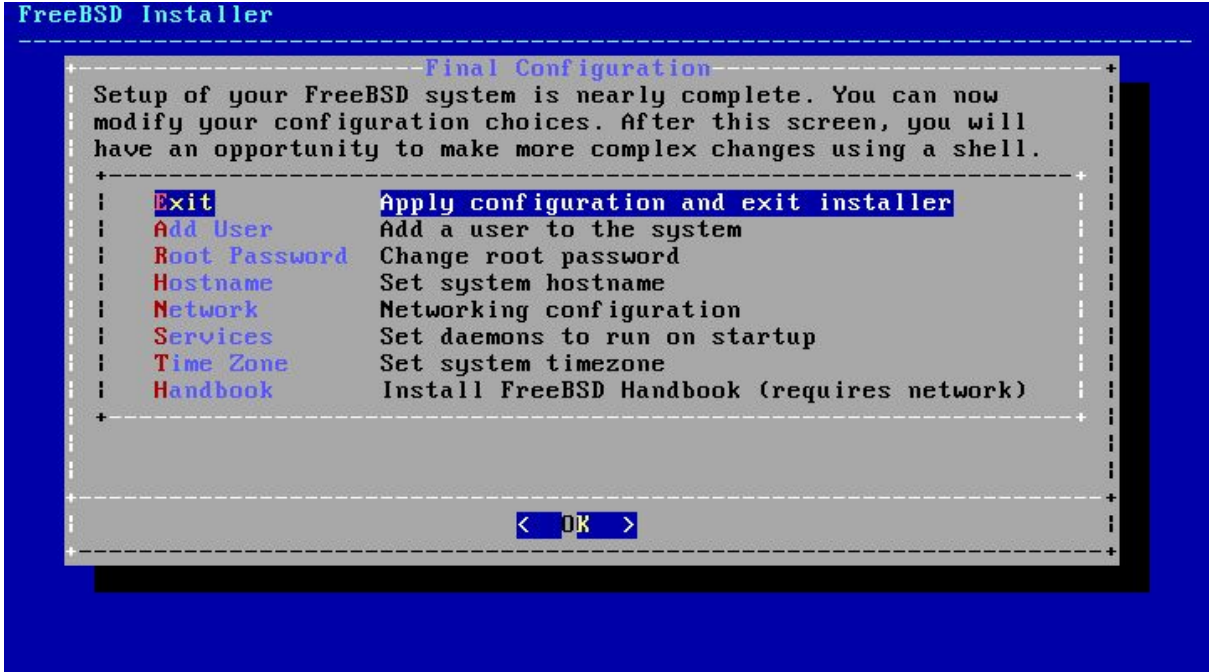


8. Aşama: Bu bölümde tarih ve saat ayarlarını gerçekleştiriyoruz.
9. Aşama : Sistem Servisleri Etkinleştirme. Bu aşama, sistem önyüklediğinde hangi sistem hizmetlerinin başlatılacağını yapılandırmak için kullanılır. Bu servislerin tümü isteğe bağlıdır.



10. Aşama : Kullanıcı oluşturma. Bu bölüm'ü kurulum sonrası terminal üzerinden yapacağız.

11. Aşama : Kurulum Tamamlandı !!!! Eğer değiştirmek istediğiniz bir ayar varsa tekrar düzenleyip kurulumu sonlandırabilirsiniz.



3. Kullanıcı Yönetimi ve Yetkiler

FreeBSD yüklü bilgisayarı kullanabilmek için kullanıcının sisteme giriş yapması gerekir. Kullanıcı hesaplarının açılması, kapatılması ve zaman zaman değiştirmesi sistem yönetimi için oldukça önemlidir. Sisteme bütün erişimler bu hesaplar üzerinden gerçekleştirilir ve bütün işlemler yine bu kullanıcılar tarafından çalıştırılır.

3.1 Hesap Özellikleri

Her kullanıcı hesabı, kullanıcı adı, ve bununla birlikte tanımlanan bazı bilgilerle oluşturulur. Kullanıcı bilgileri /etc/passwd, /etc/master.passwd dosyalarında tutulur. Bu dosyalarda her satırda ayrı bir kullanıcı bilgisi yer alır. Grup bilgileri ise /etc/group dosyasında yer alır. Örnek bir kullanıcı satırı şu şekildedir.

```
l0ser:$6$.HBH4w4Xjz4HmGrS$A4.dsXPEI3u6wmNXX4Ak9giVI1lq9gXhsWj/0Nv4IA1RP779WXQFKG
urtBt2f1SkYwxVDErTs.LJDxdSRET80eE1:1001:1001::0:0:XXX
YYY:/home/xxx:/usr/local/bin/bash
```

görüldüğü üzere her satırdaki değişken ':' ile ayrılmıştır. Bu değişkenler sırasıyla şunlardır.

- **Kullanıcı Adı** : Kullanıcı adı, oturum açma işlemi için kullanılır. Her kullanıcının benzersiz bir kullanıcı adı olmalıdır. Kullanıcı adları büyük harflerden, küçük harflerden, sayılardan '_' ve '-' den oluşabilir. Kullanıcı adı en fazla 16 karakter olmalıdır.
- **Parola**: Her hesabın bir parolası olmalıdır. Kullanıcı parolaları /etc/master.passwd dosyasında şifrelenmiş bir şekilde tutulmaktadır.
- **Kullanıcı ID'si (UID)**: UID kullanıcıyı FreeBSD sistemine benzersiz bir şekilde tanımlamak için kullanılan bir sayıdır. 0-65535 aralığında benzersiz bir sayı olmalıdır. Root kullanıcısının UID değeri 0'dır.
- **Grup ID'si (GID)**: 0-65535 aralığında olup kullanıcıların dahil olduğu gurubu tanımlayan numaradır. Örneğin Mai sevisini belirli kullanıcıların kullanabilmesi gibi.
- **Giriş Sınıfı (Login Class)** : Giriş sınıfı kullanıcıların çevre değişkenlerini atamak, kaynaklara erişimi belirlemek için kullanılır. Bu değer default olarak 0 gelir.
- **Parola değiştirme zamanı, Hesap son kullanma tarihi,**
- **Kullanıcı tam adı**
- **Kullanıcı ana dizini**
- **Kullanıcı kabuğu (shell)**

3.2 Yetkili Kullanıcı

Genellikle root adı verilen süper kullanıcı hesabı, sistemi yönetmek için sınırsız yetki ile kullanılır. Bu nedenle, posta gönderme ve alma, sistemin genel olarak keşfedilmesi veya programlama gibi günlük işler için kullanılmamalıdır.

3.3 Sistem Hesapları

Sistem hesapları, DNS, e-posta,web,ftp sunucu, gibi servisleri çalıştırmak için kullanılır. Sistem hesabı açmanın birincil nedeni güvenli bir işlem gerçekleştirmektir. Sistem servisleri eğer root olarak çalışırsa olası güvenlik problemleri ortaya çıkacaktır. daemon, operator, bin, man, sshd, ve news gibi kullanıcı adları FreeBSD'nin kullandığı default kullanıcı adlarıdır.

3.4 Kullanıcı işlemleri

Kullanıcı hesaplarını yönetmek için birçok komut mevcuttur. Başlıcaları;

Komut	İşlem
adduser	Kullanıcı ekler.
rmuser	Kullanıcı siler.
chpasswd	Kullanıcı bilgilerini değiştirir.
passwd	Kullanıcı parolalarını değiştirmek için kullanılır.
pw	Bütün kullanıcı işlemlerini güçlü ve sağlam bir komut.
su	Bir kullanıcıdan diğer kullanıcıya geçmeye yarar.

a. Adduser Komutu Örnek uygulama

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
```

b. **rmuser** : Bir kullanıcıyı sistemden tamamen kaldırmak için, rmuser komutunu root kullanıcı olarak çalıştırılmalıdır. Bu komut aşağıdaki adımları yerine getirir:

1. Varsa, kullanıcının crontab işlemini kaldırır.
2. Kullanıcıya bağlı süreçleri tüm işleri kaldırır.
3. Kullanıcıya ait süreçleri tüm işleri sonlandırır.
4. Kullanıcıyı sistemin yerel parola dosyasından kaldırır.
5. İsteğe bağlı olarak, kullanıcının sahibi olduğu kullanıcının giriş dizinini kaldırır.
6. Kullanıcı Grup dosyasındaki tüm gruplardan çıkartılır.
7. Kullanıcıya gelen e-postaları /var/mail altından siler.
8. Kullanıcın /var/tmp altındaki dosyaları silinir.

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Removing user (jru): mailspool home passwd.
```

c. **chpass** : Herhangi bir kullanıcı varsayılan shell'i ve kullanıcı hesaplarıyla ilişkili kişisel bilgilerini değiştirmek için chpass'ı kullanabilir. Root kullanıcısı chpass'ı herhangi bir kullanıcının bilgilerini değiştirmek için kullanabilir.

Örnek : Root kullanıcısı olarak chpass ile düzenleme.

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information
```

d. **passwd** : Herhangi bir kullanıcı parolasını passwd kullanarak kolayca değiştirebilir. Yanlışlıkla veya yetkisiz değişiklikleri önlemek için, bu komut yeni bir parola ayarlanmadan önce kullanıcının orijinal parolasını ister:

```
% passwd
Changing local password for jru.
Old password:
New password:
```



```
Retype new password:
passwd: updating the database...
passwd: done
```

- e. **pw** : komut satırından çalışan ve kullanıcı oluşturma, silme, kullanıcı bilgilerini güncelleme, kullanıcı ve grupları görüntüleme gibi işlemlerin yapılabildiği bir araçtır. Sadece root kullanıcısı tarafından kullanılabilir.

3.5 toor Kullanıcısı

toor ismi, root isminin tersten yazımıdır. root kullanıcısının haklarına sahip ve root kullanıcısına alternatif bir kullanıcıdır. Kabuk olarak /bin/sh kullanır. root kullanıcısının parolası unutulduğunda ya da root'un kabuğu /etc/master.passwd dosyasında yanlış yazıldığında root kullanıcısı sisteme giremez. Bu gibi ve daha farklı root problemleri durumunda toor kullanıcı alternatif olarak işlem yapabilir. Öntanımlı olarak toor kullanıcısının parolası olmadığından bu kullanıcı ile sisteme giriş yapılamaz. Bu yüzden toor kullanıcısı için parola atanmalıdır.

```
# passwd toor
```

3.6 Yetkiler ve İzinler

BSD ve Linux sistemler çok kullanıcıli işletim sistemleridir. Bu yüzden BSD ve Linux sistemler için güvenlik son derece önemlidir. Windows gibi lakayt bir tutum sergilenmez. Yetki düzenlemeleri üç temel için yapılmaktadır. Okuma(read: r) yazma(write: w) ve çalışma(execute: x). Her bir yetkinin sayısal bir değeri vardır :

İşlem	Decimal	Binary
execute	1	001
write	2	010
read	4	100

Bu değerlerin birleşiminden diğer izinler elde edilir.

Value	Permission	Directory Listing
0	No read, no write, no execute	---
1	No read, no write, execute	--x
2	No read, write, no execute	-w-
3	No read, write, execute	-wx
4	Read, no write, no execute	r--
5	Read, no write, execute	r-x
6	Read, write, no execute	rw-

Eğer / dizini altında `ls -la` komutunu çalıştırırsak aşağıdaki çıktıyı almış olacağız.

```

-rw-r--r--  2 root  wheel   951 Dec  7  2018 .cshrc
-rw-r--r--  2 root  wheel   470 Dec  7  2018 .profile
drwxrwxr-x  2 root  operator 512 Oct 21 01:11 .snap
-r-----  1 root  wheel 16777216 Oct 21 01:11 .sujournal
-r--r--r--  1 root  wheel  6177 Dec  7  2018 COPYRIGHT
drwxr-xr-x  2 root  wheel  1024 Dec  7  2018 bin
drwxr-xr-x  9 root  wheel  1536 Oct 21 01:17 boot
dr-xr-xr-x  8 root  wheel   512 Oct 21 01:17 dev
-rw-----  1 root  wheel  4096 Oct 21 01:17 entropy
drwxr-xr-x 25 root  wheel  2560 Oct 21 01:47 etc
lrwxr-xr-x  1 root  wheel    8 Oct 21 01:16 home -> usr/home
drwxr-xr-x  4 root  wheel  1536 Dec  7  2018 lib
drwxr-xr-x  3 root  wheel   512 Oct 21 01:12 libexec
drwxr-xr-x  2 root  wheel   512 Dec  7  2018 media
drwxr-xr-x  2 root  wheel   512 Dec  7  2018 mnt
drwxr-xr-x  2 root  wheel   512 Dec  7  2018 net
dr-xr-xr-x  2 root  wheel   512 Dec  7  2018 proc
drwxr-xr-x  2 root  wheel  2560 Dec  7  2018 rescue
drwxr-xr-x  2 root  wheel   512 Dec  7  2018 root
drwxr-xr-x  2 root  wheel  2560 Dec  7  2018 sbin
lrwxr-xr-x  1 root  wheel   11 Dec  7  2018 sys -> usr/src/sys
drwxrwxrwt  7 root  wheel  1024 Oct 21 01:49 tmp
drwxr-xr-x 17 root  wheel  1024 Oct 21 01:16 usr
drwxr-xr-x 25 root  wheel  1024 Oct 21 01:17 var
root@l0ser:~ # █

```

Görüldüğü üzere her izin ve dosyanın belirli izinleri en soldaki sütunda görülmektedir. Bu bölümde izinler 1+3+3+3 şeklinde 10 sütundan/karakterden oluşmaktadır. İlk sütun(en soldaki) karakter, bu satırdaki öğenin türünü(dizin, dosya, link, soket vs) belirtir. İlk üç karakter (rwx) o öğenin sahibinin yetkilerini belirtir. Sonraki üç karakter, dosya sahibiyle aynı kullanıcı grubunda bulunan kullanıcıların yetkilerini, son üç karakter ise diğer kullanıcıların erişim yetkilerini belirler.

Örnekler : Dosya ve izin yetkileri

İlk olarak bir dosya sample.txt adında bir dosya oluşturalım ve oluşturduğumuz dosyanın izinlerini inceleyelim.

```

root@l0ser:~ # ls -la
total 24
drwxr-xr-x  2 root  wheel   512 Dec  7  2018 .
drwxr-xr-x 19 root  wheel  1024 Oct 21 02:55 ..
-rw-r--r--  2 root  wheel   951 Dec  7  2018 .cshrc
-rw-r--r--  1 root  wheel   149 Dec  7  2018 .k5login
-rw-r--r--  1 root  wheel   392 Dec  7  2018 .login
-rw-r--r--  2 root  wheel   470 Dec  7  2018 .profile
root@l0ser:~ # touch sample.txt
root@l0ser:~ # ls -la
total 24
drwxr-xr-x  2 root  wheel   512 Oct 20 18:52 .
drwxr-xr-x 19 root  wheel  1024 Oct 21 02:55 ..
-rw-r--r--  2 root  wheel   951 Dec  7  2018 .cshrc
-rw-r--r--  1 root  wheel   149 Dec  7  2018 .k5login
-rw-r--r--  1 root  wheel   392 Dec  7  2018 .login
-rw-r--r--  2 root  wheel   470 Dec  7  2018 .profile
-rw-r--r--  1 root  wheel    0 Oct 20 18:52 sample.txt
root@l0ser:~ # █

```

sample.txt dosyamızın yetkileri rw-r--r-- şeklinde gözüküyor. Dosya ve dizin yetkileri sistem yöneticisi tarafından chmod komutuyla değiştirilebilir. Şu an sample.txt dosyamız bizim tarafımızdan yazılıp okunabiliyor. Grup ve diğer kullanıcılar ise sadece okuma hakkına sahip. Bu dosyanın sadece bize ait olmasını istediğimizi farzedelim.

```
root@l0ser:~ # chmod 700 sample.txt
root@l0ser:~ # ls -la
total 24
drwxr-xr-x  2 root  wheel   512 Oct 20 18:52 .
drwxr-xr-x 19 root  wheel 1024 Oct 21 02:55 ..
-rw-r--r--  2 root  wheel   951 Dec  7 2018 .cshrc
-rw-r--r--  1 root  wheel   149 Dec  7 2018 .k5login
-rw-r--r--  1 root  wheel   392 Dec  7 2018 .login
-rw-r--r--  2 root  wheel   470 Dec  7 2018 .profile
-rwx----- 1 root  wheel     0 Oct 20 18:52 sample.txt
root@l0ser:~ #
```

```
root@l0ser:~ # chmod 711 sample.txt
root@l0ser:~ # ls -la
total 24
drwxr-xr-x  2 root  wheel   512 Oct 20 18:52 .
drwxr-xr-x 19 root  wheel 1024 Oct 21 02:55 ..
-rw-r--r--  2 root  wheel   951 Dec  7 2018 .cshrc
-rw-r--r--  1 root  wheel   149 Dec  7 2018 .k5login
-rw-r--r--  1 root  wheel   392 Dec  7 2018 .login
-rw-r--r--  2 root  wheel   470 Dec  7 2018 .profile
-rwx--x--x  1 root  wheel     0 Oct 20 18:52 sample.txt
root@l0ser:~ #
```

4. Temel Komutlar

4.1 Dosya/Dizin Komutları

a. Dosya Listeleme

Sistemde var olan dosya ve dizinleri listelemek için **ls** (list directory) **ls** komutunu kullanıyoruz. Genel yapısı;

```
root@l0ser # ls [İşlev (-ABCFGHILPRSTUWZabcdghiklmnopqrstuvwxyz1,)]  
[--color=when] [-D format] [dosya veya dizin]
```

şeklinde. **ls** komutu herhangi bir parametre verilmeden de kullanılabilir. Bu şekilde kullanılırsa o an bulunduğunuz dosyayı listeler. Görüldüğü üzere opsiyonel olarak bir ton parametre alabilmekte. Bunlardan bazılarını inceleyelim.

-a: Tüm dosya ve dizinleri listeler **-a** parametresi **ls** komutunun dotfile(gizli dosya) olarak adlandırdığımız dosya ve dizinleri de ekrana basar.

```
root@l0ser:/usr/home/derectus # ls -a  
.  
..  
.cshrc  
.login  
.login_conf  
.mail_aliases  
.mailrc  
.profile  
.shrc  
deneme1  
deneme2  
deneme3  
deneme4  
test1  
test2  
test3
```

-l: Dosya ve dizinleri listeleterek ekrana basar. Dosya izinleri, kullanıcı ve grup bilgisini, boyutları ve son değiştirme tarihlerini de bu parametreyle kullanıcıya gösterir.

```
root@l0ser:/usr/home/derectus # ls -l  
total 44  
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc  
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login  
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf  
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases  
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc  
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile  
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc  
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme1  
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme2  
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme3  
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme4  
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test1  
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test2  
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test3
```

Görseli incelediğimizde ilk sütun bize izinler ve dosyanın tipi hakkında bilgi vermektedir. Üçüncü sütun bize dosyanın sahibini göstermektedir. Altıncı satır ise bize dosyanın son değiştirilme tarihini gösteriyor. Ayrıca bu ve diğer parametreler harmanlanarak birlikte de istenilen çıktıyı elde etmek için kullanılabilir.

```

root@l0ser:~/usr/home/derectus # ls -la
total 52
drwxr-xr-x 6 derectus derectus 512 Nov 4 01:45 .
drwxr-xr-x 3 root wheel 512 Oct 21 01:47 ..
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme1
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test3

```

-h: (human readable) Dosya izin boyutlarını Kilobayt, Megabayt, Gigabayt göze hitap edecek şekilde ekrana basar.

```

root@l0ser:~/home/derectus # ls -lh
total 44
-rw-r--r-- 1 derectus derectus 1.0K Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392B Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163B Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379B Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339B Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954B Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851B Oct 21 01:47 .shrc
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme1
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test3
root@l0ser:~/home/derectus #

```

-R: Dizinlerin eğer alt dizinleri varsa onları da listelemektedir.

```

-rw-r--r-- 1 derectus derectus 1.0K Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392B Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163B Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379B Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339B Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954B Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851B Oct 21 01:47 .shrc
drwxr-xr-x 5 root derectus 512B Nov 4 20:29 deneme1
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512B Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0B Nov 4 01:45 test3

./deneme1:
total 12
drwxr-xr-x 3 root derectus 512B Nov 4 20:29 altdizin1
drwxr-xr-x 2 root derectus 512B Nov 4 20:29 altdizin2
drwxr-xr-x 2 root derectus 512B Nov 4 20:29 altdizin3

./deneme1/altdizin1:
total 4
drwxr-xr-x 2 root derectus 512B Nov 4 20:29 yeterulandizin1
--More-- (byte 1157)

```

Görüldüğü üzere önce tüm dizinlerimizi listeledi daha sonra deneme1/ dizinin altındaki diğer dizinleri ve daha sonra deneme1/altdizin1/ altdındaki var olan tüm dizinleri gruplandırarak bize geri döndürdü.

b. Dosya ve Dizin oluşturma

Linux sistemlerde dosya oluşturma birçok şekilde gerçekleştirilebilmektedir. Eğer boş bir dosya oluşturmak istiyorsak imdadımıza hemen touch komutu yetişir. Asli işlevi dosya oluşturmaktan ziyade dosyaların erişim tarihlerini değiştirmekir.

```
# touch (parametere) (dosya_adi_1, dosya_adi_2 ...)
```

Farz edelim ki şu an bilmediğimiz/bildiğimiz herhangi bir değer dönecek bir işlem yaptık ve bu dönen değeri yeni oluşturacağımız bir dosyaya kaydetmek isteyelim.

```
root@l0ser:/home/derectus # echo "bak nasıl yazıyorum ama dosyaya" >> olmayan_dosya
a.md
root@l0ser:/home/derectus # ls -l
total 52
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc
-rw-r--r-- 1 root derectus 0 Nov 4 21:11 a.txt
drwxr-xr-x 5 root derectus 512 Nov 4 20:29 deneme1
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 30 Nov 5 01:05 olmayan_dosya.md
-rw-r--r-- 1 root derectus 14 Nov 5 01:01 sx.txt
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test3
```

Eğer bir dizin oluşturmak istiyorsak mkdir (make directory) komutunu kullanıyoruz. Kullanım şekli;

```
# mkdir (parametre) (dizin/dizinler)
```

```

root@l0ser:/home/derectus # mkdir Dosya1
root@l0ser:/home/derectus # ls -l
total 56
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc
drwxr-xr-x 2 root derectus 512 Nov 5 01:10 Dosya1
-rw-r--r-- 1 root derectus 0 Nov 4 21:11 a.txt
drwxr-xr-x 5 root derectus 512 Nov 4 20:29 deneme1
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 30 Nov 5 01:05 olmayan_dosya.md
-rw-r--r-- 1 root derectus 14 Nov 5 01:01 sx.txt
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test3

```

Bu komutu kullanmaya başladığınızda parametresiz kullanımın eğer alt dizinleri olmayan bir dosya oluşturacaksak çok efektif olacağını göreceksiniz. Eğer alt dizinleri olan bir dizin oluşturacaksak, her seferinde oluşturduğumuz dizinin içerisine gidip orada tekrar yeni bir dosya mı oluşturmak zorundayız? Hayır tabii ki bu durumda da -p parametresi imdadımıza yetişmekte.

```

root@l0ser:/home/derectus # mkdir -p Anadizin/Alt_dizin/Alt_Alt_dizin
root@l0ser:/home/derectus # ls -la
total 68
drwxr-xr-x 8 derectus derectus 512 Nov 5 01:17 .
drwxr-xr-x 3 root wheel 512 Oct 21 01:47 ..
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc
drwxr-xr-x 3 root derectus 512 Nov 5 01:17 Anadizin
drwxr-xr-x 2 root derectus 512 Nov 5 01:10 Dosya1
-rw-r--r-- 1 root derectus 0 Nov 4 21:11 a.txt
drwxr-xr-x 5 root derectus 512 Nov 4 20:29 deneme1

```

Peki eğer oluşturulma sırasında biz bu dosyanın izinlerini ayarlayamaz mıyız? Oluştururuz. Oluşturalım.

```

root@l0ser:/home/derectus # mkdir -m 777 Anadizin
root@l0ser:/home/derectus # ls -la
total 68
drwxr-xr-x 8 derectus derectus 512 Nov 5 01:21 .
drwxr-xr-x 3 root wheel 512 Oct 21 01:47 ..
-rw-r--r-- 1 derectus derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r-- 1 derectus derectus 392 Oct 21 01:47 .login
-rw-r--r-- 1 derectus derectus 163 Oct 21 01:47 .login_conf
-rw----- 1 derectus derectus 379 Oct 21 01:47 .mail_aliases
-rw-r--r-- 1 derectus derectus 339 Oct 21 01:47 .mailrc
-rw-r--r-- 1 derectus derectus 954 Oct 21 01:47 .profile
-rw-r--r-- 1 derectus derectus 851 Oct 21 01:47 .shrc
drwxrwxrwx 2 root derectus 512 Nov 5 01:21 Anadizin

```

Görüldüğü üzere -m parametresiyle de dosya ve dizinlerimizin oluşturma esnasında yetkilerini ayarlayabiliyoruz.

c. Dosya Okuma ve Düzenleme

Bir çok dosyayı terminal üzerinden okumak çoğunlukla daha efektiftir. En çok kullanılan ve bilenen komutlarımızdan biri de cat komutudur. Gel gelelim sadece dosya okumaktansa dosya yaratmak için de kullanılabilir. Cat komutuyla sistemde bulunan herhangi bir dosya okunabilir sadece metin tabanlı dosyaları okumak zorunda değiliz. Kullanım şekli aşağıdaki gibidir.

```
# cat (parametre) (dosya_adi)
```

Dosya oluşturma ile başlayalım...

```
root@l0ser:/home/derectus # cat >> cat_dosyasi.md
sa
asa
sasa
sasa
^C
root@l0ser:/home/derectus # ls
.cshrc          Anadizin      deneme4
.login          Dosya1        olmayan_dosya.md
.login_conf    a.txt         sx.txt
.mail_aliases  cat_dosyasi.md test1
.mailrc        deneme1       test2
.profile       deneme2       test3
.shrc          deneme3
root@l0ser:/home/derectus #
```

şekilde gibi kullanımda shell bizden eğer dosyanın içine bir şey yazacaksak girdi bekler. Biz dosyamıza görseldeki rastgele kelimeleri girdik. Control - D ile bunu kesebilirsiniz eğer ctrl+d kullanırsanız dosyanın sonuna end of file eklenmiş olur. Şu an dosyamız oluştu ve içine atadığımız değerler yazıldı. Şimdi cat komutunu kullanarak dosyamızı okuyalım bakalım doğru yapmış mıyız?

```
root@l0ser:/home/derectus # cat cat_dosyasi.md
sa
asa
sasa
sasa
root@l0ser:/home/derectus #
```


Evet görüldüğü üzere hem dosyamızı oluşturduk hem içine bir şeyler yazdık hem de yazdığımız dosyayı okuduk. Şimdi de oluşturduğumuz dosyanın içine bilgiler eklemek isteyelim. Bu seneryoda iki ayrı durum söz konusu. Birincisi ya dosyanın sonuna ekleme yapacağız. İkincisi ise dosyanın içindekileri silip içine yeni bir satır ekleyeceğiz.

```
root@l0ser:/home/derectus # cat > cat_dosyasi.md
Bu bir yeni satirdir.
root@l0ser:/home/derectus # cat cat_dosyasi.md
Bu bir yeni sattirdir.
```

">" ile tüm satırları silip var olan dosyamızın içerisine yeni bir satır ekledik. Eğer dosyanın sonuna eklemek isteseydik. ">>" kullanacaktık. Büyüktür ya da küçüktür işareti dosyanın adının olduğu yani data akışının olacağı yöne olmak zorundadır.

```
root@l0ser:/home/derectus # cat >> cat_dosyasi.md
Bu dosyan?n ikinci satiri.
Bu dosyanin 3. satiri.
Nuri Çilengir.
DEU
root@l0ser:/home/derectus # cat cat_dosyasi.md
Bu bir yeni sattirdir.
Bu dosyan?n ikinci sat?iri.
Bu dosyanin 3. satiri.
Nuri Çilengir.
DEU
root@l0ser:/home/derectus #
```

cat komutu o kadar işlevseldir ki sadece bunlarla sınırlı değildir yaptıkları. Örneğin elimizde 3 adet dosya var ve bunları tek bir dosyada toplamak istiyoruz. Yine imdadımıza cat komutu yetişiyor.

```
drwxrwxrwx 2 root derectus 512 Nov 5 01:21 Anadizin
drwxr-xr-x 2 root derectus 512 Nov 5 01:10 Dosya1
-rw-r--r-- 1 root derectus 0 Nov 4 21:11 a.txt
-rw-r--r-- 1 root derectus 99 Nov 5 02:00 cat_dosyasi.md
drwxr-xr-x 5 root derectus 512 Nov 4 20:29 deneme1
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme2
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme3
drwxr-xr-x 2 root derectus 512 Nov 4 01:45 deneme4
-rw-r--r-- 1 root derectus 43 Nov 5 02:04 ikincinci_dosya.txt
-rw-r--r-- 1 root derectus 30 Nov 5 01:05 olmayan_dosya.md
-rw-r--r-- 1 root derectus 14 Nov 5 01:01 sx.txt
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test1
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test2
-rw-r--r-- 1 root derectus 0 Nov 4 01:45 test3
-rw-r--r-- 1 root derectus 31 Nov 5 02:04 ucuncu_dosya
```

```

root@l0ser:/home/derectus # cat cat_dosyasi.md ikinci_dosya.txt ucuncu_dosya
> yeni_birlesmis_dosya.txt
root@l0ser:/home/derectus # ls
.cshrc          deneme2
.login          deneme3
.login_conf     deneme4
.mail_aliases   ikinci_dosya.txt
.mailrc         olmayan_dosya.md
.profile        sx.txt
.shrc           test1
Anadizin        test2
Dosya1          test3
a.txt           ucuncu_dosya
cat_dosyasi.md yeni_birlesmis_dosya.txt
deneme1

```

```

root@l0ser:/home/derectus # cat yeni_birlesmis_dosya.txt
Bu bir yeni sattirdir.
Bu dosyan?n ikinci sat?iri.
Bu dosyanin 3. satiri.
Nuri Çilengir.
DEU
deneme
deneme 2
deneme 3
deneme 4 sat?iri
test sat?iri 1
test satiri 2

```

cat komutu anlatmak ile bitmez. Bu yüzden cat komutunun manual sayfasını incelemenizi tavsiye ederim.

d. Dosya Taşıma, Kopyalama ve Silme

Kopyalama komutu cp (copy)

cp komutu bir dosya ya da dizini başka bir dosya ya da dizine kopyalamaya yarar. Kullanım şekli aşağıdaki gibidir.

```
# cp (kopyalanacak dosya dizin adı) (kopyalancak yeni dizin ya da adı)
```

Örneğin Xxx adında bir dizinimiz olsun ve /home/user_name dizinimiz altında da sample.txt diye başka bir dosyamız olsun ve sample.txt'yi Xxx dizinimize kopyalamak ve kopyalarken bunu markdown formatta kaydetmek isteyelim.

```

root@l0ser:/home/derectus # cp sample.txt Xxx
root@l0ser:/home/derectus # ls -l Xxx
total 0
-rw-r--r-- 1 root derectus 0 Nov 5 02:39 sample.txt
root@l0ser:/home/derectus #

```

cp komutuyla var olan dosya başka bir isimde de yedeklenebilir.

```

root@l0ser:/home/derectus # cp Xxx/sample.txt Xxx/yeni_isim.md
root@l0ser:/home/derectus # ls -l Xxx
total 0
-rw-r--r-- 1 root derectus 0 Nov 5 02:39 sample.txt
-rw-r--r-- 1 root derectus 0 Nov 5 02:48 yeni_isim.md
root@l0ser:/home/derectus #

```

Kopyalama komutu mv (move)

mv komutu adı üstünde dosya ve dizinlerimizi taşımamıza yarasa da diğer tüm komutlar gibi başka işlemlere de sahiptir. Örneğin mv komutuyla dosyalarınızı yeniden adlandırabilirsiniz. Kullanımı cp komutuyla aynıdır.

```
# mv (taşınacak dosya ya da dizin adı) (hedef dizin ve dosya adı)
```

Örneğin, XxX dosyasına kopyaladığımız yeni_isim.md dosyamızı bir üst dizine taşıyalım.

```
root@l0ser:/home/derectus # mv XxX/sample.txt sample.txt
root@l0ser:/home/derectus # ls
.cshrc          .login_conf    .mailrc        .shrc          sample.txt
.login         .mail_aliases  .profile       XxX
root@l0ser:/home/derectus #
```

Şimdi de taşıdığımız dosyanın adını değiştirelim...

```
root@l0ser:/home/derectus # ls
.cshrc          .login_conf    .mailrc        .shrc          yeni_isim.txt
.login         .mail_aliases  .profile       XxX
root@l0ser:/home/derectus #
```

mv komutu cp komutuna benzese de unutulmamalıdır ki move dosyaları taşır yediğini almaz. Kullanımlarının ve birçok işlevinin benzer olması sizi yanıltmasın.

Dosya Dizin Silme rm (remove) komutu

Sisteminizdeki dosyaları ve dizinleri silmenize yarar. Fakat kullanırken yetki hatalarıyla karşılaşabilirsiniz o yüzden hangi dosyayı sileceğinizi bilmeniz gerekir. Ayrıca rm komutu ile silinen herhangi bir şey çöp kutusuna taşınmaz direkt olarak silinmektedir. Kullanımı;

```
# rm (parametre) (dosya ya da dizin adı)
```

Görüldüğü üzere bir.txt iki.txt on.txt uc.md dort adlı dosyalarımız ve XxX adında bir dizinimiz bulunmakta. ilk önce bir.txt dosyasını silelim.

```
root@l0ser:/home/derectus # ls -la
total 40
drwxr-xr-x  3 derectus  derectus   512 Nov  5 03:55 .
drwxr-xr-x  3 root      wheel     512 Nov  5 03:40 ..
-rw-r--r--  1 derectus  derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r--  1 derectus  derectus  392 Oct 21 01:47 .login
-rw-r--r--  1 derectus  derectus  163 Oct 21 01:47 .login_conf
-rw-----  1 derectus  derectus  379 Oct 21 01:47 .mail_aliases
-rw-r--r--  1 derectus  derectus  339 Oct 21 01:47 .mailrc
-rw-r--r--  1 derectus  derectus  954 Oct 21 01:47 .profile
-rw-r--r--  1 derectus  derectus  851 Oct 21 01:47 .shrc
drwxr-xr-x  2 root      derectus   512 Nov  5 03:44 XxX
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 bir.txt
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 dort
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 iki.txt
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 uc.md
-rw-r--r--  1 root      derectus    0 Nov  5 02:39 yeni_isim.txt
```

```

root@l0ser:/home/derectus # rm bir.txt
root@l0ser:/home/derectus # ls -la
total 40
drwxr-xr-x  3 derectus  derectus   512 Nov  5 04:00 .
drwxr-xr-x  3 root      wheel     512 Nov  5 03:40 ..
-rw-r--r--  1 derectus  derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r--  1 derectus  derectus  392 Oct 21 01:47 .login
-rw-r--r--  1 derectus  derectus  163 Oct 21 01:47 .login_conf
-rw-----  1 derectus  derectus  379 Oct 21 01:47 .mail_aliases
-rw-r--r--  1 derectus  derectus  339 Oct 21 01:47 .mailrc
-rw-r--r--  1 derectus  derectus  954 Oct 21 01:47 .profile
-rw-r--r--  1 derectus  derectus  851 Oct 21 01:47 .shrc
drwxr-xr-x  2 root      derectus  512 Nov  5 03:44 XxX
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 dort
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 iki.txt
-rw-r--r--  1 root      derectus    0 Nov  5 04:00 on.txt
-rw-r--r--  1 root      derectus    0 Nov  5 03:55 uc.md
-rw-r--r--  1 root      derectus    0 Nov  5 02:39 yeni_isim.txt

```

Şimdi de XxX dizinini silmeyi deneyelim.

```

root@l0ser:/home/derectus # rm XxX/
rm: XxX/: is a directory

```

Evet hata verdi “XxX bir dizin”miş :) Eğer eğer silmek istediğimiz dizinlerin alt klasörleri ya da dosyaları varsa -r parametresini kullanmamız gerekir. -r recursive’in kısaltmasıdır.

```

root@l0ser:/home/derectus # rm -r XxX/
root@l0ser:/home/derectus # ls
.cshrc          .mail_aliases  .shrc          on.txt
.login         .mailrc       dort           uc.md
.login_conf    .profile      iki.txt        yeni_isim.txt
root@l0ser:/home/derectus #

```

Peki her dosyayı tek tek silmek ile mi uğraşacağız? Bazı durumlarda hayır. Örneğin şu an dizinimizde bulunan .txt uzantılı tüm dosyaları silelim.

```

root@l0ser:/home/derectus # rm *.txt
root@l0ser:/home/derectus # ls
.cshrc          .login_conf    .mailrc       .shrc          uc.md
.login         .mail_aliases  .profile      dort
root@l0ser:/home/derectus #

```

Şimdi ki seneryomuzda da 4 adet test1.sh test2.sh test3.sh test4.sh adında dosyaların olduğunu farz edelim.

```

root@l0ser:/home/derectus # ls
.cshrc          .mail_aliases  .shrc          test2.sh       uc.md
.login         .mailrc       dort           test3.sh
.login_conf    .profile      test1.sh      test4.sh
root@l0ser:/home/derectus #

```

Birden üç’e kadar olanları silmek istiyoruz ama test4.sh bizim için önemli.

```
root@l0ser:/home/dereectus # rm test[123].sh
root@l0ser:/home/dereectus # ls
.cshrc          .login_conf    .mailrc        .shrc          test4.sh
.login         .mail_aliases  .profile       dort           uc.md
root@l0ser:/home/dereectus #
```

Görüldüğü üzere istediğiniz regular expressiona göre istediğiniz işlemi yapabilmektesiniz. Aslında bu şu ana kadarki tüm komutlar için geçerlidir.

e. Dosya ve Dizin arama

Sistemimiz üzerinde bir dosyayı veya mevcut bir dosya içinde bölüm aramak için kullanabileceğimiz find, grep, sed gibi komutlar bulunmaktadır.

find Komutu : yüzlerce binlerce dosya ve dizinin bulunduğu durumlarda ls komutu dosya aramak için pek işlevsel olmamaktadır. find komutu burada imdadımıza yetişmekte.

```
# find (Aramaya başlanacak dizin) (parametreler) (dosya_adi)
```

Örnek senaryolarla komutumuzu tanımaya başlayalım...

Örneğin /etc dizininde bir konfigürasyon dosyanın varlığını sorgulamamız gerekli. /etc dizini sisteminizdeki konfigürasyon dosyalarının bulunduğu dizindir. dolayısıyla çok fazla dosya ve dizin içerir. İlk örnek için /etc altında hosts dosyasını arayalım.

```
root@l0ser:/etc # find /etc -name hosts
/etc/bluetooth/hosts
/etc/hosts
root@l0ser:/etc #
```

/etc dizini altında ve /etc/bluetooth/ dizini altında iki ayrı host dosyamız bulundu. Burada dikkatinizi çekmek isterim ki name adında bir parametre atayarak isme göre sorgulama yaptık. Spesifik bir dosyadan ziyade boyuta göre de arama yapabiliriz. Ee o halde yapalım.

```
root@l0ser:/ # find /bin/ -size +50kb
/bin/rmail
/bin/red
/bin/csh
/bin/ps
/bin/pax
/bin/sh
/bin/ed
/bin/tcsh
root@l0ser:/ # █
```

Burada 50kb'den büyük dosyaları listelettik. Şimdi ise hem dosya tipine hem dosya boyutuna hem de dosya yetkisine göre bir arama yapalım.

```
root@l0ser:/ # find / -size +50kb ! -perm 653 -type d
/usr/local/lib/python3.6/test/__pycache__
/usr/src/sys/gnu/dts/arm
/usr/share/man/man3
/usr/share/man/man9
/usr/share/openssl/man/man3
/usr/ports/www
/usr/ports/devel
/var/db/portsnap/files
root@l0ser:/ # █
```

Son olarak sistemimizdeki boş dosya ve dizinleri arayalım ve find komutunu tamamlayalım.

```
root@l0ser:/ # find / -type f -type d -empty
root@l0ser:/ # █
```

Benim sistemimde yokmuş :) Find komutunu kesinlikle manual sayfasından incelemelisiniz.

grep komutu : Şimdiye kadar bir dosya ya da dizini birçok şekilde aramayı gördük. Peki ya dosyanın içeriğine göre bir arama yapmak istersek? Grep komutu bize bu konuda oldukça büyük destek sağlamakta. Kullanım şekli:

```
# grep (parametre) (aranan) (dosya/dosyalar)
```

etc/hosts dosyamızın default içeriği aşağıdaki gibidir. Seneryolarımızı bu dosya üzerinden yürütelim.

```

#
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1          localhost localhost.my.domain
127.0.0.1    localhost localhost.my.domain
#
# Imaginary network.
#10.0.0.2    myname.my.domain myname
#10.0.0.3    myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#       10.0.0.0      -   10.255.255.255
#       172.16.0.0   -   172.31.255.255
#       192.168.0.0  -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers. Do not try to invent your own network
# numbers but instead get one from your network provider (if any) or
# from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfrinIC.)
#

```

İlk önce dosyanın içerisinde geçen bir kelime veya kelime grubunu aralayalım.

```

root@l0ser:/ # grep Internet /etc/hosts
# private nets which will never be connected to the Internet:
# In case you want to be able to connect to the Internet, you need
root@l0ser:/ # █

```

Görüldüğü üzere Internet kelimesinin geçtiği iki adet satır getirdi. Şimdi de önünde ve sonunda boşluk karakterinin olduğu ve RIPE kelimesinin kullandığı satırları listeleyelim.

```

root@l0ser:/ # grep -iw RIPE /etc/hosts
# from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfrinIC.)
root@l0ser:/ # █

```

Bu sefer ise aradığımız anahtar kelimelerin dosyada kaç kere kullanıldığını görelim.

```

root@l0ser:/ # grep -iwc RIPE /etc/hosts
1
root@l0ser:/ # grep -iwc Internet /etc/hosts
2

```

Grep komutuyla yapılabileceklerin gerçekten bir sınırı yoktur. Bu yüzden kesinlikle vakit ayırıp bakmanızı öneririm.

f. Dosyalara Link Oluşturma

Linking(bağlama), bir dosyayı başka bir dosyaya(link) bağlamaya denilebilir. Hardlink oluşturulan bağlantının aynı dosyaymış gibi davranmasına sebep olurken sembolik link ise aynı dosyayı gösteren bir gösterici olarak düşünülebilir.

Hard Link

Hard Link bir dosyaya alternatif isimli bir dosya oluşturmak istenildiğinde kullanılır.

Oluşturulan hard link orjinal dosya ile aynı inode numarasına sahip olacaktır. Bu sebeple birisine yapılan değişiklik iki dosyaya da etki edecektir. Bir dosyaya istenilen sayıda hard link oluşturulabilir. Aynı zamanda bir hard link'e başka bir hard link oluşturulabilir. Ancak hard link oluşturamayacağımız durumlar vardır. Bunlardan bazıları:

- Dizinlere
- Dosya Sistemlerine
- Partisyonlara hardlink oluşturulamaz.

/ dizinimiz altında sample.md adında bir dosyamız var.

```

-r----- 1 root wheel 16777216 Oct 21 01:11 .sujournal
-r--r--r-- 1 root wheel 6177 Dec 7 2018 COPYRIGHT
drwxr-xr-x 2 root wheel 1024 Dec 7 2018 bin
drwxr-xr-x 9 root wheel 1536 Oct 21 02:55 boot
dr-xr-xr-x 8 root wheel 512 Nov 4 20:25 dev
-rw----- 1 root wheel 4096 Nov 4 20:25 entropy
drwxr-xr-x 25 root wheel 2560 Oct 21 01:47 etc
lrwxr-xr-x 1 root wheel 8 Oct 21 01:16 home -> usr/home
drwxr-xr-x 4 root wheel 1536 Dec 7 2018 lib
drwxr-xr-x 3 root wheel 512 Oct 21 01:12 libexec
drwxr-xr-x 2 root wheel 512 Dec 7 2018 media
drwxr-xr-x 2 root wheel 512 Dec 7 2018 mnt
drwxr-xr-x 2 root wheel 512 Dec 7 2018 net
dr-xr-xr-x 2 root wheel 512 Dec 7 2018 proc
drwxr-xr-x 2 root wheel 2560 Dec 7 2018 rescue
drwxr-xr-x 2 root wheel 512 Oct 20 18:52 root
-rw-r--r-- 1 root wheel 0 Nov 5 05:29 sample.md
drwxr-xr-x 2 root wheel 2560 Dec 7 2018 sbin
lrwxr-xr-x 1 root wheel 11 Dec 7 2018 sys -> usr/src/sys
drwxrwxrwt 7 root wheel 1024 Nov 5 03:38 tmp
drwxr-xr-x 17 root wheel 1024 Oct 21 01:16 usr
drwxr-xr-x 25 root wheel 1024 Nov 4 20:25 var

```

Bu dosyayı /etc altında linkleyelim.

```

root@l0ser:/ # ln sample.md /etc/sample_link.md
root@l0ser:/ # ls /etc/sample_link.md
/etc/sample_link.md
root@l0ser:/ # ls -la /etc/sample_link.md
-rw-r--r-- 2 root wheel 0 Nov 5 05:29 /etc/sample_link.md
root@l0ser:/ #

```

Bu dosyalardan herhangi biri silinse de artık diğerine bir şey olmayacak hatta yapılan değişiklikler ikisine uygulanacaktır.

Soft Link

Sembolik linkler daha kullanışlı olabilir. Hard linkteki gibi kısıtlamaları yoktur. Dizinlere, diğer dosya sistemlerine ve diğer partiyonlara sembolik link oluşturulabilir. Aslında windows ortamındaki kısayollar gibi düşünülebilir. Sembolik linkler bir dosyanın konumunu işaret ederler. Bir sembolik link, hard link gibi dosyanın diskteki fiziksel konumunu işaret etmek yerine dosyanın yolunu(path) işaret ederler.


```
# ln -s <dosya-ya-da-dizin> <sembolik-link-adi>
```

Yukarıdaki seneryonun aynısı bu sefer usr/home/derectus klasörüne uygulayalım. Az önce /user/home/derectus dizinine hardlink oluşturamazdık çünkü bizim partitasyon sistemimiz buna izin vermeyecekti /usr ve / dizini farklı partisyonlarda bulunuyor. Fakat şimdi bunu yapabileme imkanına sahibiz.

```
root@l0ser:/ # ln -s /sample.md /home/derectus/deneme_link.md
root@l0ser:/ # ls -la /home/derectus/
total 36
drwxr-xr-x  2 derectus  derectus   512 Nov  5 05:50 .
drwxr-xr-x  3 root     wheel     512 Nov  5 03:40 ..
-rw-r--r--  1 derectus  derectus 1054 Oct 21 01:47 .cshrc
-rw-r--r--  1 derectus  derectus  392 Oct 21 01:47 .login
-rw-r--r--  1 derectus  derectus  163 Oct 21 01:47 .login_conf
-rw-----  1 derectus  derectus  379 Oct 21 01:47 .mail_aliases
-rw-r--r--  1 derectus  derectus  339 Oct 21 01:47 .mailrc
-rw-r--r--  1 derectus  derectus  954 Oct 21 01:47 .profile
-rw-r--r--  1 derectus  derectus  851 Oct 21 01:47 .shrc
lrwxr-xr-x  1 root     derectus   10 Nov  5 05:50 deneme_link.md -> /sample.md
-rw-r--r--  1 root     derectus    0 Nov  5 03:55 dort
-rw-r--r--  1 root     derectus    0 Nov  5 04:11 test4.sh
-rw-r--r--  1 root     derectus    0 Nov  5 03:55 uc.md
```

5. Process Yönetimi

Sistemde çalışan her bir komut ve program bir process'dir . Her işlem benzersiz bir işlem kimliği (PID) adı verilen bir numara ile tanımlanır. Dosyalara benzer şekilde, her işlemin bir sahibi ve grubu vardır. Process'in sahibi ve grubu hangi dosya veya cihazlara erişimine yetki olduğunu hesaplamakta kullanılır. Sistemde processleri görüntülemek için **ps** ve **top** komutları bulunur.

ps : komutu o anda çalışan processlerin istatistiklerini, PID numaralarını, bellek kullanımını, çalıştığı dizinleri vb. görüntüler.

Kullanım: parametrelere ulaşmak ps -h ve man ps sayfalarından ulaşılabilir. Bana göre en efektif kullanım şekli aşağıdaki gibidir.

```
root@l0ser:~ # ps auxw | more
```

Çıktı :

```
USER      PID    %CPU %MEM    USZ    RSS TT      STAT  STARTED      TIME COMMAND
root      11  100.0  0.0      0      16 -      RNL   18:50    184:20.11 [idle]
root      0    0.0  0.0      0     224 -      DLs   18:50    0:12.71 [kernel]
root      1    0.0  0.0    9952   1016 -      ILs   18:50    0:00.05 /sbin/init --
root      2    0.0  0.0      0      16 -      DL    18:50    0:00.00 [crypto]
root      3    0.0  0.0      0      16 -      DL    18:50    0:00.00 [crypto returns 0]
root      4    0.0  0.0      0      32 -      DL    18:50    0:00.45 [cam]
root      5    0.0  0.0      0      16 -      DL    18:50    0:00.00 [sctp_iterator]
root      6    0.0  0.0      0      16 -      DL    18:50    0:01.21 [rand_harvestq]
root      7    0.0  0.0      0      16 -      DL    18:50    0:00.00 [soaiod1]
root      8    0.0  0.0      0      16 -      DL    18:50    0:00.00 [soaiod2]
root      9    0.0  0.0      0      16 -      DL    18:50    0:00.00 [soaiod3]
root     10    0.0  0.0      0      16 -      DL    18:50    0:00.00 [audit]
root     12    0.0  0.0      0     224 -      WL    18:50    0:10.08 [intr]
root     13    0.0  0.0      0      48 -      DL    18:50    0:00.13 [geom]
root     14    0.0  0.0      0      16 -      DL    18:50    0:00.00 [sequencer 00]
root     15    0.0  0.0      0     160 -      DL    18:50    0:00.96 [usb]
root     16    0.0  0.0      0      16 -      DL    18:50    0:00.00 [soaiod4]
root     17    0.0  0.0      0      48 -      DL    18:50    0:01.39 [pagedaemon]
root     18    0.0  0.0      0      16 -      DL    18:50    0:00.00 [vmdaemon]
root     19    0.0  0.0      0     112 -      DL    18:50    0:01.70 [bufdaemon]
root     20    0.0  0.0      0      16 -      DL    18:50    0:00.05 [vnrlu]
root     21    0.0  0.0      0      16 -      DL    18:50    0:00.28 [syncer]
root     377    0.0  0.1   11352  2592 -      Is    18:50    0:00.03 dhclient: system.sys
--More-- (byte 1700)
```

top, komutu ise bütün processleri görüntüler ve sürekli olarak bu görüntüyü yeniler. Böylece sistemde çalışan prosesler interaktif olarak takip edilebilir.

```
last pid: 1115; load averages: 0.33, 0.41, 0.48 up 0+03:07:25 21:57:37
20 processes: 1 running, 19 sleeping
CPU: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
Mem: 2304K Active, 11M Inact, 62M Wired, 20M Buf, 2880M Free
Swap: 4096M Total, 4096M Free

  PID USERNAME      THR PRI NICE   SIZE    RES STATE   TIME    WCPU COMMAND
 1115 root              1  20   0    13M    3508K RUN      0:00    0.04% top
  435 root              1  20   0     10M    1432K select  0:01    0.01% devd
  506 root              1  20   0     11M    2640K select  0:01    0.01% syslogd
  702 root              1  20   0     17M    6932K select  0:00    0.00% sendmail
  781 root              1  20   0     13M    3888K pause   0:00    0.00% csh
  709 root              1  20   0     11M    2608K nanslp  0:00    0.00% cron
  380 root              1  52   0     11M    2724K select  0:00    0.00% dhclient
  726 root              1  20   0     11M    2436K select  0:00    0.00% moused
  773 root              1  21   0     12M    3100K wait    0:00    0.00% login
  377 root              1  20   0     11M    2592K select  0:00    0.00% dhclient
  434 _dhcp            1  20   0     11M    2856K select  0:00    0.00% dhclient
  777 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
  778 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
  780 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
  705 smmsp           1  20   0     16M    6620K pause   0:00    0.00% sendmail
  775 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
  774 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
  776 root              1  52   0     11M    2264K ttyin   0:00    0.00% getty
```

4.1 Process sonlandırma

ps ya da top ile görüntülenen prosesler **kill** komutuyla sonlandırılabilir.

```
# kill 3551 96475 // process numaraları 3551 ve 96475 olan prosesler ölür.
```

Root dışında her kullanıcı sadece kendi processlerini sonlandırabilir.

kill komutu parametre olarak sinyal numarası alabilir (Kill bir interrupt oluşturur. Sistem referanslarını kullanarak kendi kill komutlarınızı oluşturabilirsiniz)

1	HUP (hang up)
2	INT (interrupt)
3	QUIT (quit)
6	ABRT (abort)
9	KILL (non-catchable, non-ignorable kill)
14	ALRM (alarm clock)
15	TERM (software terminator signal)

5. Dizin yapısı

FreeBSD dizini yapısı, sistemi genel olarak anlamak için esastır. En önemli dizin kök veya "/" dir. Bu dizin, önyükleme sırasında takılan ilk dizindir ve işletim sistemini çok kullanıcı çalışmaya hazırlamak için gereken temel sistemi içerir. Kök dizin ayrıca çok kullanıcı çalışmaya geçişte bağlanan diğer dosya sistemleri için bağlama noktaları içerir. FreeBSD dosya sistemi olarak **ufs** (Unix File System) kullanır. Aynı zamanda ext2,ext3,ext4, nfs, ntfs vb gibi diğer dosya sistemlerini de desteklemektedir.

Directory	Description
/	Dosya sisteminin kök dizini.
/bin	Temel kullanıcı programlarını ve komutlarını içerir.
/boot	İşletim sistemi önyüklemesi sırasında kullanılan programlar ve yapılandırma dosyaları.
/dev	FreeBSD aygıtlarını içerir.
/etc	Sistem yapılandırma dosyaları ve komut dosyaları.
/mnt	Genellikle sistem yöneticileri tarafından geçici bir bağlama noktası olarak kullanılan boş dizin.
/proc	İşlem dosya sistemi.
/rescue	Acil durumlarda kurtarma için statik olarak bağlı programların bulunduğu dizin.
/root	root kullanıcının ana dizinidir.
/sbin	Genellikle sistem programlarını içerir.
/tmp	Genellikle sistem genelinde korunmayan geçici dosyalar sistem yeniden başlatılışına kadar tutulduğu ve işlendiği dizin.
/usr	Kullanıcılarla ilgili programları ve dosyalarını içerir.
/usr/ports	FreeBSD port ağacının bulunduğu dizin.
/var	Çeşitli log dosyaları ve spooling dizinini içerir.