

Linux Komut Yapısı

Unix ve benzeri işletim sistemlerinde kullanıcının komut yazmasını sağlayan bu komutları yorumlayarak gerekli işlemleri yapan programlara kabuk(shell) adı verilir. Unix’de bir kullanıcı bir dizi kabuktan istediğini seçebilir. Kullandığınız kabuk ne olursa olsun, gerek kabuktan kaynaklanan, gerekse Unix komutlarının hepsinin uyduğu bazı standartlardan kaynaklanan bazı geleneksel yapılar vardır. Bunlara bilmeniz ilk kez duyduğunuz bir komutun kullanımını bile kolayca çıkarabilmeniz yanı sıra, bir dizi genel hatadan kaçınmanızı da sağlayacaktır.

Unix’te (ve Linux’te) bütün komutlar ve dosya isimlerinde büyük/küçük harf ayrımı önemlidir. Sistem komutlarının ve dosyaların çoğu küçük harfle yazılır. Komut ve dosya adlarında kullanacağınız bazı karakterlerin gerek dosya ve dizin yapısı, gerekse kabuk ve diğer komutlar nedeniyle bazı özel anlamları vardır. Örneğin “/” karakteri hiçbir dosya adına bulunamaz(dosya ve dizinler için ayraç olarak kullanıldığından). “-“ ile başlayan bir dosya oluşturulabilir olsa da silmeye kalktığında yeni bir Unix kullanıcısının başına dert açabilir. Unix’de komutlara seçenek verirken, seçenektan önce “-“ karakterleri kullanılır.

```
$ ls -l
```

Unix komutları tersi istenmedikçe girdilerini standart girdiden(klavve) alır, çıktılarını standart çıktıya(ekran) yazar. Bu özellik yönlendirme ve boru(pipe) operatörleri ile birlikte komut satırından birçok işlemi kolayca yapmanızı sağlar. Unix kabukları komut satırından verilen komutu çalıştırmadan önce bir dizi karakterleri yorumlayarak dosya adlarını çevirirler. Bu karakterler:

* (Yıldız Karakteri) sıfır dahil herhangi bir sayıda karakter yerine geçer.

Örneğin **rm *** komutu bütün dosyaları siler, **ls -la*** komutu “a” ile başlayan dosyaların listesini verir.

? (Soru İşareti) tek bir karakter yerine geçer.

Örneğin ?? adı iki karakterden oluşan bütün dosyalar anlamına gelir.

[] (Köşeli Parantez) karakterleri arasında yazılan liste içindeki herhangi bir harfe dönüştürülür.

Örneğin **cp *[abc] /tmp**

komutu “a”, “b” ya da “c” ile biten bütün dosyaları /tmp dizinine kopyalayacaktır. Liste içinde aralarına “-“ işareti koyarak aralıklar verebilirsiniz.

Örneğin **[a-z]*** büyük harfle başlayan bütün dosyalar anlamına gelir.

Dosya isimlerini koyarken ciddi bir kısıtlama ile karşılaşmazsınız. Aşağıdakiler kullanılabilir dosya isimlerine birkaç örnektir:

1. uzun+bir=dosyası_adı-değilmi
2. aslındadosya isimleriikiyüzellibeşkarakterekadaruzayabilir
3. ya da içinde boşluk karakteri bulunabilir
4. başka [hangi_karakterler] iki:nokta:üstüste
5. ya.da.bol.noktalı

Dosya adının başında ya da sonunda boşluk karakteri bulunabilir, ancak iş silmeye gelince bu dosya başınızı ağrıtabilir. Siz en iyisi dosya isimlerinin içinde boşluk karakterleri kullanmayın.

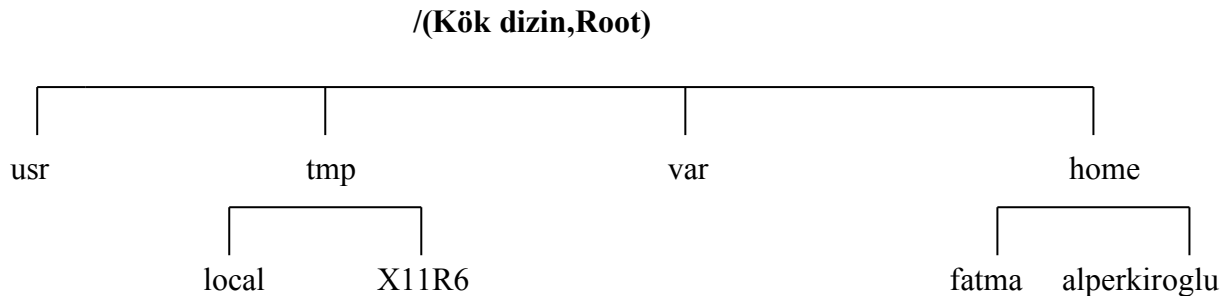
[Kaynak 1]

Dosya ve Dizin Yapısı

Unix altında bazı karakterlerin özel anlamları vardır. Dizin ve dosya isimlerinin başında nokta olması durumunda bu dosyalar gizli dosya haline gelir ve parametresiz yazılan **ls** Komutuyla görünmez. Dosya ve dizin isimleri 255 karakteri aşamazlar. Sisteme girince önceden tanımlanmış bir dizin altında bulunursunuz. Bu dizin normal kullanıcılar için genellikle /home ve ardından gelen kullanıcı dizinin ismidir. Bulunduğunuz dizinin ismini görmek için **pwd**(print working directory) yazın.

```
$ pwd
/home/alperkiroglu
$ cd /etc
$ pwd
/etc
```

Unix komut yapısı DOS'a çok benzer. Dizin değiştirmek için **cd**, dizin yaratmak için **mkdir** komutları sıkça kullanılır.



Hiyerarşik bir sıraya sahip olan Unix'te en üstte / dizini (kök dizin) yer alır. Sistemdeki tüm diğer dosya ve dizinler bunun altında toplanırlar.

~ işareti, kullanıcının ev dizinini gösterir. Aşağıdaki komutların yardımıyla ev dizinine geçin ve **mkdir** komutu ile benim isimli bir dizin yaratın. Dizin yolunu, bir başka deyişle kök dizininden itibaren ismini vererek de yaratabilirsiniz.

```
$ cd ~
$ pwd
/home/alperkiroglu
$ mkdir /home/alperkiroglu/benim
```

[Kaynak 1]

KOMUTLAR

pwd Komutu :

Kullanıcı dizinler üzerinde hareket ederken zaman zaman hangi dizin üzerinde çalıştığını öğrenmek isteyebilir.Bu durumda "pwd" kullanılır.Her hangi bir ilave parametre kullanılmaz.

Örnekler :

1.

```
$ pwd
/home/alperkiroglu
```
2.

```
[kiroglu@localhost alperkiroglu]$ pwd
/home/alperkiroglu
```
3. cd komutu kullanılarak bir üst dizine geçildiğinde;

```
[kiroglu@localhost alperkiroglu]$ cd..
[kiroglu@localhost alperkiroglu]$ pwd
/home
[kiroglu@localhost alperkiroglu]$
```

[<http://atlas.cc.itu.edu.tr/~coskunu/>]

ls Komutu :

Bir dizin içinde başka alt dizinler yanı sıra ,dosyaları listelemek amacıyla kullanılır.

Söz Dizimi:

`ls[seçenekler][dosya veya dizin...]`

Seçenekler :

- a : (.) ve (..) ile başlayan dosyalar dahil,dizini listele.
- A : (.) ve (..) ile başlayan dosyalar hariç,dizini listele.
- e : Her dosyanın modunu, bağlantı sayısını, grubunu, boyunu (byte olarak), son değişiklik tarihini, adını listele.
- f : Her bir argümanın bir dizin olarak yorumlanmasına neden olur.
- F : Dosya isimleri sonuna (*), dizin isimleri sonuna (/)koy.
- g : Tüm bilgiler ile dosya sahibinin grup adını listele.
- l : Her dosyanın modunu, bağlantı sayısını, grubunu, boyunu (byte olarak), son değişiklik sayısını listele.
- m : Dosya isimlerini virgülle ayırarak listele.
- p : Dizinlerin sonuna (/) koy.
- r : Sıralamayı sondan başa doğru yap.
- R : Belirlenen bir dizin içindeki dosyaların yanı sıra, varsa tüm alt dizinleri içerikleriyle birlikte listele.
- s : Blok cinsinden dosyaların boyutunu kilobyte olarak listele.
- t : Dosyaları değişime uğrama zamanına göre listele.
- u : Dosyaları en son erişim zamanlarına göre listele.

Örnekler :

1. Dizindeki tüm dosyaları listelemek için:

```
$ls -a
$
```

2. Detaylı bilgi almak için:

```
$ls -l setup.log
$
```

3. Örneğin prog isimli dizindeki:

Dosyaları listelemek için:

```
$ls/usr/muh/prog
muhpg007
muhpg009
muhpg018
muhpgelk
muhpggun
muhpgs08
muhdata1
muhdata2
$
```

Tüm dosyaları ayrıntılı listelemek için :

```
$ls-la
total 8
-rwxr--r-- 1 alperkiroglu kioglu 33104 Mar  9 15:10 muhpg007
-rwxr--r-- 1 alperkiroglu kioglu 25001 Mar  9 15:10 muhpg009
-rwxr--r-- 1 alperkiroglu kioglu 30822 Mar  9 15:10 muhpg018
-rwxr--r-- 1 alperkiroglu kioglu 10888 Mar  9 15:10 muhpgelk
-rwxr--r-- 1 alperkiroglu kioglu  1860 Mar  9 15:10 muhpggun
-rwxr--r-- 1 alperkiroglu kioglu  1850 Mar  9 15:10 muhpgs08
drwxrw---- 1 muhasebe usr    656 Mar  9 15:10 muhdata1
drwxrw---- 1 muhasebe usr    302 Mar  9 15:10 muhdata2
$
```

Bu dosyalar içinden muhpg007 dosyasını ele alalım:

-rwxr--r-- :

ilk karakter : d ise bir dizindir,- ise sıradan bir dosyadır.

sonraki 3 karakter: Dosya sahibinin kendine verdiği izin.

sonraki 3 karakter: Grubun sahip olduğu izin.

sonraki 3 karakter: Kullanıcının sahip olduğu izin.

r : Okuma izni
w : Yazma izni
x : Çalıştırma izni
- : İzin yok

muhtpg007 için:

ilk karakter : - (sıradan bir dosya)
sonraki 3 karakter : rwx (dosya sahibi okuma,yazma,çalıřtırma iznine sahip)
sonraki 3 karakter : r-- (grup sadece okuma iznine sahip)
sonraki 3 karakter : r-- (kullanıcı sadece okuma iznine sahip)

1 : Bu dosya için geçerli bağlantı(link) sayısı
alperkiroglu : Kullanıcının bağlantı ismi(login name)
kiroglu : Kullanıcının dahil olduđu grubun ismi
33104 : Dosyanın byte cinsinden boyu
Mar 9 15:10 : Dosyanın en son değıştirildiđi tarih
muhtpg007 : Dosyanın adi

Bir dizin içinde yer alan dosya ve alt dizinleri listelemek için:

```
$ls -f  
muhtpg007* muhtpg009* muhtpg018* muhtpg026* muhtpgc13* muhtpgelk*  
muhtpggun* muhtpgs08* muhtpgs09* muhtdata1/ muhtdata2/  
$
```

dosya isimlerinin sonuna * , dizinlerin sonuna ise / işareti yerleştirilmiştir.

[<http://atlas.cc.itu.edu.tr/~coskunu/>]

cp Komutu:

cp copy demektir, yani dosya kopyalama komutudur.

Seçenekler:

- i : Bu parametre herhangi bir dosyanın üzerine yazıyorsanız sizden onay istemesini sağlar. Çođu linux sisteminde zaten bu parametre default olarak açıktır.Genelde kullanmanıza gerek yoktur.
- l : Bu önemli bir parametredir. Bununla dosya kopyalanmaz sadece hedef dizininde bir link yaratılır.
- r : Çok kullanılan bir parametredir.Çođu komutta aynı işi ifade eder. Bu parametre cp komutuna alt dizinleri de kopyalamasını sağlar.
- R : Sadece altdizinleri kopyalar.
- u : Hedef dizinde, kopyalanacak dosyaların aynısı ya da yeni tarihli yoksa kopyalar.
- v : Yapılan işlemleri gösterir.

Örnekler:

1. \$cp install.txt /user/alper/

Bu komut davul.tgz dosyasını user dizini altındaki alper dizinine kopyalar. Kopyalayacağınız dosyadan sonra yazdığınız dizin adı "/" ile başlamazsa o dizinin dosyanın altındaki bir altdizine kopyalayacağınız manasına gelir.

2 . Dosyanın bulunduğu dizinin altında "setup" diye bir dizin olsun.Dosyayı bu dizine kopyalamak için;

```
$cp install.txt setup/ yazmanız yeterlidir.
```

3 .

```
$cp prog.c prog.bak
```

prog.c adlı dosyayı prog.bak adlı dosya olarak kopyalar. Eğer prog.bak dosyası yoksa yenisi yaratılır.Eğer zaten mevcutsa prog.c'nin bir kopyası ile değiştirilir.

4 .

```
$cp *pg* /users
```

İsimleri içinde pg harf çifti yer alan dosyaları users dizinin altına kopyalar.

5.

```
$cp muh001 muh002 muh009 /users
```

muh001 muh002 muh009 isimli dosyaları users dizinine kopyalar.

6 .

```
$cp muh?? /users
```

İsimleri 5 karakterli ve isimlerinin ilk 3 karakteri muh olan dosyaları users dizinine kopyalar.

[Kaynak 12]

mv Komutu:

Bir dosyanın bir dizinden başka bir dizine taşınması gerekebilir. Böyle durumlarda mv komutu kullanılır. Bu komut yardımıyla bir dosyanın ismini de değiştirmek mümkündür.

Söz Dizilimi:

```
mv [-i] [-f] dosya1 [dosya2] dosya veya dizin
```

Seçenekler:

-f: mv komutu kullanılırken, dosyanın hareket ettirileceği yeni dosya veya dizine yazma izni olmalıdır. Eğer bir sınırlama söz konusu ise, izin kodu görüntülenecek ve yanıt isteyecektir. Şayet y olarak yanıtlanırsa mv komutu çalışır. Soru sorulması istenilmiyorsa **-f** parametresi tercih edilir.

-i: Bu parametre kullanılmazsa aktif dizinde ismi aynı olan dosya üzerine yazma işlemi yapar. Eğer kullanılırsa kullanıcıyı uyarır.

mv komutunun yaptığı işlem cp'ye benzer. Aralarındaki fark, cp ile dosyanın bir kopyasının yaratılmasına rağmen, cp'de kopyası oluşturulur, fakat orijinali silinir.

Örnekler:

1. Bir dizindeki dosyaların başka bir dizine aktarılması

```
../usr/muh/...
```

Kullanıcı şu anda aktif olan usr dizininde bulunsun.

```
$ mv muh/muh1/* muh/muh2/*
```

Yukarıdaki komut dizilimi sayesinde muh1 dizinindeki tüm dosyalar muh2 dizinine aktarılır.

2. Dosya ismi değiştirmek için

Bir dizinin içinde aşağıdaki dosyaların mevcut olduğunu varsayalım.

```
$ ls
total 3
drwxr-xr-x 1 alperkiroglu kiroglu 512 Dec 24 18:35 uygula1.html
drwxr-xr-x 1 alperkiroglu kiroglu 518 Dec 24 18:35 uygula2.html
drwxr-xr-x 1 alperkiroglu kiroglu 523 Dec 24 18:35 uygula3.html
$_
$ mv uygula* deneme*
$ ls
total 3
drwxr-xr-x 1 alperkiroglu kiroglu 512 Dec 24 18:35 deneme1.html
drwxr-xr-x 1 alperkiroglu kiroglu 518 Dec 24 18:35 deneme2.html
drwxr-xr-x 1 alperkiroglu kiroglu 523 Dec 24 18:35 deneme3.html
$_
```

Yukarıda da görüldüğü gibi uygula adlı bütün dosyalar deneme adıyla değiştirildi.

[Kaynak 12]

mkdir Komutu

Linux işletim sisteminin önemli bileşenlerinden biri olan dizinler, istenildiğinde yaratılabilir. Bu amaçla mkdir komutu kullanılır. mkdir komutu ile dosya yaratılmaz.

Söz Dizilimi:

```
mkdir [-m Mode] [-p] dizin adı
```

mkdir komutu dizin parametresi ile belirtilmiş, bir ya da daha fazla yeni dizin oluşturur. Bu komut kullanıldığında yaratılan dizinler 777 izin koduna sahip olurlar. Kullanıcı "umask" komutunu kullanmışsa, izinler istediği biçimde oluşur. Linux'de bir dizin veya dosya yaratılırken, otomatik olarak sistem tarafından izinleri ayarlanır. Bu izinler dosyalar için rw_rw_rw şeklindedir. Octal karşılığı ise 666'dır. Dizinler için, rwxrwxrwx şeklindedir. Tüm kullanıcılara söz konusu dizinlerde okuma, yazma ve çalıştırma izni veren bu durumun oktal karşılığı 777'dir. Kullanıcı isterse yukarıdaki sistem kabullerini değiştirebilir. Bu amaçla bir maske oluşturur. Her yeni dosya ve dizinin izinlerini bu maskeye göre ayarlar. Bu maskeleme işlemi için **umask** komutu kullanılır.

umask [nnnn]

nnnn sayısal değeri modu ayarlar.Bütün yeni dizinler .(nokta)ve ..(nokta-nokta)standart girişine sahiptir.

Çıkış Statüleri:

Bu komut çıkış değeri olarak:

0 : Bütün dizinler başarı ile yaratıldı veya -p opsiyonu kullanıldı ve sonuçta bütün dizinler oluştu.

>0 : dizin yaratırken bir sorun oluştu.

Seçenekler:

-m : Yeni yaratılan dizin için Mode değişkeni ile belirtilen değere göre izin belirtilir.chmod komutu için Mode değişkeni Mode parametresi ile hem sembolik hem de sayıca aynı değeri alır.-m opsiyonunu sembolik format kullanarak belirttiğimiz zaman '+' ve '-' karakterler izine göre yazılır. '+' karakterler hata modu için izni ekler ve '-' karakter hata modu için olan izni yok eder.

-p : Yeni bir dizin oluştururken arada olmayan ara dizini yaratır.Eğer -p ile belirtilmezse her yeni yaratılan dizinin ana dizini mutlaka var olmalıdır. Var olan bir dizin -p ile belirtilmişse herhangi bir hata vermez.Aradaki dizinler aşağıdaki 3.örnekte mkdir komutları ile otomatik olarak yaratılmıştır.

--help : Kullanılan bir mesajı standart bir çıktı üzerine yazdırır ve başarıyla çıkar.

Örnekler:

1. Çalışılan son dizinde yeni isimli yeni bir dizin oluşturulacağı zaman:

```
$ mkdir yeni
$ _
```

2. Önceden yaratılmış /home/alperkiroglu/belge dizininde rwxr-xr-x izni olan yeni bir dizin yaratmak için

```
$ mkdir -m 755 /home/alperkiroglu/belge/yeni
```

3. Daha önce yaratılmış bir dizini tekrar yaratmak istediğimizde hata mesajı verir.

```
$ mkdir images
images:Don't specify an existing file .
```

4. Daha önce sadece okunabilir ve çalıştırılabilir amaçla açılmış bir dizin içine yeni bir dizin yaratılmaz.

```
$ mkdir -m 500 images
$ cd images
$ mkdir mail
mkdir:0653-357 cannot access directory ..
.:The file access permissions do not allow the specified action.
```


5. Birden fazla dizin yaratılabilir.

```
$ mkdir images html mail  
$ _
```

[Kaynak 12]

rmmdir Komutu:

İçi boş olan dizinleri siler.mkdir ile yaratılan dizinler rmmdir komutuyla yok edilir.

Söz Dizimi:

```
$ rmmdir dizin adı
```

şeklinde tanımlanmaktadır.Bir rmmdir komutuyla birden fazla dizin silinebilir.Dizinlerin silinebilmesi için;dizinlerin içinde hiçbir şey olmamalı ve komut satırında belirtilen dizinin kök dizinine yazım hakkı olması gerekir. Eğer dizin sadece okunabilir ,bir sistem dosyasında yer alıyorsa, rmmdir komutu hatalı çalışır. Eğer dizinin içi doluyrsa rmmdir komutu hata verir."ls -al" komutuyla dizinin içinin boş mu dolu mu olduğunu anlayabilirsiniz.

Seçenekler:

-p : Komutta belirtilen yolda bulunan bütün dizinler silinir.Kök dizinleri bos olmak zorundadır ve kullanıcı dizinleri sistemden uzaklaştırmadan önce kök dizini üzerinde yazma hakkına sahip olmalıdır.

Komuttan çıkış:

0 : Dizin parametresiyle belirtilen her bir dizin başarıyla silinmişse;

>0 : Hata oluşmuşsa döner.

--help : Komutun kullanım seklini standart çıktıda belirtir.

Örnekler:

1. home/alper/mail isimli dizini yok etmek için:

```
$ cd /home/alper  
$ rmmdir mail  
$ _
```

2. Bu komut önce mail, daha sonra alper ve son olarak da home dizinini siler.Eğer dizinin içi boş değilse ve kullanıcıya yazma hakkı tanınmamışsa, komut işlevine son verir.

```
$ rmmdir -p /home/alper/mail  
$ _
```

3. Eğer silinmesi istenilen dizinin içi dolu ise hata mesajı görüntülenir.

```
$ rmmdir projeler  
$ projeler not empty
```

4. Daha önce yaratılmamış dizini silmek istersek hata mesajı verir.

```
$ rmdir yok
rmdir:0653 -609 cannot remove yok.
A file or directory in path name doesn't exist.
$ _
```

5. "--help" opsiyonu ile rmdir komutunun kullanım biçimi ve opsiyonları gösterilir.

```
$ rmdir--help
Usage:rmdir [-p] Directory name
```

6. Birden fazla içi bos dizini siler.

```
$ rmdir images html
$ _
```

7. Dosyayı **rmdir** komutu ile silinmez."index.html" dosyadır.

```
$ rmdir index.html
rmdir: 0653-609 Cannot remove index.html.
A parameter must be a directory.
$ _
```

8. Komut satırında silinmesi istenen dizinin içinde isek, o dizini silmek için dizinden dışarı çıkılmalıdır.

```
$ pwd
/projeler
$ rmdir projeler
rmdir: 0653-609 Cannot remove projeler.
A file or directory in the path name does not exist.
$ cd ..
$ rmdir projeler
$ _
```

[Kaynak 12]

date Komutu

Sistem tarihini ve zamanını gösterir. date komutu "date +%X" şeklinde kullanılır. Format "+%X" şeklindedir. X'in yerine gelebilecek seçenekler aşağıda sıralanmıştır.

Seçenekler:

S : saniyeyi göster
M : dakikayı göster
1-12 :arası saati göster
p :am or pm (sabah veya öğleden sonra)
H : 0-23 arası saati göster
T :hh:mm:ss formatında zamanı göster(h:saat-m:dakika-s:saniye)
Z :zaman dilimini göster

A :gün ismini tam olarak yaz
a : gün isminin ilk üç harfini yaz
w :gün ismini sayısal olarak belirtir (0-6)
d :ayın kaçı olduğunu belirt (1-31)
j :yılın kaçınıcı gününde bulunduğunu belirt (1-366)
U :yılın kaçınıcı haftasında bulunduğunu belirt (0-53)
m :yılın kaçınıcı ayında bulunduğunu belirt (1-12)
b :içinde bulunan ay isminin ilk üç harfini yaz
B :ay isminin tamamını yaz
y :hangi yılda bulunduğunu 2 basamaklı yaz
Y :hangi yılda bulunduğunu 4 basamaklı yaz
D :tarihi MM/DD/YY (ay/gün/yıl) formatında yaz
c : tarih ve zamanı aynı anda yaz

Örnekler:

```
$ date
Mon Jan 11 13:29:48 MEST 1999

$ date +%j
011

$ date +%D
01/11/99

$ date "+%A %B %Y"
Monday January 1999

$ date +%A%D%Y
Monday01/11/991999

$ date "+%j %U %Y"
011 02 1999

$ date +%Z
MEST

$ date +%A%B%Y
FridayJanuary1999

$ date "+%d %m %y"
15 01 99

$ date "+%a %b"
Fri Jan

$ date "+%I:%M %p"
10:44 AM

$ date "+%A %B %d %Y"
Friday January 15 1999
```

[Kaynak 12]

tar Komutu:

tar (Tape ARchive) komutu, dosyaları arşivlere yazarak veya arşiv saklama ortamından dosyaları açarak arşivleri kontrol eder. Yani size elinizdeki bir arşivin sonuna dosya ekleme veya bu arşivdeki dosyaları değiştirme olanağı sağlar. tar, verilerin arşiv araçlarına kaydedilmesi için kullanılan ender komutlardan biridir. Tipik olarak tar,teyplerle veya floppy disklerle kullanılır.Verileri arşive kaydetmek için kullanılan komutlardan kullanımı en kolay olanıdır.tar komutu dosyaları kaydeder, yedekler ve arşivin içeriğini listeler.

Seçenekler:

tar komutunun seçenekleri mecburi(gerekli) ve seçmeli(opsiyonel) olarak iki bölümde incelenebilir.Mecburi olan seçenekler tar komutunun faaliyetlerini kontrol eder ve -c,-r,-t,-v,-x seçeneklerini içerir.Bu mecburi seçeneklerden en az bir tanesi mutlaka tar komutu ile kullanılmalıdır.Mecburi seçenek kullanıldıktan sonra seçmeli seçenekler kullanılabilir ama zorunlu değildir.

Mecburi seçenekler:

- c : Yeni bir arşiv oluşturur ve belirtilen dosyaları arşive kaydeder.
- r : Bir veya daha fazla sayıdaki dosyayı arşivin sonuna ekler.
- u : Bir veya daha fazla sayıdaki dosyayı, eğer dosya arşivde yoksa veya bu dosyalarda bir değişiklik yapılmışsa,arşivin sonuna ekler.

Not:-r ve -u seçenekleri hiçbir teyp cihazı için geçerli değildir.Çünkü bu cihazlar teypin sonuna bilgi eklenmesine izin vermezler.

- t : Arşivdeki görünüş sırasına göre dosyaları listeler.(Dosyalar birden fazla listelenebilir.)
- x : Bir veya birden fazla sayıdaki dosyayı arşivden çıkarıp alır. Eğer dosyayı belirtmemişseniz, tar bütün dosyaları arşivden çıkarır. Bir arşivde aynı dosyadan birkaç kopya olduğu zaman çıkarılan son kopya önceki kopyaların üzerine yazılır. Eğer çıkarılan dosya sistemde yoksa, dosya oluşturulur.Eğer uygun hakkınız varsa, tar komutu bütün dosyaları ve dizinleri aynı sahipleri ve grup ID'leri ile birlikte ortaya çıkarır. Eğer yoksa, dosyalar ve dizinler sizin sahip olduğunuz ID ile ortaya çıkar.

Secimli seçenekler:

-b Blocks : Her kayda,512KB'lık blok sayısını belirtir.Teyp kayıtlarına uygun olarak ilk varsayılan(default) da maksimum da 20'dir.Kayıt boşluklarına göre, büyük blok faktörleriyle yazılan teypler,her kayıta 1 blok olan teyplerden daha çok veri tutarlar. (-x veya -t seçeneklerinde)Teypler okunduğu zaman otomatik olarak blok genişliği belirlenir.Arşivler -u ve -r fonksiyonlarıyla günlendiği zaman, ortaya çıkan kayıt genişliği kullanılır.tar komutu sadece -c seçeneği ile yeni arşiv yaratırken belirtilen [Blocks] parametresini kullanarak arşivleri yazar. -f seçenekli sıradan dosyaların çıktısı için,disk bloklarının genişliğine uyan bloklama faktörünü kullanarak diskte boş yer koruyabilirsiniz. (Mesela 2048 byte disk bloğu için -b4 seçeneği kullanılabilir.)

-C Directory : [Directory] değişkeninde belirtilen dizini cd komutuna parametre olarak alır ve uygular.tar,-C seçeneğini kullanarak, farklı dizinler altındaki dosyaların arşivlenmesine izin verir. -C seçeneği, diğer bütün seçeneklerden önce bulunmalıdır ve verilen dosya isimlerinin listesinde görünebilir.

-d : Normal olarak tar komutu özel dosyaları arşivlemeyecektir. Arşive -d seçeneği ile yazıldığında, tar, -x seçeneği ile birlikte boş dizinleri, özel dosyaları ve FIFO piped işlemini ortaya çıkarmayı mümkün kılar.

Not: Herkes özel dosya arşivleyebildiği halde, sadece root kullanıcı yetkilerine sahip olan kullanıcılar bu dosyaları bir arşivden çıkarıp alabilir.

-F : Dosyanın tipini arşivlemeden önce kontrol eder. SCCS, RCS, core, errs, a.out isimli dosyalar ve .o (nokta o) ile biten dosyalar arşivlenmez.

-h : tar komutunun sembolik linkleri, normal dosya veya dizinler gibi izlemesini sağlar. Normalde tar komutu sembolik linkleri içermez.

-i : Header checksum hatalarını atlar. tar komutu arşivdeki her dosya için bir checksum içeren file header yazar. Bu seçenek belirtilmediği zaman sistem; header bloklarının içeriğini, checksum'ları yeniden hesaplayarak doğruluğunu saptar ve bir uyumsuzluk olduğu zaman bir "dizin checksum hatası" ile durur. Bu seçenek belirtildiği zaman tar komutu hatayı kaydeder ve geçerli bir header blok bulana kadar ilerisini tarar. (Bu, birden fazla parçaya ayrılmış tar dosyalarının herhangi birinin bozuk olması durumunda, sonrakilerin açılmasını sağlar.)

-m : Dosyaların tarihi, arşivden çıkarıldıkları tarih olarak değiştirilir. Ön tanımlı (default) durum, dosyaların arşivlendiği tarihlerin korunması yönündedir.

-o : tar komutunun eski versiyonlarıyla uyumunu sağlar. Bu seçenek okumada kullanıldığı zaman, çıkarılıp alınan dosyayı arşivdekiler yerine, programı çalıştıran kullanıcının UID ve GID'sini taşır. Bu sıradan kullanıcı için olan default davranıştır.

-s : Eğer tar komutu iki dosyayı bağlama girişiminde (sıradan link) başarısız olmuşsa -s seçeneği ile sembolik link oluşturmaya çalışır.

-v : İşleme giren dosya adlarını listeler. -t seçeneği ile -v, dosya genişliğini, son değişim zamanını, UID, GID haklarını içeren teyp bilgilerini ayrıntılı olarak verir.

-w : Dosya adını izleyen işlemi görüntüler ve kullanıcının onayını bekler. Eğer karşılık doğrulayıcıysa işlemi yapar; değilse dosyayı atlar.

Örnekler:

1. Örneğin elimizde sec.html, komut.html adında iki html dosyası bulunsun. Bunları html.tar adlı dosyayı oluşturup bu arşiv dosyasında saklamak için;

```
$ tar -cvf html.tar sec.html komut.html
a sec.html 1 blocks.
a komut.html 1 blocks.
```

2. Şimdi de "lost" adlı bir dizini find.tar dosyasına arşivleyelim.

```
$ tar -cvf find.tar lost
a lost
a lost/html1
a lost/html1/first.html 1 blocks.
a lost/html2
a lost/html2/second.html 1 blocks.
```

Görüldüğü gibi "lost" içindeki alt dizinler ve dosyalar da arşive dahil edilmiştir.

3. Daha önceden oluşturulmuş html.tar arşivimize ornek.html dosyasını eklemek için;

```
$ tar -rvf html.tar ornek.html
a ornek.html 1 blocks.
```

4. Eğer Jan 10 14:14 tarihinde "\$tar -xmf html.tar komut.html" komutunu yazarsak buradaki m seçeneği komut.html dosyasının tarihini, bu komutu yazdığımız tarih olarak değiştirecektir.

```
$ ls -l
total 512
-rw----- 1 bbm903 obbm 143 Nov 26 13:59 #pico16830#
-rw-r--r-- 1 bbm903 obbm 353 Dec 18 14:39 dead.letter
-rw-r--r-- 1 bbm903 obbm 10240 Jan 10 13:40 html.tar
-rw-r----- 1 bbm903 obbm 65 Jan 10 14:14 komut.html<-Zamana dikkat
drwx----- 2 bbm903 obbm 512 Jan 09 15:28 mail
```

Görüldüğü gibi komut.html dosyasının değişim zamanı,onu arşivden çıkarttığımız zaman olarak değişti.

5. Eğer dosyaları -w seçeneği ile Command.tar dosyası olarak arşivlersek;

```
$ tar -cwf Commands.tar connect.wav
Archive connect.wav? y
```

Bu soruya olumlu yanıt verirse dosya arşivlenir.Oluşturulmuş olan arşivin içindekilerin görmek için;

```
$ tar -tvf Commands.tar
-rw-r----- 6274 6075 14354 Jan 08 17:21:46 1999 connect.wav
```

[Kaynak 12]

cat Komutu:

Dosya yaratmaya veya çıktı almaya yarar.

Seçenekler:

- b : boş olmayan satır sayısı
- n : çıktıya giden tüm satır sayısı
- e : her satırın sonuna \$ görüntüler.
- u : UNIX'e uygunluğu önemsiz
- v : karakterlerden önce yer alan '^' ile TAB ve LFD hariç kontrol karakterlerini görüntüler.
- et: satır sonuna \$, TAB yerine ^I görüntüler.
- h : kullanım mesajı verir.
- s : çok sayıda yer alan boşlukları bire indirir.

Söz Dizimi:

UNIX'te çok kullanılan ,çok işlevli bir komuttur.Bu işlevlerden bir tanesi,MS-DOS'taki TYPE komutu ile aynıdır. En basit kullanım formu:

```
$ cat dosya_adı şeklindedir.
```

Bu formda kullanıldığı zaman,dosya_adı adlı dosyayı ekrana gönderir.Standart çıktı birimi genellikle ekran olduğu için,dosyayı ekrana listelemek için kullanılır.Örnek komut:

```
$ cat /etc/motd
```

Dosya yaratmak için kullanılan form ise daha farklı...

```
$ cat > yenedosya
```

[Kaynak 12]

head Komutu

Dosyaların ilk kısmını listeler

Seçenekler:

b :512 byte blok

k :1 kilobyte blok

m :1 mega byte blok

harflerinden biri eklenir.

-q , --quiet , --silent :dosya adı başlarını hiçbir zaman listelemez.

-v , --verbose :dosya adı başlarını her zaman listeler.

Örnekler:

```
$head -4 readme
```

Head komutu, belirtilen dosyanın istenen satır kadarını konsolda listeler.

[Kaynak 12]

find Komutu:

Bir kılavuz ağacı içinde kütüklerin aranmasıdır.Bu kılavuz, find komutunun GNU versiyonunu ele almaktadır. find komutu, belirtilen kütük isimlerinin kılavuz ağacında aranmasını sağlar. Bu arama işlemi ifadelerde belirtilen biçime göre yapılır. İfadeler, soldan sağa doğru, öncelik kurallarına uyularak değerlendirilir.

`- ', `(' , `)` , ` , ' veya `!' işaretlerinin herhangi birisiyle başlayan ilk argüman ifadeler kısmının başlangıcı olarak kabul edilir. Bundan önce gelenler aranacak yolları, sonra gelenler ise geri kalan ifadeleri oluştururlar. Eğer herhangi bir yol belirtilmemişse, içinde bulunan kılavuz kullanılır. Eğer herhangi bir ifade belirtilmemişse, ` -print ' ifadesi kullanılır. find, bütün kütükleri başarıyla işlenmesi halinde 0, bir hata olduğu takdirde 0'dan daha büyük bir değer alır.

Söz dizimi:

```
find [yol...] [ifade]
```

Seçenekler:

Bütün seçenekler her zaman doğru geri döndürürler ve ifadelerdeki sıraları göz önünde bulundurulmaksızın işlem üzerinde her zaman etkilidirler. Bu nedenle, komut satırının anlaşılabilirliği için, ifadelerin başında kullanılmalarında fayda vardır.

-daystart : Bugünün başlangıcından sonraki zamanları ölç (-amin, -atime, -cmin, -ctime, -mmin ve -mtime için).

-depth : Kılavuzun içeriğini kılavuzun kendisinden önce işleme koy.

-maxdepth n : Komut satırında belirtilen argümandan en fazla n (negatif olmayan tamsayı) basamak kılavuz in. n'in 0 olması halinde işlemler komut satırında belirtilen argümanlar üzerinde uygulanır.

-mount :Diğer kütük sistemlerinin kılavuzlarına inilmez.

-noleaf : Kılavuzların, link sayılarının iki eksiği kadar alt kılavuzlara sahip oldukları göz önünde bulundurulmaz. Bu seçenek, Unix kılavuz-link düzeni dışındaki bir kütük sisteminde (CD-ROM veya MS-DOS kütük sistemi) arama yapıldığında gereklidir.Unix kütük sisteminde normalde her kılavuzun en az iki linki vardır: adı ve `.' girişi. Ayrıca her alt kılavuz (varsa) için bir `..' vardır.

-xdev : Diğer kütük sistemlerinin kılavuzlarına inilmez.

Örnekler:

`find ".profile"` :profile isimli kütüğü o anki dizinden başlayarak alt dizinler boyunca ara

`find "*doom"` :o anki dizinden başlayarak alt dizinler boyunca sonu doom ile biten tüm kütükleri bulur.

`find / -name "startx" :root (/) dizininden başlayarak tüm alt dizinler boyunca startx isimli kütüğü arar.`

`find . -user root :o anki dizinden başlayarak tüm alt dizinler boyunca kullanıcısı root olan tüm kütükleri bulur.`

[Kaynak 12]

echo Komutu

Belirtilen ifadeleri görüntülemek üzere echo komutundan yararlanılır.Genellikle kabuk programlarında kullanılır.

Örnek:

```
$echo"Merhaba"  
Merhaba  
$
```

[Kaynak 12]

who komutu:

Sistem yöneticilerinin en sık yaptıkları işlemlerden birisi, sistemde o anda kimlerin çalıştığını saptamaktır. Bu sayede, kullanıcıların sisteme giriş zamanlarını, sistem tarihinin en son değiştiriliş tarihini, sistemin çalışma düzeyini vb. bilgileri elde edebilecektir. who komutu aşağıda gösterildiği gibi tanımlanmaktadır. Belirtilen seçenekler dışında istenirse doğrudan doğruya kullanılabilir.

Söz Dizimi:

who [seçenekler] [isimler]

Seçenekler:

- u** : Sadece o anda sisteme bağlanmış kullanıcıları listeler. Liste içinde aşağıdaki kolonlar yer almaktadır.
 - name** : Kullanıcının bağlantı adı.
 - line** : /dev dizininde tanımlanan cihaz.
 - time** : Kullanıcının sisteme bağlandığı zaman (ay, gün ve saat cinsinden).
 - Activity** : Kullanıcının sisteme bağlandığı andan itibaren geçen süre. Eğer o sırada işlem yapıyorsa (.) işareti görüntülenir.
 - l** : Sistemde tanımlı, fakat henüz açılmamış terminalleri listeler.
 - q** : Seçeneği who komutunun sadece kullanıcı isimlerini listelemesini sağlar. Bu seçenek ile birlikte başka seçenekler de kullanılırsa bunlar göz önüne alınmaz.
 - d** : Seçeneği bitmiş ve init tarafından yeniden çalıştırılmayacak olan işlemleri listeler. Tamamlanmış işlemlerin nasıl sonuçlandığı hakkında bilgi sağlar.
 - b** : Sistemin en son açılış tarihini ve zamanını görüntüler.
 - r** seçeneği init işleminin o andaki çalışma düzeyi hakkında bilgi sağlar. Bu sayede sistemin tek ya da çok kullanıcıli durumda olup olmadığı anlaşılır.
- Ayrıca "Who am i" komutu ile kullanıcı kendi bağlantı isminin ne olduğunu sisteme sorabilir.

Örnekler:

Kullanıcı kendi bağlantı ismini öğrenmek isterse;

```
$ who am i
bbm883          pts/27      Jan   11   10:35  (dblab03.cs.hun.edu.tr)
$_
```

Sistemde kimlerin çalıştığını görmek amacıyla who komutu en basit biçimiyle kullanılabilir.

```
$ who
bbm883          pts/30      Jan   11   10:35  (dblab05.cs.hun.edu.tr)
bbm884          pts/27      Jan   11   13:03  (dblab03.cs.hun.edu.tr)
bbm842          pts/28      Jan   11   16:32  (dblab04.cs.hun.edu.tr)
b964655         pts/53      Jan   11   12:13  (bimpc.02.cs.hun.edu.tr)
bbm904          pts/84      Jan   11   14:56  (cr19.ee.hun.edu.tr)
$_
```

Birinci kolon kullanıcı ismini, ikinci kolon terminal ismini verir. Geriye kalanlar ise zaman bilgileridir. Kullanıcılar hakkında biraz daha bilgi almak için -u seçeneği kullanılır.

[<http://atlas.cc.itu.edu.tr/~coskunu/>]

chmod Komutu

Dosya ve dizinlere erişim yetkilerini değiştirmek için kullanılır.İzin modlarını iki türlü ifade etme olanağı vardır.Bunlardan birincisi onaltılı sisteme (octal) göre tanımlama,ikincisi ise sembolik tanımlama şeklindedir.Onaltılı sisteme göre:

Söz Dizimi:

```
chmod [-R] [-h] [-f] izin kodu {dosya...|dizin...}
```

Yetki tanım grupları aşağıdaki gibi sayısal birer değerle eşleştirilir:

r : 4
w : 2
x : 1

Bu eşleştirmeler dosya/dizin sahibi (owner),grup ve diğerleri (others) için ayrı ayrı yapılır.

Sembolik Tanımlama :

```
chmod [-R] [-f] [-h] [[u][g][o][a]]{[-|=+}[r][w][x][s]]{dosya...|dizin...}
```

Bu formda chmod komutu sembolik harflerle kullanılır.Bunlar:

İzin değişiminin kiminle ilgili olduğu:

u : dosya/dizinin sahibi (user)
g : dosya/dizinin sahibiyle aynı grupta olanlar (group)
o : diğer kullanıcılar (others)
a : herkes (all)

İzinler üzerindeki işlemler için:

+ : yetki ekleme
= : yetki eşitleme
- : yetki çıkarma

İzinlerin sembolleri:

r : okuma yetkisi (read)
w : yazma yetkisi (write)
x : çalıştırma yetkisi (execute)

Seçenekler:

-f : Geçersiz izinler ve kullanım cümleleri dışında bütün hata mesajlarını ihmal eder.
-h : Sembolik link tarafından gösterilen dosya veya dizinler için mod değişimini ihmal eder.
-R : Tekrarlı olarak dizinlere iner.-R seçeneği her bir dizinin ve belirlenmiş bir şablona uyan bütün dosyaların dosya modu bitlerini değiştirir.

Örnekler:

1 . prog isimli dosyanın erişim yetkilerinin rwxr-x--x olması isteniyor.Bu yetki kalıbı üçer üçer ayrılmış olarak düşünülüp (rwx r-x--x),yukarıdaki eşleştirmelere göre verilmek istenen yetkilere karşılık gelen sayılar üçer üçer toplanır ve elde edilen üç rakam yan yana getirilip üç basamaklı bir sayı elde edilir.Başka bir deyişle:

Dosya sahibi için : $4+2+1 = 7$
Grup için : $4+0+1 = 5$
Diğer kullanıcılar için : $0+0+1 = 1$
Elde edilen üç basamaklı sayı =751

2 . prog isimli dosyanın sahibine okuma,yazma ve çalıştırma;grubuna ve diğerlerine de yalnızca çalıştırma yetkisi vermek için yazılması gereken kod.

```
$ chmod 711 prog  
veya  
$ chmod go-rw prog
```

3 . Örnek olarak deneme1 ve deneme2 isimli dosyaların gruplarına kuma(read) izni verelim.

```
$ chmod g+r deneme 1 deneme 2
```

4 . Bir defada birden çok izin değiştirmek için:

```
$ chmod go-w+x deneme.htm  
Bunu ayrı ayrı yazmak istersek:  
$ chmod g-w deneme.htm  
$ chmod g+x deneme.htm  
$ chmod o-w deneme.htm  
$ chmod o+x deneme.htm
```

kodlarını yazmamız gerekir.

[Kaynak 12]

chown Komutu:

Bir dosyanın veya dizinin, parametresinde tanımlı bulunan dosya veya dizin sahibi parametrelerini değiştirir.Öyleki dosya veya dizin sahibi parametresi /etc/passwd dosyasında, kullanıcı kimliği(user ID) veya bağlantı ismi(login name) olarak kayıtlı bulunmak zorundadır.Aynı şekilde, dosya veya dizin gurubu parametresi de /etc/group dosyasında, gurup kimliği group ID) veya gurup ismi(group name) olarak kayıtlı bulunmak zorundadır.

```
chown [-f] [-h] [-R] sahibi[:Grubu] {Dosya | Dizin}
```

Seçenekler:

-f : Kullanım için hata mesajları dışında bütün error mesajlarının ihmal edilmesini sağlar.

-h : Bulunan bir sembolik linkin ve sembolik linke bağlı olmayan dosya ve dizinlerin sahipliğini değiştirir.

-R : Tekrarlamalı olarak dizinlere iner ve her dosyanın sahipliğini değiştirir. Sembolik linkle karşılaşıldığında ve link bir dizini gösteriyorsa dizinin sahipliği değişir.

Örnekler

1 . inf.dat isimli dosyanın sahibini değiştirmek;

Bir dosyanın sahibini öğrenmek amacıyla ls komutu çektiğimizi düşünelim, ekrandaki görüntü aşağıdaki gibidir,

```
$ ls -l
-rwxrwxr-x  1  bbm819  obbm  5744  oct 15 12:11  inf.dat
$ _
```

yukarıdaki inf.dat isimli dosyanın sahibi bbm819 isimli kullanıcıdır, gurubu ise obbm'dir. Dosyanın sahibini bbm870 isimli kullanıcı yapmak istiyoruz. O halde chown komutu,

```
$ chown bbm870 inf.dat 'dir
-rwxrwxr-x 1  bbm870  obbm  5744  oct 15 12:11  inf.dat
```

şimdi kullanıcı izni artık bbm870'e geçmiştir.

2. (/etc/src) isimli dizinin içindeki tüm dosyaların sahipliği değiştirirken, sistemde bbm819 isimli kullanıcının olduğunu varsayalım,

```
$ chown -R bbm819 /etc/src
```

komutu, /etc/src dizinini ve alt dizinlerinin sahipliğini bbm819'a vermiş oldu.

[Kaynak 12]

mount Komutu:

Üzerinde bir dosya sistemi olan bir disk birimine veya parçasına okuma veya yazma amacıyla ulaşabilmek için bu birimi veya parçayı , / dosya yapısında bir alt dizine mount edilmiş olması gerekir./ dizini, bilgisayarın açılması sırasında otomatik olarak mount edilmektedir. Eğer bu disk dizini, bilgisayarın açılması aşamasında mount edilmişse, o bilgisayar zaten açılmaz. Diğer disk ve disk parçalarının otomatik olarak mount edilmesi için gerekli işlemler ise, sistem yöneticisi(root) tarafından yapılır. Otomatik olarak mount edilmesi istene diskler ve mount edilecekleri dizinler /etc/filesystems dosyasında tanımlanır. Genel olarak söz dizimi aşağıdaki gibidir.

```
mount [ -f ] [ -n Node ] [ -o Options ] [ -p ] [ -r ] [ -v VfsName ] [ -t
Type | [ Device | Node:Directory ] Directory | all | -a ]
```

Seenekler:

- a** : /etc/filesystems içindeki mount=true özellikli tüm kütük sistemlerini **mount** eder.
- all** : **-a** ile aynı işleve sahiptir.
- f** : Sistemin başlatılması sırasında kök kütük sistemi üzerinde mount etmeyi yapılabilir hale getirir.
- n (node)** : Birleştirilecek rehberi tutan yabancı düğümü belirtir.
- o (options)** : Seçenekleri belirtir. Komut satırına girdiğiniz seçenekleri ayırmak için sadece virgül kullanılmalıdır, virgül ve boşluk birlikte kullanılmamalıdır. Aşağıdaki file-system-specific seçeneklerini tüm virtual-file-system' lerde uygulanamaz .
- t** : type=Type vasfını içeren ve mount edilmemiş /etc/filesystems kütüğündeki bütün dizinleri birleştirir. Type değişkeni, grubun adını belirtir.

Örnekler:

1. \$ mount

Tüm mount edilmiş kütük sistemlerini ve bunların özelliklerini gösterir. Ekranı aşağıdaki ifadeler yazılır.

node	mounted	mounted	vfs	date	options	over
	/dev/hd0	/	jsf	Dec 17 08:04	rw, log	=/dev/hd8
	/dev/hd1	/temp	jsf	Dec 17 08:04	rw, log	=/dev/hd8
	/dev/hd2	/home	jsf	Dec 17 08:06	rw, log	=/dev/hd8
	/dev/hd3	/usr	jsf	Dec 17 08:06	rw, log	=/dev/hd8
sue	/home/local/src	/usr/code	nfs	Dec 17 08:06	ro , log	=/dev/hd8

2. \$ mount /dev/hd0

0 numaralı diski /home dizinine mount eder.

\$ mount -n nodeA /home/tom.remote /home/tom.local

node A isimli bilgisayarın /home/tom dizinini yerel bilgisayarın /home/tom dizinine mount eder.

3. \$ mount all

/etc/filesystems kütüğünde tanımlanmış mount=true özellikli tüm standart kütük sistemlerini mount eder.

4. \$ mount -r /dev/hd2 /usr

/dev/hd2 diskini /usr dizinine sadece okuma amacıyla mount eder.

5. \$ mount /cdrom

Bu komut /etc/filesystems kütüğünde bulunan ve mount noktası /cdrom olan kütük sistemini mount eder. Mount daha çok bu şekilde kullanılır ve bu kullanım çok kullanılan kütük sistemleri içindir. Burada uzun uzun komut yazmak yerine sadece mount noktası yazılmıştır. Yazılması gereken diğer bilgiler /etc/filesystems kütüğü içinde bulunur.

6. `$ mount /dev/sr0`

Bu komut /etc/filesystems kütüğünde bulunan /dev/sr0 isimli cdrom sürücüsünü mount eder. Mount daha çok bu şekilde kullanılır ve bu kullanım çok kullanılan kütük sistemleri içindir. Burada uzun uzun komut yazmak yerine sadece düğüm yazılmıştır. Yazılması gereken diğer bilgiler /etc/filesystems kütüğü içinde bulunur.

7. `$ mount -t remote`

/etc/filesystems altında bulunan tipi remote olan tüm dosya sistemlerini mount eder.

8. `$ mount -o ro /dev/fd0 /mnt`

Üzerinde UNIX dosya sistemi olan yazmaya karşı korumalı olan bir disketi /mnt dizinine mount eder.

9. `$ mount -o rw /dev/fd0 /mnt`

Üzerinde UNIX dosya sistemi olan bir disketi /mnt dizinine hem yazma hem de okumaya izinli olarak mount eder.

[Kaynak 12]

umount Komutu:

Mount edilmiş bir kütük sistemini, bilgisayarın /kütük sisteminden ayırmak için kullanılır. (Bu is genellikle disket ve CD 'ler için anlamlıdır.) Genel olarak söz dizimi aşağıdaki gibidir.

```
{umount | unmount} [-f] [-a] | [all | allr | Device | Directory | File |
FileSystem | -n Node | -t Type ]
```

Secenekler:

-a :Mount edilmiş tüm kütük sistemlerini unmount eder.(bağlantılarını keser)

all :Mount edilmiş tüm kütük sistemlerini unmount eder.(bağlantılarını keser)

allr :Bilgisayara dışarıdan mount edilmiş tüm kütük sistemlerini unmount eder.(bağlantılarını keser)

Dışarıdan mount edilmiş şeyler device,directory,file ve filesystem parametrelerini de kapsar.Dolayısıyla allr komutu tüm bu mount edilmiş şeylerinde bağlantısını keser;umount eder.

-n node:Bilgisayara dışarıdan mount edilmiş diğer bir bilgisayarın(node A) tüm bağlantılarını keser;umount eder.

-t type :Tipi type olan tüm kütük ve kütük sistemlerini unmount eder;bağlantılarını keser.

Örnekler:

1. `$ umount -a | all`

Mount edilmiş tüm kütük sistemlerini unmount eder.

2. `$ umount allr`

Bilgisayara dışarıdan mount edilmiş tüm kütük ve kütük sistemlerini unmount eder.

3. `$ umount -n nodeA`

A bilgisayarından mount edilmiş tüm kütük ve kütük sistemlerini unmount eder.

4. `$ umount -t test`

Bilgisayara tipi test olarak mount edilmiş tüm kütük ve kütük sistemlerini unmount eder.

5. `$ mount -r -t hsfs /dev/sr /cdrom`

Komutuyla mount edilmiş bir CD'yi unmount etmek için:

6. `$ umount /dev/sr0`

veya

`$ umount /cdrom`

komutları kullanılabilir.

7. `$ mount -r -t pcfs /dev/fd0 /disket`

komutuyla mount edilmiş bir disketi unmount etmek için:

8. `$ umount /disket`

veya

`$ umount /dev/fd0`

komutları kullanılabilir.

[Kaynak 12]

chgrp Komutu:

Bir dosyanın yada klasörün grup sahipliğini değiştirmeye yarar.

Sözdizimi:

```
chgrp [ -f ] [ -h ] [ -R ] Grup { Dosya..... | Klasör..... }
```

Tanım:

chgrp komutu daha belirtilen grupla ilişkilendirilmiş olan dosya veya klasörü belirtilen grup ismi veya grup ID si ile değiştirir. Bir sembolik linkle karşılaştığında ve siz **-h** parametresini özelleştirmedığınız zaman, "chgrp" komutu belirtilen linkin grup sahipliğinde olan dosya veya klasörün göstermekte olduğu linki değiştirir ama linkin kendi grup sahipliğini değiştirmez.

Eğer **-h** ve **-R** parametrelerini özelleştirirseniz, "chgrp" komutu belirtilen klasörleri özyineli olarak azaltır, ve sembolik linkle karşılaştığı zaman, linkin grup sahipliği değiştiriliyor ama linkin işaretlediği dosya veya klasörün grup sahipliğinde bir değişiklik meydana getirmiyor

Parametreler:

- f : Kullanım hataları hariç bütün hataları çıkartır.
- h : sembolik linkin grup sahipliğini değiştirir ama linkin işaretlediği dosya veya klasörün grup sahipliğini değiştirmez.
- R : Klasörleri özyineli olarak indirger,her dosya için belirtilen grup ID'sini yerleştirir.Eğer bir link ile karşılaşır ve link bir klasörü işaretliyse,bu klasörün grup sahipliğini değiştirir ama klasörün daha içlerine girilmez.

Çıkış durumu:

Bu komut aşağıdaki değerleri döndürür

0 : Başari ile tamamlandı.

>0 : Bir hata oluştu.

Örnekler:

1. dosya1 isimli dosya veya klasörün grup sahipliğini grup1'e değiştirmek için:

```
$ chgrp grup1 dosya1
```

Dosya1'in grup giriş izinleri artık grup1 olarak ayarlanır.

2. klasör1 isimli klasörün ve içindeki bütün dosyaların ve alt klasörlerin grup sahipliklerini grup1 olarak değiştirmek için:

```
$ chgrp -R grup1 klasör1
```

Klasöre1'in ve içindeki tüm dosyaların ve alt klasörlerin grup giriş izinleri grup1 olarak değiştirilir.

[Kaynak 12]

In Komutu:

Dizinleri veya dosyaları link eder.

Seçenekler:

-s : Dizinler ve de farklı disk parçalarındaki dosyalar link edilirken mutlaka kullanılmalıdır.

Örnekler:

1. Farzedin ki, bir UNIX sisteminin yöneticisiniz.Sizden,matlab isimli yeni bir uygulama programı yüklemenizi istediler.Ancak, uygulama programının bir gereği olarak , program paketine ilişkin dosyaların /usr/local dizininin altında açılacak bir dizinde yer alması gerekiyor. /usr diskinde, yeni programa ilişkin dosyalar için yeterli boş yer yok, ve silebileceğiniz gereksiz dosyalar da yok!İşte bu sorunu *ln* kavramıyla rahatlıkla çözebiliriz.Disklerin birinde, örneğin /home dizininin bulunduğu disk bölümünde (partition), yeni yükleyeceğiniz program için bir izin oluşturunuz.

```
# mkdir /home/matlab
```


Sonra bu dizini /usr/local altında yer alıyormuş gibi göstermek için,

```
# ln -s /home/matlab /usr/local/matlab komutunu veriniz.
```

Böylece, gerçekte /home altında yer alan matlab dizini, aynı zamanda /usr/local altında da varmış gibi davranacaktır. Bu dizini kullanırken isterseniz /home/matlab, isterseniz /usr/local/matlab dizin adreslerini kullanabilirsiniz. Bir başka deyişle, dosyalarının /usr/local altında bulunmasını isteyen matlab yazılımını kandırmış olursunuz.

2. Diyelim ki elinizde mhsb1995 isimli bir dosya var ve muhasebe departmanının kullandığı muhasebe programı bu dosyayı mutlaka bu isimde görmek istiyor. Öte yandan yeni satın aldığınız bir mali analiz programı, aynı muhasebe verilerini acct95 adıyla görmek istiyor. Söz konusu dosyanın adı mhsb1995 olduğunda muhasebe departmanının sorunu yok ama siz mali analiz programınızı çalıştıramıyorsunuz. Analiz çalışmaları için dosyanın adını değiştirirseniz, siz çalışabiliyorsunuz ama bu sefer muhasebe departmanındaki program kullanılmıyor. Dosyanın adını mhsb1995 olarak tutup, kendi analiz çalışmalarınız için acct95 adında kopya çıkardığınızda ve siz bu kopya üzerinde çalıştığınızda problem kısmen çözülüyor ama çok kullanıcı ortamında siz analizler üzerinde çalışırken öte taraftan muhasebe personeli yeni kayıtlar girip sizin analizlerinizin eskimiş kayıtlar üzerinde yapılmasına neden oluyorlar. İşte böyle bir durumda link kullanımı sizi kurtaracaktır.

```
# ln ./mhsb1995 ./acct95
```

Bu komutla mhsb1995 dosyasını acct95 isimli bir dosyaya bağladığınızda (aslında sadece tek bir asıl kopya var; o da mhsb1995. acct95 isimli bir dosya yok sadece diğer dosyanın bir başka adı), bu sayede mhsb1995 dosyasında yapılan her değişiklik acct95 diye tanınan dosyada da aynen gözlenebilmektedir. İşin bir başka yararlı tarafı da, acct95 adlı dosyanın diskte hiç yer kaplamayacak olmasıdır.

Uyarı:

1. İlk örnekte kullanılan ln komutunda -s diye bir parametre kullanılmış, ikinci örnekte ise kullanılmamıştır.

* Eğer ln komutuyla birbirlerine bağlanacak olan dosya sistemi elemanları birer dizinse; -s parametresini kullanmak zorundasınız.

* Eğer ln komutuyla birbirlerine bağlanacak olan dosya sistemi elemanları birer dosyaysa ve farklı disk parçalarında (bir başka deyişle, birbirinden farklı dosya sistemleri altında) yer alıyorsa, gene -s parametresini kullanmak zorundasınız.

* ln komutuyla , bir dizini ve bir dosyayı birbirlerine bağlayamazsınız. Bağlanacak olan elemanların ikisi de dizin, ya da ikisi de dosya olmalıdır.

Aynı dosya sisteminde yer alan ve birbirine bağlı olan dosyalardan birini silmeniz diğerini etkilemez. Asıl dosyayı silseniz bile UNIX bu bağlantıyı fark edip dosyayı diskten gerçekten silmeyecektir. UNIX her dosya için bağlantıları sayar ve her silme işleminde bağlantı sayısını bir azaltır. Gerçek silme işi bu bağlantı sayısı sıfır olunca yapılır.

2. Farklı dosya sistemlerinde yer alan bağlantılar için bu bağlantı sayma işine güvenmeyiniz. Farklı dosya sisteminde bağlantısı olan bir dosyayı silerseniz başınız derde girer. Asıl dosya silinir ve diğer sistemde gerçekte var olmayan bir dosyayı gösteren bir bağlantınız kalır.

[<http://atlas.cc.itu.edu.tr/~coskunu/>]

kill Komutu:

Çalışan süreci durdurmak için sinyal gönderir. Süreçlere sinyal göndermek için:

```
kill [-s {sinyalismi|sinyalnumarası}]Süreç tanıtım numarası (PID)
```

```
kill [-sinyalismi|-sinyalnumarası]PID
```

Sinyal isimlerini listelemek için:

```
kill -l[ExitStatus]
```

kill komutu çalışan süreci durdurmak için sinyal(varsayılan SIGTERM sinyali) gönderir.Bu varsayılan işlem süreci durdurur.Eğer bir süreci durdurmak istiyorsanız öncelikle PID değişkenlerinden süreç tanıtım numarasını(PID) öğrenmelisiniz. Kabuk , arka alanda çalışmakta olan tüm süreçlerin süreç tanıtım numarasını bildirir.(Bir pipeline'da birden fazla süreç başlatmadıkça kabuk son sürecin numarasını bildirir.)Komutların PID numaralarını bulmak için ps komutunu kullanabilirsiniz.Bir root kullanıcı herhangi bir süreci kill komutuyla öldürebilir.Eğer root kullanıcı değilseniz öldürmek istediğiniz süreci başlatmış olmanız gerekir.Yani sadece kendinize ait işlemleri öldürebilirsiniz.Kill komutu ile sinyaller tanımlanabilir.Sinyal ismi SIG öneksiz de tanınabilir. Eğer belirtilen sinyal numarası 0 ise,kill komutu belirtilen PID numarasının geçerliliğini kontrol eder.

Süreç Tanıtım Numarası(PID):

Durdurulacak süreç veya süreç grubunu gösteren ondalık sayıyı belirtir.Eğer PID numarası pozitif bir değerse, kill komutu süreç tanıtım numarası PID'e eşit olan süreci öldürür.Eğer PID numarasının değeri 0'sa, kill komutu süreç grup tanıtım numarası göndericinininkine eşit olan süreçlere sinyal gönderir.Sinyal, PID'i 0 veya 1 olan süreçlere gönderilmez.Eğer Pıd -1 ise ,kill komutu göndericinin etkili kullanıcısına ait tüm süreçlere işaret gönderir.Sinyal, PID'i 0 veya 1 olan süreçlere gönderilmez.Eğer negatif ve -1'den farklı ise kill komutu PID'in mutlak değerine eşit süreç grup tanıtım numarası olan tüm süreçlere sinyal gönderir.

Seçenekler:

-s{sinyalismi|sinyalnumarası} : Sinyali, SIGKILL -9 veya SIGTERM -15 gibi sinyal ismi veya sinyal numarası olarak belirtir.

-sinyalismi : SIGHUP gibi sinyal ismini belirtir.Bu kullanılmayan bir sözdizimidir.

-işaret numarası : Sinyal numarasını belirtir.Bu da kullanılmayan bir sözdizimidir.

NOT: Bu sözdiziminde varsayılacak negatif bir PID numarasını belirtmek için "--"i işaret olarak belirtmelisiniz.Aksi takdirde ilk operand bir sinyal numarası olarak yorumlanacaktır.

-l : Tüm sinyal isimlerin listeler.

-lExitStatus : Ortak SIG öneki kaldırılmış sinyal isimlerini listeler.

ExitStatus komutu aşağıdaki çıkış değerlerini döndürür:

0 : Her PID operandı için en az bir eşleştirilmiş süreç bulunur ve bu eşleştirilmiş süreç için belirtilen sinyal başarıyla işlemde geçirilir.

>0 : Bir hata meydana gelir.

Örnekler:

1-Çalışmakta olan işlemleri görüntülemek için iki yol vardır.Birincisi who do komutuyla olur.Bu komut yardımıyla, o anda hangi kullanıcının hangi işlemleri çalıştırdığı işlem numaraları ile birlikte öğrenilebilir.İkinci yol ise ps komutu yardımcı olur.

```
$ ps
  PID      TTY      TIME    COMMAND
 1140      015      0:03      sh
 1289      015      0:00      ps
 2341      015      0:00      sh
$_
```

En alt satırda yer alan ve 2341 numaralı işlemi öldürmek istiyoruz.Kill komutu aşağıda görüldüğü gibi kullanılabilir.

```
$ kill -9 2341
killed
$_
```

Son durumu görmek için yeniden ps komutunu kullanıyoruz.

```
$ ps
  PID      TTY      TIME    COMMAND
 1140      015      0:03      sh
 1289      015      0:00      ps
$_
```

2. Verilen bir süreci durdurmak için

```
$ kill 1342
```

giriniz.Bu, süreci varsayılan SIGTERM sinyali göndererek durdurur.Süreç 1342,SIGTERM sinyalinin geçersiz kılınan özel bir düzenleme yaparsa bu sürecin aslında durdurulamayacağına dikkat ediniz.

3. Varsayılan sinyali geçersiz kılan birçok süreci durdurmak için

```
$ kill -9 2098 1586
```

giriniz.Bu,9 sinyalinin,SIGKILL sinyalinin,2098 ve 1586 süreçlerine gönderir.SIGKILL sinyali normalde geçersiz kılınamayan özel bir sinyaldir.

4-Tüm süreçlerinizi durdurmak ve sonlandırmak için

```
$ kill -9 -0
```

giriniz.Bu,göndericinin süreç grup tanıtım numarasına eşit olan süreçlere 9 sinyalinin yani SIGKILL sinyalinin gönderir.

5-Sahip olduğunuz tüm süreçleri durdurmak için

```
$ kill -9 -1
```

giriniz.Bu,diğer workstationlarda başlatılsa da ya da diğer süreç gruplarına ait olsa da etkili kullanıcı tarafından sahiplenilen tüm süreçlere 9 sinyalini,SIGKILL sinyalini,gönderir.Eğer istediğiniz bir liste yazdırılıyorsa o da durdurulacaktır.

6-Bir sürece farklı bir sinyal kodu göndermek için

```
$ kill -USR1 1103
```

giriniz.Kill komutunun işlevi burada yanılıyor gibi.Çünkü SIGUSR1'i içeren birçok sinyal süreçleri durduramaz. SIGUSR1 üzerinde yapılan işlem,çalıştırdığınız özel bir uygulama ile tanımlanır.

[Kaynak 12]

ps Komutu:

Aktif işlemler hakkında çeşitli bilgileri görüntüler.

Kullanım :

```
$ps [seçenekler]
```

Seçenekler :

- e : Tüm işlemlerle ilgili bilgilerin görüntülenmesini sağlar.
- d : Grup liderleri hariç, tüm işlemler hakkında bilgi sağlar.
- a : Grup liderleri ile bir terminalle ilişkisi olmayan işlemler hariç tüm işlemler hakkında bilgi sağlar.
- x : Herhangi bir terminalle ilişkisi olmayan işlemleri görüntüler.
- f : Bilgi listenin tam olarak listelenmesini sağlar.
- l : Listenin ayrıntılı olarak görüntülenmesini sağlar.
- p liste : liste de ID numarası tanımlanan işlemlere ait özet bilgileri görüntüler.
- t liste : liste de adı yazılan terminallerden yürütülen işlemler hakkında özet bilgileri görüntüler.
- u liste : liste de belirlenen kullanıcılarla ilişkili işlemler hakkında özet bilgi sağlar.
- g liste : liste de tanımlanan grup liderlerinin ilişkili olduğu tüm işlemler hakkında bilgi sağlar.

Örnekler :

Kullanıcı ps komutunu parametresiz olarak kullanırsa ekrana kendisine ait işlemler hakkında kısa bilgi görüntülenir.

```

$ps
  PID  TTY  TIME  CMD
  1140 015  0:02   sh
  2170 015  0:00   ps
$_

```

İşlemler hakkında ayrıntılı bilgi almak için -l parametresi kullanılır.

```

$ps -l
F  S  UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  TTY  TIME  CMD
1  S  138  1140  1  0  30  20  462  56  80038ae8 015  0:02   sh
1  R  138  2170  1140 26  73  20  4bc  119  015  0:00   ps
$_

```

Görüldüğü gibi kullanıcıya ait işlemler ayrıntılı olarak kolonlar halinde -l parametresi ile görüntülendi. Görüntülenen bu kolonlardan;

F : İşlemler hakkında bilgi veren bazı kodları ekrana getirir.

- 00: işlem disk üzerinde
- 01: bellekte
- 02: sistem tarafından yapılan bir işlemi
- 04: bellekte kilitli
- 10: diske yazılıyor (swapping)
- 20: başka bir işlem tarafından izleniyor
- 40: bir başka izleyici

S : İşlemin durumu hakkında bilgi verir.

- O: mevcut değil
- S: uyuyor
- W: bekliyor
- R: çalışıyor
- I: ortada
- T: bitirilmiş
- Z: durdurulmuş

UID : İşlem sahibinin kullanıcı numarasını(ID) görüntüler

PID : İşlemin numarasını bildirir. Bu numara özellikle işlemin kill komutu ile sonlandırılmasında kullanılır.

PPID : Bir işlem yardımıyla başka işlemler yaratılıyorsa, ana işlemin numarasını verir.

C : İşlem yönetimi için işlemci gereksinimi verir.

STIME : İşlemin başlama zamanını bildirir.

PRI : İşlemin önceliğini belirtir. Büyük sayılar işlemin daha az işlem üstünlüğüne sahip olduğunu belirtir.

NI : İşlem önceliklerinin hesaplanmasında kullanılan değer. İstenilen işlemlerin öncelikleri değiştirilmek isteniyorsa, nice komutunda bu değerler göz önüne alınır.

ADDR : işlem bellekte ise işlemin bellek adresi, diskte ise işlemin disk adresini bildirir.

SZ : İşlemin bellekte ne kadar yer kapladığını belirtir.

WCHAN: İşlemin uyuma veya bekleme durumunu belirtir. Boş ise işlemin çalışmakta olduğunu bildirir.

TTY : İşlemin kontrol edildiği terminali bildirir.

TIME : İşlemin birikimli çalışma zamanını verir.

CMD : Çalıştırılan komutu bildirir.

Sistemdeki tüm işlemleri görüntülemek için -e parametresi kullanılır.

```
$ps -e
  PID   TTY   TIME   COMMAND
    0     ?    0:00   swapper
    1     ?    2:08   init
    0     ?    0:00   rddtya
    0     ?    0:00   wrttyb
  186   033    0:00   getty
 2143   002    0:01   cu
  165     ?    0:00   getty
 1719   032    0:00   getty
  169     ?    0:00   getty
 2158   017    0:00   sh
$_
```

Grup liderleri hariç sistemdeki tüm işlemleri görmek için -d parametresi kullanılır.

```
$ ps -d
  PID   TTY   TIME   COMMAND
    1     ?    2:08   init
    b    0:01   errdemon
 2158   017    0:00   sh
 2143   002    0:01   cu
 1644   017    0:00   menu.o
 2144   002    0:07   cu
 2166   015    0:00   ps
$_
```

Grup liderleri ile bir terminalle ilişkisi olmayan işlemler hariç tüm işlemleri görmek için -a parametresi kullanılır.

```
$ ps -a
  PID   TTY   TIME   COMMAND
    b    0:01   errdemon
 2158   017    0:00   sh
 2143   002    0:01   cu
 2169   015    0:00   ps
 1644   017    0:00   menu.o
 2144   002    0:07   cu
$_
```

Herhangi bir kullanıcıya ait işlemleri görüntülemek için -u parametresi kullanılır. Kullanıcılar arasında yer alan bbm854'e ait işlemler aşağıdaki gibi görüntülenir.

```
$ ps -u bbm854
  PID   TTY   TIME   COMMAND
 2158   017    0:00   sh
 1644   017    0:00   menu.o
 1456   017    0:01   sh
$_
```

[Kaynak 12]

wc Komutu

Dosyada bulunan satır,kelime veya karakter sayısını gösterir.

Seçenekler:

- l : Satır sayısını göster.
- w : Kelime sayısını göster.
- c : Karakter sayısını göster.

Örnekler:

```
$ wc index.html
59 95 1188 index.html

$ wc -l index.html
59 index.html

$ wc -w index.html
95 index.html

$ wc -c index.html
1188 index.html
```

[Kaynak 12]

gzip Komutu:

Gzip komutu dosya ya da dizinleri sıkıştırma, gziple sıkıştırılmış dosya ya da dizinleri açma işlemini yapar.Dosyanın kullanıcı hakları, erişim ve değiştirme zamanları gibi özellikleri korunarak, dosya .gz uzantılı bir dosya ile değiştirilir.Gzip komutu bir çok seçenek ile kullanılmakla beraber genel formatı aşağıdaki gibidir.

```
gzip [seçenekler] [-S sonek] [dosya adı[dosya adı...]]
```

Sıkıştırılmış dosyalar gzip -d , gunzip ya da zcat kullanılarak orijinal hallerine geri getirilebilir.

Seçenekler:

- a : Ascii text dosyalarındaki satır sonlarının saklanması sağlar.
- c : Çıktının standart çıkışta (ekran) görülmesini sağlar.
- d : Sıkıştırılmış dosyaların eski haline getirilmesini sağlar.
- f : Sıkıştırma ya da açma işlemlerinde; dosya başka yerlere link içeriyorsa, böyle bir dosya varsa ve üzerine yazma tehlikesi oluşmuşsa ya da dosya terminal tarafından kullanımdaysa, bu durumlarla ilgili uyarı vermeden sıkıştırma işlemini yapar.
- h : Gzip komutunun seçenekleri hakkında kısa bir bilgi verir.
- l : Sıkıştırılmış dosyalar hakkında bilgi verir.
- r : Gzip ile birlikte bir dizin adı verilmişse, bu dizinin tüm alt dizinlerini inceler ve bulduğu tüm dosyaları ayrı ayrı sıkıştırır.

-v : Her dosya için isim ve sıkıştırma yüzdesi bilgisini verir.

Sonek:

-S .suf : Sıkıştırdığınız dosyaların uzantılarının .gz yerine .suf olmasını sağlar.

Örnekler:

1. Gzip dosyanın türüne göre farklı oranlarda sıkıştırma işlemi yapabilmektedir.

```
$gzip -v deneme.out
deneme.out:      62.6% -- replaced with deneme.out.gz
$
```

Burada çalıştırılabilir bir dosya %62.6 oranında sıkıştırılmıştır.

```
$gzip -v ornek.txt
ornek.txt:       30.3% -- replaced with ornek.txt.gz
$
```

Burada da bir text dosyası %30.3 oranında sıkıştırılmıştır.

2. Gzip ile sıkıştırdığımız bir dosyayı gzip -d ile açmamız mümkün.

```
$gzip -d odev1.txt.gz
$ls -l
total 120
-rw-r--r--  1 bbm897  obbm  17336 Jan 06 12:20 gzip.txt
-rw-r-----  1 bbm897  obbm   3833 Dec 21 10:54 odev1.txt
-rw-r-----  1 bbm897  obbm   2209 Jan 06 12:17 odev2.txt
-rwx-----  1 bbm897  obbm  30574 Jan 06 12:20 resim.bmp
$
```

gzip komutu çalıştırılırsa, bu dosya, uni.kom.tar.gz dosyasına sıkıştırılır.

[Kaynak 12]

uname Komutu

İşletim sistemi hakkında bilgi verir.

Seçenekler:

- m : Makinenin donanım adını verir.
- n : Nodename ' i yazar.Nodename sistemin network iletişimi için kullanılan adıdır.
- r : Print the operating system release.
- s : Sistem ismini yazar.
- v : İşletim sisteminin versiyonunu yazar.
- a : Yukarıdaki bilgilerin tamamını yazar.

Örnekler:

1. \$ uname -m
 004210925398
2. \$ uname -n
 lidyA
3. \$ uname -r
 2
4. \$ uname -s
 AIX
5. \$ uname -v
 4

[Kaynak 12]

lpr Komutu:

Dökümanları standart yazıcıya gönderir ve sıraya sokar.

Seçenekler:

- #sayı : Yazdırılacak dokümanın kopya adetini belirler.Sayı değişkenini kopya adeti olarak kullanılarak , her bir doküman için istenilen sayıda, çıktının kopyasını sağlar.
- g : Dökümanların belirli standart bilgiler içerdiği belirtilmiş olur.
- j : Görev numarası : mmm olan mesajın standart çıktısının sunulacağını belirtir.Burada mmm görev numarası olarak tahsis edilmiştir.Bu durum sadece işin yerel yazım sırasına sunulduğu zaman gerçekleşir.
- l : Kontrol karakterinin kullanılmasına izin veren filtre (filter) kullanır.
- m : İsmi bildirilen dokümanın, basılmak üzere lp isimli yazıcıya gönderilmesini ve basım tamamlandığında, komutu veren kullanıcıya bir mesaj (mail) gönderilmesini sağlar.
- r : İsmi bildirilen dokümanın basılmak üzere lp isimli yazıcıya gönderilmesini ve dosyanın, ilgili yazıcının sırasına alınmasından hemen sonra diskten silinmesini sağlar.
- s : Komut satırı belirtilmiş dökümanları kopyalamaya çalışmaktansa onları yazdırır.(istenildiği kadar fazla doküman yazdırılabilir.) Bunun anlamı bilgi dokumandan yazdırılana kadar silinemez veya değiştirilemez. Dikkat edilmesi gereken nokta bu tanımlama şekli sadece yerel bilgisayar ağları ve isimlendirilmiş dokümanlar için geçerlidir.
- w sayı : pr komutu sayı değişkenini sayfanın genişliğini ayarlamak için kullanır.
- P Yazıcı_adı : Çıktıyı Yazıcı_adı isimli yazıcıya yönlendirir. Eğer bu tanımlama şekli belirtilmemişse takip eden durumlar meydana gelir:
Default içeriyorsa, lpr komutu default yazıcısını kullanır.
Eğer LPDEST çevre değişkeni kuruluysa lpr, LPDEST değişkeni tarafından belirtilen dosyayı kullanır.Eğer PRINTER değişkeni kurulu ve hiçbir LPDEST değişkeni kurulu değilse, lpr, PRINTER değişkeni tarafından belirtilen değeri kullanır.
- p : Dokümanı formatlamak için pr komutunu kullanır.

Örnekler:

1. `$ lpr index.c program.c example.c`

* index.c program.c example.c isimli dökümanları yazıcı sırasına gönderir.

2. `$ lpr -#3 index.c prgram.c example.c`

* index.c program.c example.c isimli dökümanların herbirinin üçer adet kopyasını yazdırır.

3. `$ lpr -j Myfile`

* Myfile isimli dosyayı sıraya sokar ve görev numarasını geri döndürür.

4. `$ lpr -PHPLazer index.c`

* index.c isimli dosyayı HPLazer isimli yazıcıya yönlendirir.

5. `$ lpr -m index.c`

* index.c isimli dosyayı yazdırılmak üzere yazıcıya gönderir ve işlem tamamlandığında komutu veren kullanıcıya mesaj gönderir.

6. `$ lpr -#4 -PHPLazer index.c`

* index.c isimli dokumanı HPLazer isimli yazıcıya gönderir ve dört tane kopyasının yazılmasını sağlar.

[Kaynak 12]

lprm(line printer remove) Komutu:

Printer'da döküm için bekleyen işleri sıradan çıkarır.

Açıklama:

lprm komutu printer'in işlem kuyruğundan bir veya daha fazla dökümü çıkarır. lprm komutunu,döküm numarasını girmeden , (-) işaretini yazmadan ,yada en azından bir kullanıcı adı girmeden çalıştıramazsınız.

Bir veya bir dizi kullanıcı adı girildiğinde ,lprm komutu işlem sırasında bu kullanıcı adlarına ait olan işlemleri sıradan çıkarır. Özel bir görevin görev sırasını belirterek onu işlem kuyruğundan çıkarabilirsiniz.Bu görev sırası lpq komutu kullanılarak bulunabilir.

Komut Dizimi:

`lprm [-P Printer][görev sırası][Kullanıcı adı...][-]`

Seçenekler:

(-) Bir kullanıcının sahip olduğu bütün döküm işlerini uzaklaştırır.Temel kullanıcı yetkisi olanlar işlem kuyruğundan bütün dökümleri uzaklaştırmak için bu işareti kullanabilirler.

(-P) Printer işlem sırasını özel bir Printer değişkeniyle birlikte belirtir. Eğer bu işaret kullanılmazsa ,aşağıdaki durumlar oluşabilir

Örnekler:

1. %lprm [nnn [mmm . . .]]

Sıra bekleyen dökümler arasında tanıtma numarası nnn(ve mmm vs) olan işleri iptal eder. nnn verilmezse ,komutu veren kullanıcıya ait olan ve o sırada dökülmekte olan ya da sıradaki ilk işi iptal edilir.

2. %lprm kullanıcı_adı

Sıra bekleyen işler arasında sahibi kullanıcı adı olan dökümleri iptal eder .nnn(mmm vs)

3. Lpa isimli printer işlem kuyruğundan 17 numaralı işi çıkarmak için komut
%lprm -p Lpa 17 şeklinde kullanılır.

[Kaynak 12]

cut Komutu:

Bir dosya içerisindeki satırların içerdiği alanları keserek belirli bir yere kopyalamak için kullanılır. Kesme işlemi bazı ayıraçlara göre olabildiği gibi, sabit uzunluktaki alanlara göre de yapılabilir.

Kullanımı:

cut [seçenekler] [dosya] biçiminde tanımlıdır.

Seçenekler:

- b : Karakter konumlarını belirler. Byte tipinde işlem yaparak karakterleri byte birimiyle algılar.
- c : Karakter konumlarını belirler. Byte tipinde işlem yaparak karakterleri karakter olarak algılar.
- f : Özel bir ayıraçla ayrılan alanları tanımlamak amacıyla kullanılır.
- s : -f seçeneği kullanıldığı durumlarda, ayıraca sahip olmayan satırların atlanmasını sağlar.
- n : Tek byte'tan fazla olan karakterleri tamamıyla alır.

Örnekler:

```
$ cat liste
Ahmet KARA:1975:9328
Ali ÖZKAN:1982:634
Pervin ONKUL:1992:1576
```

Sadece isimleri içeren kısmı liste dosyasından kopyalamak için:

```
$ cut -d: -f1 liste  
Ahmet KARA  
Ali ÖZKAN  
Pervin ONKUL
```

[Kaynak 12]