

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

AÇIK KAYNAK İŞLETİM SİSTEMİ YÖNETİMİ

Ankara, 2014

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. KABUK İŞLEMLERİ	3
1.1. Kabuk	3
1.1.1. Kabuk Programı	4
1.2. Komut Satırı ve Programlara Parametre Gönderme	5
1.2.1. Komut Satırı	5
1.2.2. Komut Yapısı	6
1.2.3. Yardım Alma	8
1.2.4. Dosya ve Dizinler	9
1.3. Standart Giriş Çıkış Yönlendirme	10
1.3.1. Standart Giriş Çıkış Komutları	10
1.3.2. Girdi ve Çıktıların Yönlendirilmesi	23
1.4. Temel Filtreleme Komutları	24
1.4.1. Düzenli Deyimler	24
1.4.2. Grep Komutu	25
1.4.2. Cut Komutu	26
1.4.3. Sort Komutu	27
1.5. Dosyaların Taranması	27
1.6. Boru (Pipe) İşlemleri	30
1.7. Temel Kabuk Programlama	30
1.7.1. Kabuk Programlamaya Giriş	30
1.7.2. Değişkenler	31
1.7.3. Değer Okuma	32
1.7.4. Aritmetik İşlemler	33
1.7.5. Kontrol Yapıları	34
1.7.6. Döngüler	37
1.7.7. Kabuk Fonksiyonları	39
UYGULAMA FAALİYETİ	41
ÖLÇME VE DEĞERLENDİRME	43
ÖĞRENME FAALİYETİ-2	45
2. DOSYA VE DİZİN PAYLAŞIMI	45
2.1. Dosya ve Dizin Paylaşımı	45
2.1.1. Dosya ve Dizinlerin Paylaşımına Açılması	45
2.2. Paylaşılan Dosyalara Erişim	48
2.3. Paylaşılan Yazıcılara Erişim	49
2.4. Dosya Paylaşım Programları	52
2.4.1. Apollon	52
2.4.2. KTorrent	55
UYGULAMA FAALİYETİ	58
ÖLÇME VE DEĞERLENDİRME	59
MODÜL DEĞERLENDİRME	60
CEVAP ANAHTARLARI	63
KAYNAKÇA	64

AÇIKLAMALAR

ALAN	Bilişim Teknolojileri
DAL/MESLEK	Dal Ortak
MODÜLÜN ADI	Açık Kaynak İşletim Sistemi Yönetimi
MODÜLÜN TANIMI	Bu modül, açık kaynak işletim sisteminde kabuk işlemlerini, dosya ve izin paylaşımını içeren öğrenme materyalidir.
SÜRE	40/24
ÖNKOŞUL	“Açık Kaynak İşletim Sistemi Kullanımı” modülünü tamamlamış olmak
YETERLİK	Açık kaynak kodlu işletim sisteminin yönetimini gerçekleştirmek
MODÜLÜN AMACI	Genel Amaç Bu modülle gerekli ortam sağlandığında açık kaynak kodlu işletim sisteminin yönetimini yapabileceksiniz. Amaçlar 1. İşletim sisteminde kabuk işlemlerini gerçekleştirebileceksiniz. 2. Dosya ve izin paylaşımı için ağ servislerini kullanabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Ağla birbirine bağlı internet ortamı olan bilgisayar laboratuvarı Donanım: Açık kaynak işletim sistemi yazılımı
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Günlük hayatta gerekli basit bilgisayar işlerinin çoğunu grafik ortamda halletmeniz mümkündür. Fakat bilgisayar alanında çalışan biri için her zaman basit gündelik işler yoktur. Zaman zaman bir ağı veya bir sistemi kurmanız gerekecek ya da bilgisayarda kaybolan önemli bir verinin peşine düşmek zorunda kalacaksınız. Bu işlemler bazen yapılması uzun süren, bir sürü komutun art arda girilmesini gerektiren işlemler olabilir. İlk öğrenme faaliyetinde, açık kaynak işletim sisteminin temel komutlarından bahsedeceğiz. Bu komutları nasıl kullanacağınızı ve gerektiğinde nasıl kabuk programı yazabileceğinizi öğrenecek, böylece tek bir komutla, birçok komutun işlevini yerine getirebileceksiniz.

Artık bilgi çağında olmanın gereği olarak bilgi paylaşımı önem kazanmıştır. Günümüzde bilgi paylaşımı için en önde gelen kaynak, bilgisayarlardır. Bir çalışma ortamında basit bir yerel ağ oluşturmanız birçok sorunu çözecektir. Böylelikle hem zamandan hem de emekten tasarruf edebilirsiniz. İkinci öğrenme faaliyetinde, dosya, izin ve yazıcıları, Pardus kurulu olan ve olmayan makineler arasında nasıl paylaşacağınızı öğreneceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu faaliyet sonunda açık kaynak işletim sisteminde kabuk işlemlerini öğrenecek ve kabuk işlemlerini gerçekleştirebileceksiniz.

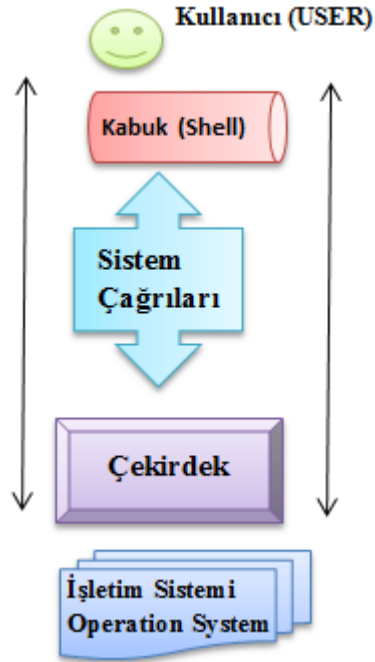
ARAŞTIRMA

- İşletim sistemlerinde kabuk yapısının görevini araştırınız.
- Komut sisteminde çalışmanın sağladığı avantaj ve dezavantajları araştırınız.

1. KABUK İŞLEMLERİ

1.1. Kabuk

Bilgisayar teriminde kabuk kelimesi daha çok çevreleyici, kaplayıcı anlamında kullanılmaktadır. İşletim sisteminde kullanıcı ile donanım arasında üç işletim sistemi katmanı bulunmaktadır.



Şekil 1.1: İşletim sistemi katmanları

Kullanıcı tarafından verilen komutlar kabuk tarafından algılanır ve sistem çağruları yardımı ile çekirdeğe iletilir. Çekirdek bilgisayar donanımı ile etkileşen ve işletim sistemi işlemlerini gerçekleştiren bölümdür.

Bilgisayar ilk açıldığında komut satırı görüntülenir. Kullanıcı tarafından komut satırına girilen bilgiler, bilgisayar tarafından işleme konular. İşletim sistemi ile kullanıcı arasındaki bu etkileşimde kabuk görev alır. Girilen bilginin geçerliliğini inceler, kullanıcının ne yapmak istediğini çözümler ve bu iş için gerekli programları devreye sokar.

Sisteme girilen ilk anda kabuk programı çalıştırılır. Bu andan sonra yapılacak tüm işlemler bu kabuk programı tarafından yönetilir ve denetlenir. Kabuk, klavyeden girilen komutları çalıştırarak bir arabirim görevi yapar. Kabuk sayesinde kullanıcı istediği komutları çalıştırabilir ve sonuçları görebilir.

1.1.1. Kabuk Programı

Kabuk programları işletim sistemi ile kullanıcı arasında köprü kuran programlardır. Kabuk (shell) işlemlerinde genellikle açık kaynak işletim sistemlerinde bulunan gelişmiş komut satırı sistemi kullanılır. Açık kaynak işletim sistemlerinde birçok kabuk programı bulunmaktadır.

- **Sh (Shell):** İlk açık kaynak işletim sistemi kabuk programı
- **Ksh (kornshell):** Sh uyumlu, birçok ek programlama özelliği içeren bir kabuk programı
- **Bash (BourneAgain Shell):** Kolay kullanımı ve etkileşimli özellikleri olan kabuk programı. Bash kabuk programı sh ve ksh uyumluluğunu korurken, özellikle etkileşimli kullanıma yönelik (komut tamamlama gibi) birçok yenilik de içerir.
- **Csh (C Shell):** C programlama diline benzer yapısı bulunan kabuk programı
- **Tcsh:** Csh kabuk programının geliştirilmiş hali

Açık kaynak işletim sistemine kullanıcı adı ve şifre ile giriş yapıldıktan sonra sistem tarafından kullanıcının kabuk programı çalıştırılır. Kullanıcı hesabı açılırken kullanıcının hangi kabuk programını kullanacağı sistem yöneticisi tarafından belirlenir ve *etc/passwd* dosyasında bu bilgi tutulur. Kullanıcı sisteme bağlandığında kullanıcı bilgileri okunurken çalıştıracağı kabuk programı bilgisi de okunur ve çalıştırılır. Kabuk, açık kaynak işletim sistemini komut satırı arabirimi ile yönetebilir. Kabuk, yalnızca verilen komutları yorumlamak ya da yerine getirmenin yanı sıra gelişmiş ve etkileşimli bir programlama ortamıdır. Kullanıcı ve uygulama yönetimi için çeşitli kabuk programları yazılabilir.

1.2. Komut Satırı ve Programlara Parametre Gönderme

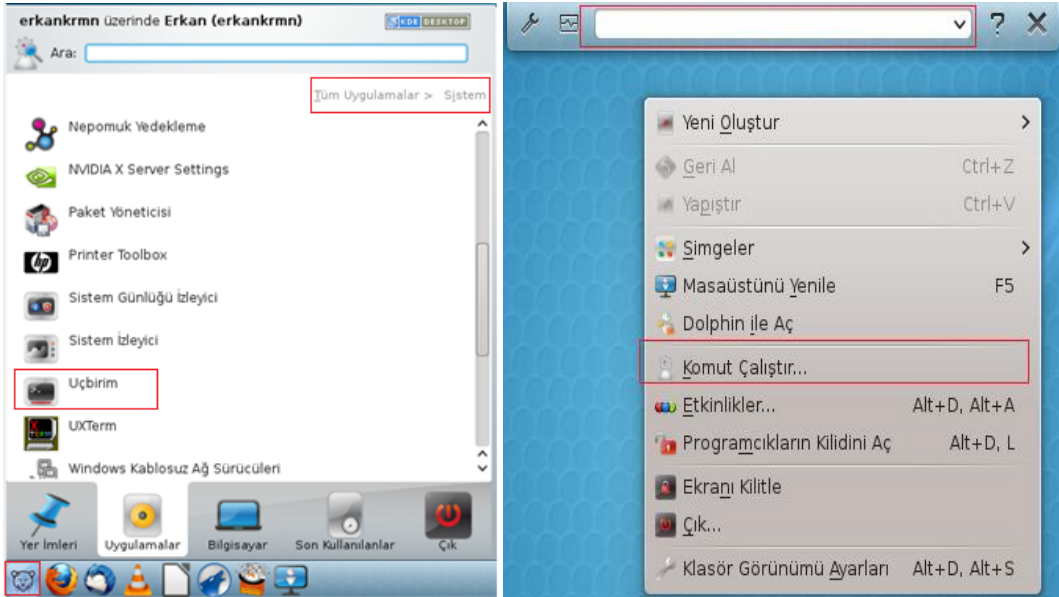
Açık kaynak işletim sisteminde grafiksel ara yüzde yapılacak işlemlerin yanı sıra birçok işlemde komut satırında gerçekleştirilebilir. Bu yöntem diğer işletim sistemlerinde kullanılsa da açık kaynak işletim sisteminin en güçlü yönüdür.

1.2.1. Komut Satırı

Açık kaynak işletim sisteminde kabuk programlama komut satırı uygulamasında gerçekleşir. Açık kaynak işletim sisteminde komut satırı uygulaması, **Konsole (Terminal Programı)** olarak isimlendirilir. Konsole uygulamasını açmak için birkaç yol izlenebilir.

Konsole (Uçbirim – Terminal Program) erişmek için;

- İşletim sisteminin ana menüsünden Uygulamalar → Sistem → Uçbirim (Konsole – Terminal Programı) seçeneği seçilebilir.
- Masaüstüne sağ tıklanır ve Komut Çalıştırılarak ekrana gelen Komut Çalıştır penceresine Konsole yazılarak çalıştırılabilir.
- Alt+F2 klavye kısa yolundan açılan Komut Çalıştır penceresinde “Konsole” yazılarak konsole programı çalıştırılabilir.
- Ctrl+Alt+F1..F6 klavye tuş kombinasyonu ile de 1'den 6'ya kadar ayrı ayrı kabuk oturumu açılması mümkündür. Bu seçenikle, komut sisteminde tam ekran modunda çalıştırılabilir.



Resim 1.1: Komut satırı çalıştırma

Konsole çalıştırıldığında işletim sistemine giriş yapılan kullanıcı hesabı ile işlem yapılabilir. Klavyeden aldığı komutları işlediğinden dolayı konsol “komut satırı arayüzü” (command line interface – CLI) olarak da isimlendirilir.

- Komutlara seçenek verirken seçenektan önce '-' karakteri kullanılır.
 - Örneğin, *ls -l*
Seçenekler genellikle tek tire "-" ya da çift tire "--" işaretinden sonra kullanılır. Seçenekler kısa yazılışlarında (tek harf) "-" uzun yazılışlarında (bir sözcük) "--" kullanılır.
 - Örneğin, *-h --help*: Komutla ilgili yardım bilgisini verir. Argümanlarda ise tire işareti kullanılmaz.
- Kabuk komut satırından verilen komutu çalıştırmadan önce bir dizi karakteri yorumlayarak dosya adına çevirir. Bu karakterler:
 - * 0 dâhil herhangi bir sayıda karakter yerine geçer.
 - o Örneğin *rm ** komutu bütün dosyaları siler,
 - o *ls -la** komutu 'a' ile başlayan dosyaların listesini verir.
 - ? tek bir karakter yerine geçer.
 - o Örneğin ??adı iki karakterden oluşan bütün dosyalar anlamına gelir.
a? yazımı, a harfi ile başlayan 2 karakterli sözcükleri ifade eder.
Burada (?) tek bir karakter yerine geçer: ab, a2, a+
 - [] karakterleri arasında yazılan liste içindeki herhangi bir harfe dönüştürülür.
 - o Örneğin *cp *[abc] /tmp* komutu 'a', 'b' ya da 'c' ile biten bütün dosyaları /tmp dizinine kopyalayacaktır.
Liste içinde aralarına '-' işareti koyarak aralıklar verilebilir.
 - o Örneğin, *[A-Z]** büyük harfle başlayan bütün dosyalar anlamına gelir.
Liste içindeki '^' karakteri sonrasında belirtilen liste dışındaki bütün karakterler anlamına gelir.
 - o Örneğin **[^0-9]** adında rakam olmayan herhangi bir dosya anlamına gelecektir.

Komut satırında komutlar işlenirken ekran çıktısı durdurulabilir, tekrar başlatılabilir, girilen bir komut satırı silinebilir ya da çalışan bir komut durdurulabilir. Bu işlemler klavye tuş kombinasyonları ile gerçekleşir.

Klavye Tuş Kombinasyonu	Etkisi
Ctrl+C	Çalışmakta olan komutu durdurur.
Ctrl+U	Komut satırında bulunan tüm karakterleri siler.
Ctrl+S	Ekrandan geçmekte olan çıktıyı durdurur.
Ctrl+Q	Ctrl+S ile durdurulmuş çıktının tekrar başlatılması sağlanır.
Ctrl+D	Dosya sonu karakteri EOF (End-of-File) ya da "çık" anlamındadır.
Ctrl+W	Komut satırında imlecin bulunduğu yerden bir önceki son kelimeyi siler.
Shift+PageUp/PageDown	Sayfa sayfa yukarı/aşağı

Tablo 1.1: Komut satırı klavye tuş kombinasyonları

1.2.3. Yardım Alma

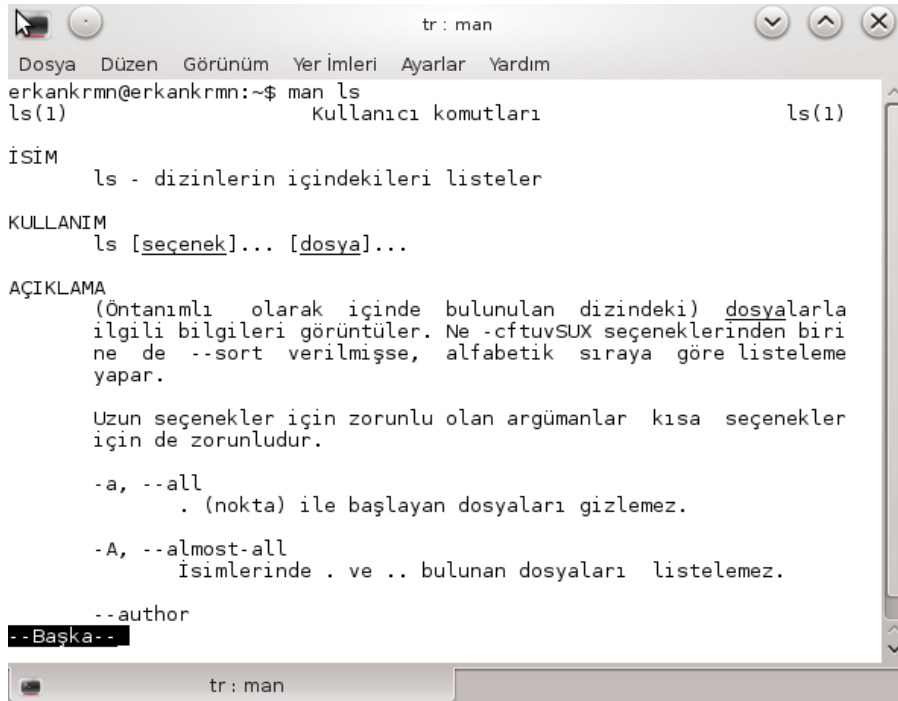
Açık kaynak işletim sisteminde sistem hakkında bilgi vererek kullanıcının sistemi kullanmasını kolaylaştıran komutlar vardır. Bunlarman, help ve info komutlarıdır.

- Man komutu İngilizce “manual” kelimesinin kısaltmasıdır. Bu komut vasıtası ile sistemdeki komutlar hakkında bilgi edinilmektedir.

Kullanım şekli:*man*<komut_adı>

Örneğin komut satırına; *manls* yazıldığında *ls* komutu hakkında kullanıcı, bilgi edinebilir.

- *Man* komutu ekranda bilgileri sayfa sayfa getirir. Geri gitmek için *b* ileri gitmek için *f* tuşu, ileri doğru incelemek için boşluk, çıkmak içinse *q* tuşukullanılır.



```
tr: man
Dosya Düzen Görünüm Yer imleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ man ls
ls(1)                               Kullanıcı komutları          ls(1)

İSİM
  ls - dizinlerin içindekileri listeler

KULLANIM
  ls [seçenek]... [dosya]...

AÇIKLAMA
  (Öntanımlı olarak içinde bulunan dizindeki) dosyalarla ilgili bilgileri görüntüler. Ne -cftuvSUX seçeneklerinden biri ne de --sort verilmişse, alfabetik sıraya göre listeleme yapar.

  Uzun seçenekler için zorunlu olan argümanlar kısa seçenekler için de zorunludur.

  -a, --all
      . (nokta) ile başlayan dosyaları gizlemez.

  -A, --almost-all
      İsimlerinde. ve .. bulunan dosyaları listelemez.

  --author

-- Başka --
```

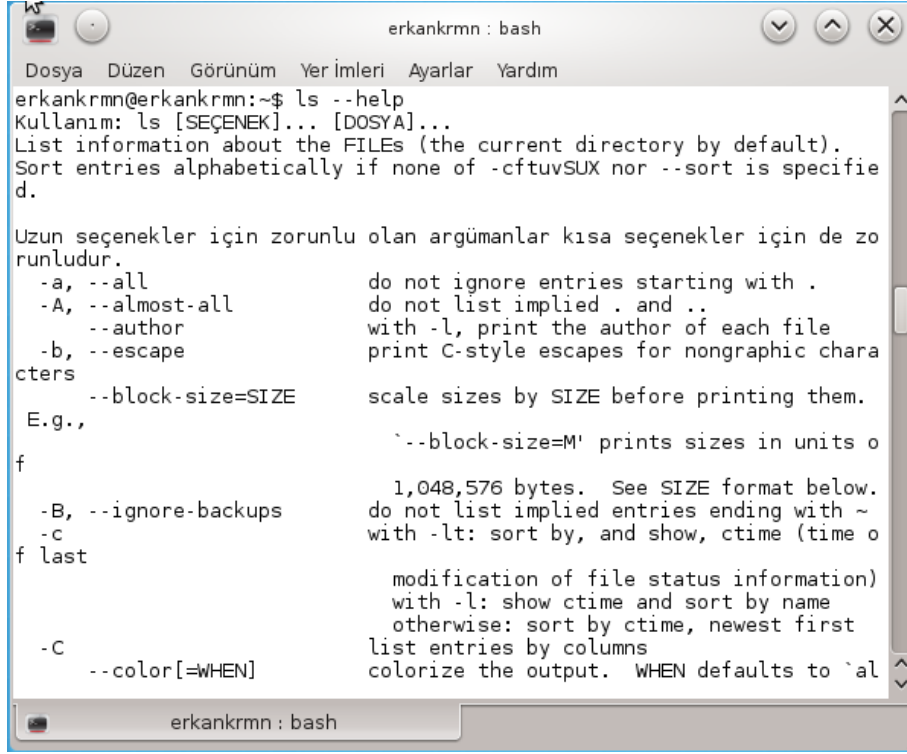
Resim 1.3: Man komutu örnek kullanımı

- *Info* yardım alma komutu da bir komut hakkında bilgi almak için kullanılır.

Kullanım şekli:*info*<komut_adı>

- *Help* komutu ile de komutlar hakkında yardım alınır. Bu komut ile tüm yardım bilgisi listelenir.

Kullanım şekli:<komut_adı>*--help*



```
erkankrnm@erkankrnm:~$ ls --help
Kullanım: ls [SEÇENEK]... [DOSYA]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Uzun seçenekler için zorunlu olan argümanlar kısa seçenekler için de zorunludur.
  -a, --all                do not ignore entries starting with .
  -A, --almost-all       do not list implied . and ..
                        --author          with -l, print the author of each file
  -b, --escape            print C-style escapes for nongraphic characters
                        --block-size=SIZE scale sizes by SIZE before printing them.
E.g.,                    '--block-size=M' prints sizes in units of
                        1,048,576 bytes. See SIZE format below.
  -B, --ignore-backups   do not list implied entries ending with ~
  -c                      with -lt: sort by, and show, ctime (time of
                        last modification of file status information)
                        with -l: show ctime and sort by name
                        otherwise: sort by ctime, newest first
  -C                      list entries by columns
                        --color[=WHEN]   colorize the output. WHEN defaults to 'al
```

Resim 1.4: Help komutu örnek kullanımı

1.2.4. Dosya ve Dizinler

Açık kaynak işletim sisteminde dosya ve izin isimleri 255 karakteri aşamaz. İşletim sisteminde oturum açıldığında önceden tanımlanmış bir dizin içine giriş yapılır. Bu dizin normal kullanıcılar için genellikle */home/* ve ardından gelen kullanıcı dizini ismidir.

Komut satırında dizinleri belirten simgeler bulunur. Bunlar:

- - bir önceki dizini gösterir.
- ~ oturum açan kullanıcının ev dizinini gösterir.
- ~kullanıcı_adıbelirtilen kullanıcının ev dizinini gösterir.
- / kök dizini gösterir.

1.3. Standart Giriş Çıkış Yönlendirme

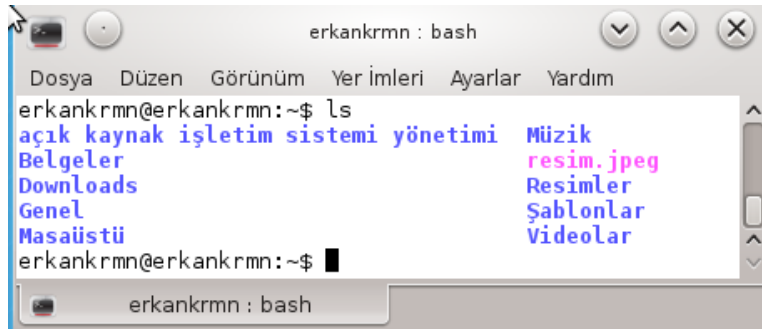
Açık kaynak işletim sisteminde uçbirim (konsol) komutlarının %90 işlevlerini standart giriş biriminden(klavye) okuyacakları veriler üzerinde yerine getirip varsa sonuçlarını standart çıkış(ekran) birimine gönderir.

1.3.1. Standart Giriş Çıkış Komutları

➤ **ls (List Directory Contents) Komutu**

İçerisinde bulunan dizinin dosya ve alt dizinlerini görebilmek için kullanılır. Dosya veya dizin ismi komut satırında belirtilmez ise current directory(.), yani ağaçta bulunan dizini listeler. Listeleme yaparken standart olarak alfabetik sıra ile listeleme işlemi yapılır.

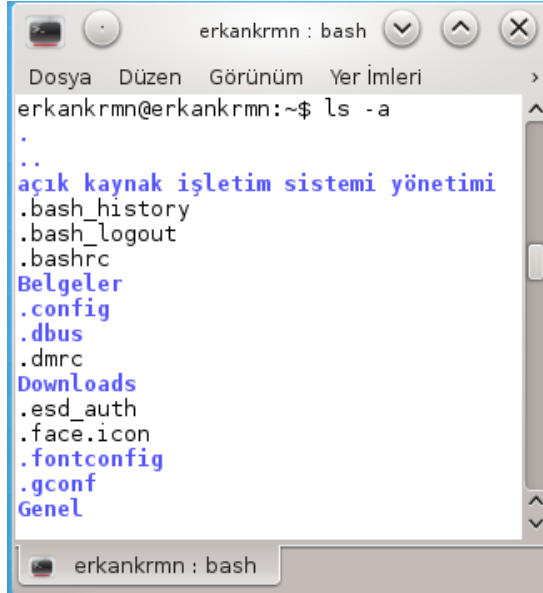
Kullanım şekli: `ls [seçenekler][dosya_veya_dizin_ismi]veya [argüman]`



Resim 1.5: ls komutu örnek kullanımı

ls komutunda bazı seçeneklerde kullanılabilir. Bunlar;

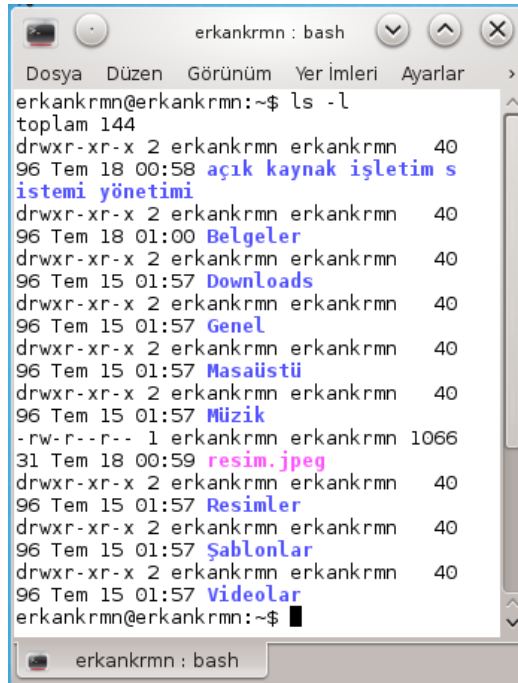
- **-a:** Gizli dosyalar(. nokta ile başlayan) dahil dizinin bütün içeriğini listeler.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri >
erkankrnm@erkankrnm:~$ ls -a
.
..
açık kaynak işletim sistemi yönetimi
.bash_history
.bash_logout
.bashrc
Belgeler
.config
.dbus
.dircache
.dnrc
Downloads
.esd_auth
.face.icon
.fontconfig
.gconf
Genel
erkankrnm : bash
```

Resim 1.6: -a seçeneği örnek kullanımı

- -l: Liste biçiminde listeleme yapar,daha çok ayrıntı verir.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ ls -l
toplam 144
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 18 00:58 açık kaynak işletim s
istemi yönetimi
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 18 01:00 Belgeler
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Downloads
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Genel
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Masaüstü
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Müzik
-rw-r--r-- 1 erkankrnm erkankrnm 1066
31 Tem 18 00:59 resim.jpeg
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Resimler
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Şablonlar
drwxr-xr-x 2 erkankrnm erkankrnm 40
96 Tem 15 01:57 Videolar
erkankrnm@erkankrnm:~$ █
erkankrnm : bash
```

Resim 1.7: -l seçeneği örnek kullanımı

- -h: Dosya boyutlarını okunabilir formatta(human readable) gösterir. Byte olarak değil KB,MB cinsinden gösterir.

- **-R:** Ağaç yapısında bulunan dizinden itibaren altlara doğru listeleme yapar.

```

erkankrnm@erkankrnm:~$ ls -R
.:
açık kaynak işletim sistemi yönetimi Müzik
B Belgeler resim.jp
Downloads Resimler
Genel Şablonlar
Masaüstü Videolar

./açık kaynak işletim sistemi yönetimi:

./B Belgeler:

./Downloads:

./Genel:

./Masaüstü:
Home.desktop trash.desktop

./Müzik:

./Resimler:

./Şablonlar:

./Videolar:
erkankrnm@erkankrnm:~$

```

Resim 1.8: -R seçeneği örnek kullanımı

Örnek: Gizli dosyaları ayrıntılı bir şekilde listelemek için **-a** ve **-l** seçeneği birlikte kullanılır. Herhangi bir argüman kullanılmayacaksa seçenekler birleştirilerek kullanılır.

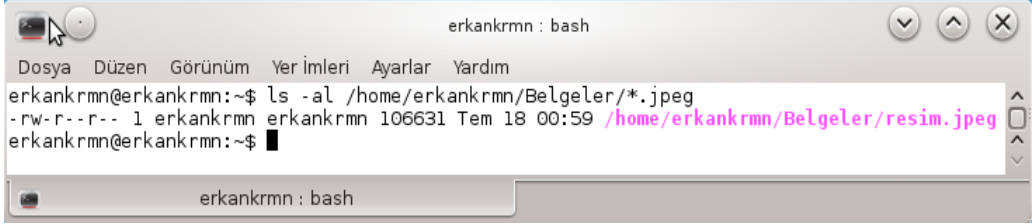
```

erkankrnm@erkankrnm:~$ ls -al
toplam 284
drwxr-xr-x 25 erkankrnm erkankrnm 4096 Tem 18 01:00 .
drwxr-xr-x 3 root root 4096 Tem 15 01:46 ..
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 18 00:58 açık kaynak işletim
sistemi yönetimi
-rw-r----- 1 erkankrnm erkankrnm 122 Tem 18 01:25 .bash_history
-rw-r----- 1 erkankrnm erkankrnm 220 Oca 1 2013 .bash_logout
-rw-r----- 1 erkankrnm erkankrnm 3206 Eki 10 2012 .bashrc
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 18 01:00 Belgeler
drwxr-xr-x 3 erkankrnm erkankrnm 4096 Tem 15 01:57 .config
drwxr-xr-x 3 erkankrnm erkankrnm 4096 Tem 15 01:57 .dbus
-rw-r----- 1 erkankrnm erkankrnm 26 Tem 15 01:57 .dmrc
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 Downloads
-rw-r----- 1 erkankrnm erkankrnm 16 Tem 15 01:57 .esd_auth
-rw-r----- 1 erkankrnm erkankrnm 4143 Tem 16 01:36 .face.icon
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 .fontconfig
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 18 00:57 .gconf
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 Genel
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 .gnome2
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 .gststreamer-0.10
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 15 01:57 .gvfs

```

Resim 1.9: -a ve -l seçenekleri birlikte kullanımı

- **Örnek:**ls komutu ile seçenekler ve argümanlar birlikte kullanılabilir. Kullanıcının belgeler klasöründeki resim dosyalarını listelemek için; ls -al /home/erkankrnm/belgelerim/*.jpeg komutu kullanılır. Bu komut ile dosya uzantısı jpeg olan tüm dosyalar listelenir.



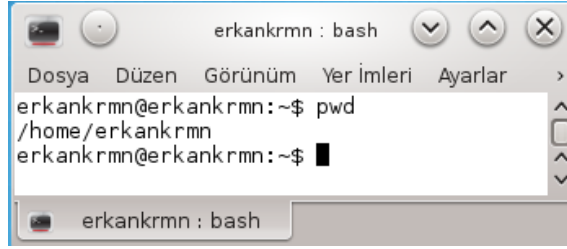
```

erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ ls -al /home/erkankrnm/Belgeler/*.jpeg
-rw-r--r-- 1 erkankrnm erkankrnm 106631 Tem 18 00:59 /home/erkankrnm/Belgeler/resim.jpeg
erkankrnm@erkankrnm:~$ █

```

Resim 1.10: Ls komutu argüman ile örnek kullanım

- **Pwd (print working directory) komutu:** Dosya sisteminde bulunan dizini öğrenmek için kullanılır.



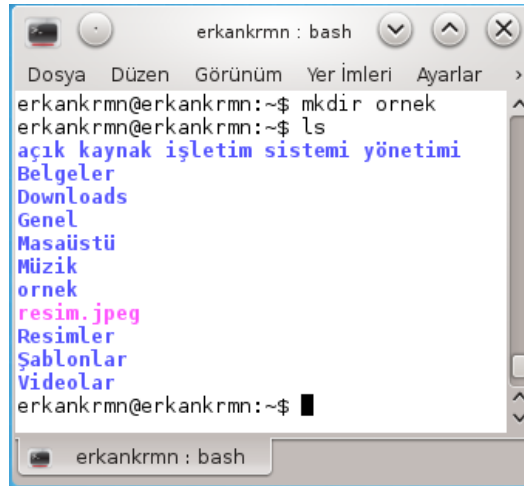
```

erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ pwd
/home/erkankrnm
erkankrnm@erkankrnm:~$ █

```

Resim 1.11: Pwd komutu örnek kullanımı

- **Mkdir (make directory) komutu:** Dizin oluşturmak için kullanılır.



```

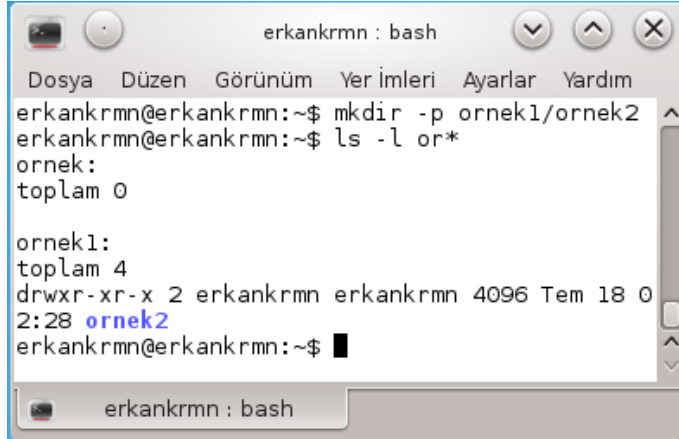
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ mkdir ornek
erkankrnm@erkankrnm:~$ ls
açık kaynak işletim sistemi yönetimi
Belgeler
Downloads
Genel
Masüstü
Müzik
ornek
resim.jpeg
Resimler
Şablonlar
Videolar
erkankrnm@erkankrnm:~$ █

```

Resim 1.12: Mkdir komutu örnek kullanımı

Mkdir komutunda bazı seçenekler de kullanılabilir. Bunlar;

- **-p:** İç içe dizin oluşturmak için kullanılır.

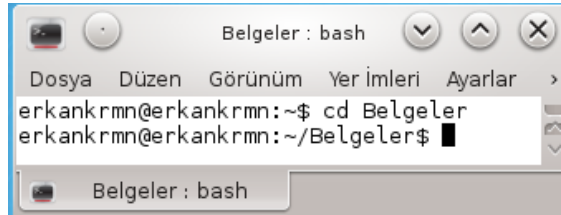


```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ mkdir -p ornek1/ornek2
erkankrnm@erkankrnm:~$ ls -l or*
ornek:
toplam 0

ornek1:
toplam 4
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 18 0
2:28 ornek2
erkankrnm@erkankrnm:~$
```

Resim 1.13: -p seçeneği örnek kullanımı

- **Cd (change directory) komutu:** Bulunulan dizinden başka bir dizine geçmek için kullanılır. Dizinler arasında geçiş işlemleri yapılabilir.

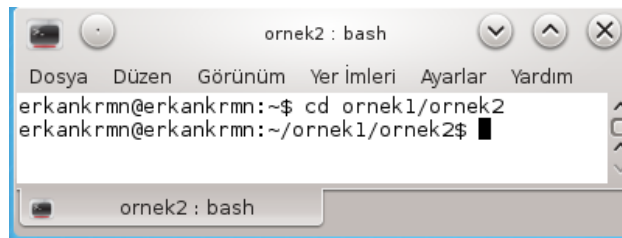


```
Belgeler : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ cd Belgeler
erkankrnm@erkankrnm:~/Belgeler$
```

Resim 1.14: Cd komutu örnek kullanımı

Cd komutu seçenek ve argümanlar ile kullanılabilir.

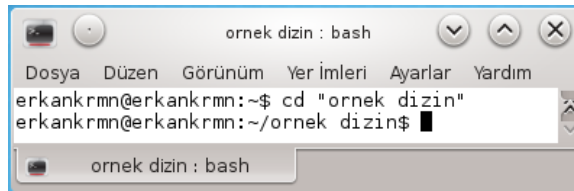
- Cd komutubir nokta ile kullanırsa içinde olunan dizini gösterir.
- İç içe dizinlere girmek için dizin yolu cd komutundan sonra yazılır.



```
ornek2 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cd ornek1/ornek2
erkankrnm@erkankrnm:~/ornek1/ornek2$
```

Resim 1.15: İç içe dizin içerisine girme

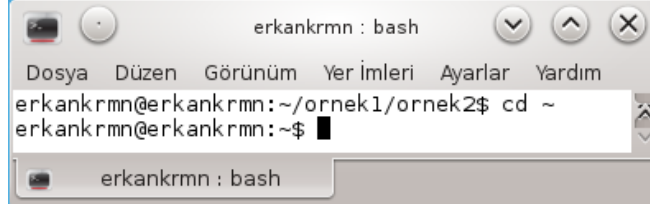
- Bir dizin isminde boşluk var ise çift tırnak argümanı kullanılır.



```
ornek dizin : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cd "ornek dizin"
erkankrnm@erkankrnm:~/ornek dizin$
```

Resim 1.16: Cd komutu ile argüman kullanımı

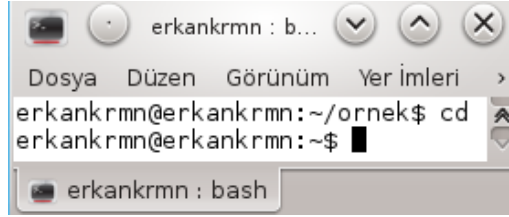
- Ev dizinine dönmek için ~ işareti kullanılır.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek1/ornek2$ cd ~
erkankrnm@erkankrnm:~$
```

Resim 1.17: Cd komutu ile ev dizine dönme

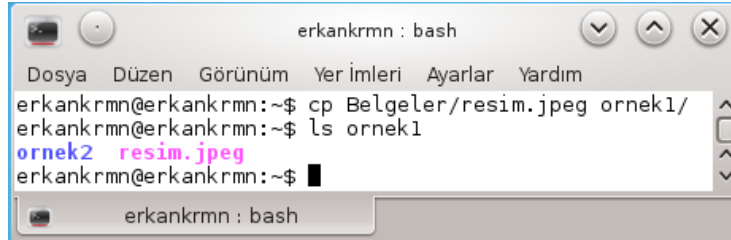
- Cd komutundan sonra bir boşluk kullanılırsa bir önceki dizine dönülür.



```
erkankrnm : b...
Dosya Düzen Görünüm Yer İmleri >
erkankrnm@erkankrnm:~/ornek$ cd
erkankrnm@erkankrnm:~$
```

Resim 1.18: Cd komutu bir önceki dizine girme

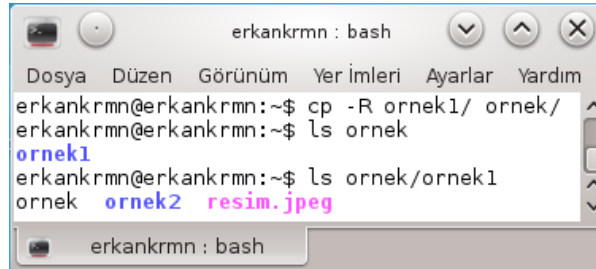
- **Cp (copy) komutu** : Dosya veya dizinleri kopyalamak için kullanılır. Kullanımı: *cp [Kaynak] [Hedef]*



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cp Belgeler/resim.jpeg ornek1/
erkankrnm@erkankrnm:~$ ls ornek1
ornek2  resim.jpeg
erkankrnm@erkankrnm:~$
```

Resim 1.19: Copy komutu örnek kullanımı

- **-R**: seçeneği ile kaynak içerisindeki dosya ve dizinler ile birlikte hedefe kopyalanır.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cp -R ornek1/ ornek/
erkankrnm@erkankrnm:~$ ls ornek
ornek1
erkankrnm@erkankrnm:~$ ls ornek/ornek1
ornek  ornek2  resim.jpeg
erkankrnm@erkankrnm:~$
```

Resim 1.20: -R seçeneği örnek kullanımı

- **Mv (move)**: Dosya ve dizinleri taşımak için kullanılır. Aynı zamanda dosya veya dizinlerin isimlerini değiştirmek için de kullanılabilir.

Kullanımı: *mv [kaynak][hedef]*

```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ mv resim.jpeg ornek
erkankrnm@erkankrnm:~$ ls ornek
ornek1 resim.jpeg
erkankrnm@erkankrnm:~$
```

Resim 1.21: Mv komutu örnek kullanımı

Resim 1.21’de görüldüğü gibi ev dizinindeki *resim.jpeg* dosyasını örnek dizine taşıma işlemi yapılmıştır.

```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ mv resim.jpeg ornek.jpeg
erkankrnm@erkankrnm:~/ornek$ ls
ornek1 ornek.jpeg
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.22: Mv komutu ile isim değiştirme işlemi

Resim 1.22’de *resim.jpeg* dosya ismi *ornek.jpeg* olarak değiştirilmiştir.

```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ mv dizin ornek
erkankrnm@erkankrnm:~$ ls ornek
dizin ornek1 ornek.jpeg
erkankrnm@erkankrnm:~$
```

Resim 1.23: Mv komutu dizin taşıma işlemi

- **ln (link) komutu:** Dosyalara veya dizinlere farklı yerlerden ulaşabilmek amacıyla dosyalara link (bağ) tanımlayan komuttur.

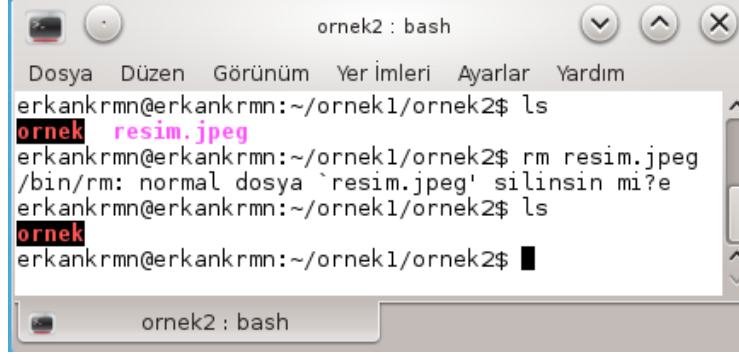
Kullanımı: *ln [seçenekler] hedef link_ismi*

```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ ln ornek/ornek.jpeg link.jpeg
erkankrnm@erkankrnm:~$ ls
açık kaynak işletim sistemi yönetimi ornek
Belgeler ornek1
Downloads ornek dizin
Genel Resimler
link.jpeg Şablonlar
Masasıüstü Videolar
Müzik
erkankrnm@erkankrnm:~$
```

Resim 1.24: ln komutu örnek kullanımı

Resim 1.24'te *ornek* dizini içerisindeki *ornek.jpeg* dosyasının ev dizini içerisinde *link.jpeg* isminde bir bağlantı oluşturulmuştur. Her iki dosyanın birinde yapılan değişiklikler diğer dosyada da aynen olmaktadır.

- **Clear komutu:** Komut satırını temizleyerek imleci ilk satıra taşır.
- **Rm (remove) komutu:** Dosya veya dizinleri silmek için kullanılır.

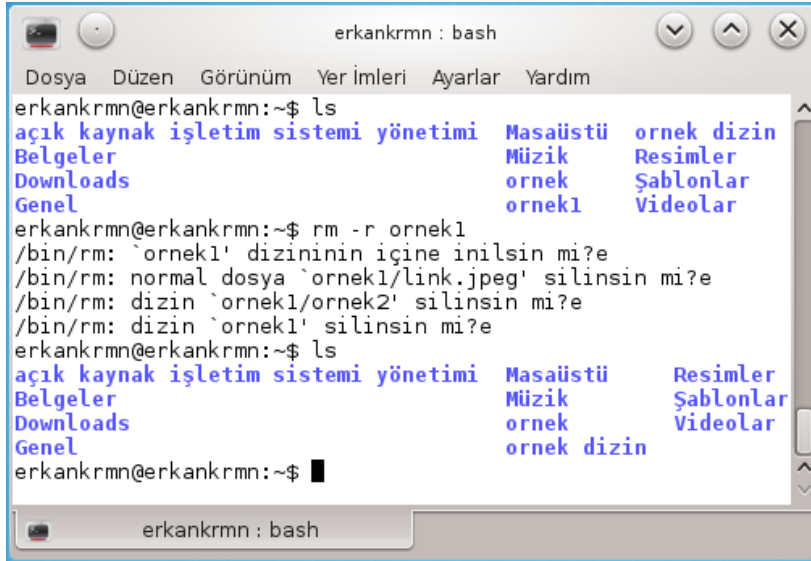


```
ornek2 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrmn@erkankrmn:~/ornek1/ornek2$ ls
ornek  resim.jpeg
erkankrmn@erkankrmn:~/ornek1/ornek2$ rm resim.jpeg
/bin/rm: normal dosya `resim.jpeg' silinsin mi?e
erkankrmn@erkankrmn:~/ornek1/ornek2$ ls
ornek
erkankrmn@erkankrmn:~/ornek1/ornek2$ █
```

Resim 1.25: Rm komutu kullanımı

Resim 1.25'te rm komutu ile dosya silme işlemi yapılmıştır. Silme işleminde kullanıcıdan dosya silme onayı istenmektedir. Rm komutu seçenekleri ile de kullanılabilir.

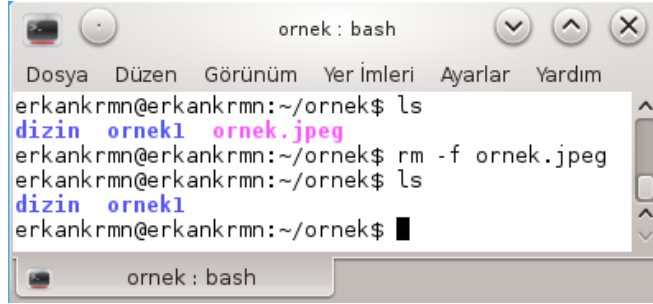
- **-r:** Dizin silme işlemi yaparken dizinin içerisi dosya veya dizinler ile dolu ise **-r** seçeneği ile kullanıcıdan onay alınarak silme işlemi yapılır.



```
erkankrmn : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrmn@erkankrmn:~$ ls
açık kaynak işletim sistemi yönetimi Masaüstü ornek dizin
Belgeler Müzik Resimler
Downloads ornek Şablonlar
Genel ornek1 Videolar
erkankrmn@erkankrmn:~$ rm -r ornek1
/bin/rm: `ornek1' dizininin içine inilsin mi?e
/bin/rm: normal dosya `ornek1/link.jpeg' silinsin mi?e
/bin/rm: dizin `ornek1/ornek2' silinsin mi?e
/bin/rm: dizin `ornek1' silinsin mi?e
erkankrmn@erkankrmn:~$ ls
açık kaynak işletim sistemi yönetimi Masaüstü Resimler
Belgeler Müzik Şablonlar
Downloads ornek Videolar
Genel ornek dizin
erkankrmn@erkankrmn:~$ █
```

Resim 1.26: Rm komutu -r seçeneği örnek kullanımı

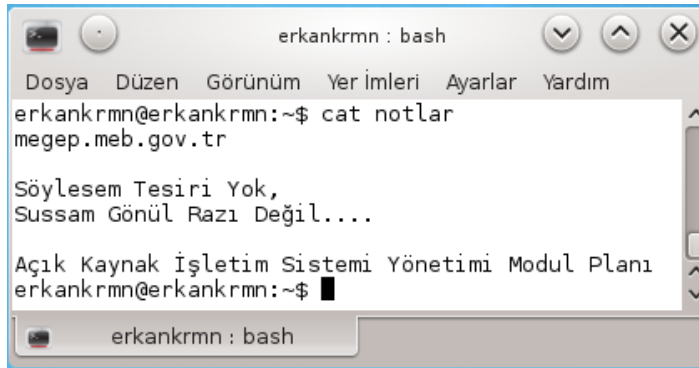
- **-f:** Dosya silme işlemi yapılırken kullanıcı onayı almaksızın dosyaları silmek için kullanılır.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ ls
dizin ornek1 ornek.jpeg
erkankrnm@erkankrnm:~/ornek$ rm -f ornek.jpeg
erkankrnm@erkankrnm:~/ornek$ ls
dizin ornek1
erkankrnm@erkankrnm:~/ornek$ █
```

Resim 1.27: Rm komutu -f seçeneği kullanımı

- **Cat (concatenate files) komutu:** Bir dosyanın içeriğini liste halinde görüntülemek için kullanılır.



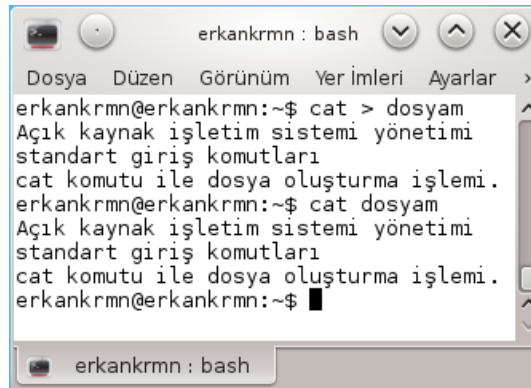
```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cat notlar
megep.meb.gov.tr

Söylesem Tesiri Yok,
Sussam Gönül Razi Değil....

Açık Kaynak İşletim Sistemi Yönetimi Modul Planı
erkankrnm@erkankrnm:~$ █
```

Resim 1.28: Cat komutu örnek kullanımı

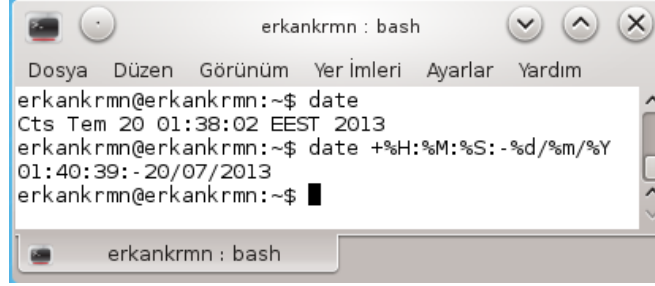
> seçeneği ile yeni bir dosya oluşturularak içine bilgi girişi yapılır. İmleç yeni satırın başına geldiğinde, dosyanın içinde yer alması istenen satırlara girilir. Girilmek istenen satırlar tamamlandıktan sonra imleç satır başındayken Ctrl + D tuşlarına birlikte basılarak standart giriş biriminde dosya sonuna geldiği belirtilir.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar >
erkankrnm@erkankrnm:~$ cat > dosyam
Açık kaynak işletim sistemi yönetimi
standart giriş komutları
cat komutu ile dosya oluşturma işlemi.
erkankrnm@erkankrnm:~$ cat dosyam
Açık kaynak işletim sistemi yönetimi
standart giriş komutları
cat komutu ile dosya oluşturma işlemi.
erkankrnm@erkankrnm:~$ █
```

Resim 1.29: Cat komutu ile dosya oluşturma ve veri girişi

- **Date komutu:** Sistem saat ve tarihini görüntülemek için kullanılır. Sistem yöneticisi olarak oturum açılarak sistem saat ve tarihi değiştirilebilir.



```

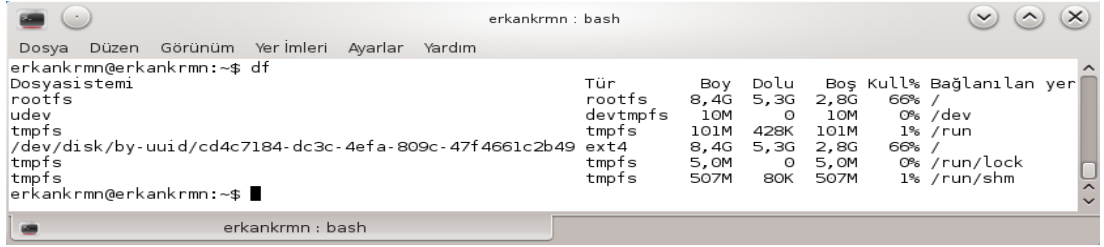
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ date
Cts Tem 20 01:38:02 EEST 2013
erkankrnm@erkankrnm:~$ date +%H:%M:%S:-%d/%m/%Y
01:40:39: -20/07/2013
erkankrnm@erkankrnm:~$ █

```

Resim 1.30: Date komutu örnek kullanımı

Resim 1.30'da sistem saat ve tarihi gösterimi ile düzenli gösterimi kullanılmıştır.

- **Df (display file system) komutu:** Bilgisayar, disk üzerindeki mevcut kapasite ve boş alan miktarını öğrenmek için kullanılır.



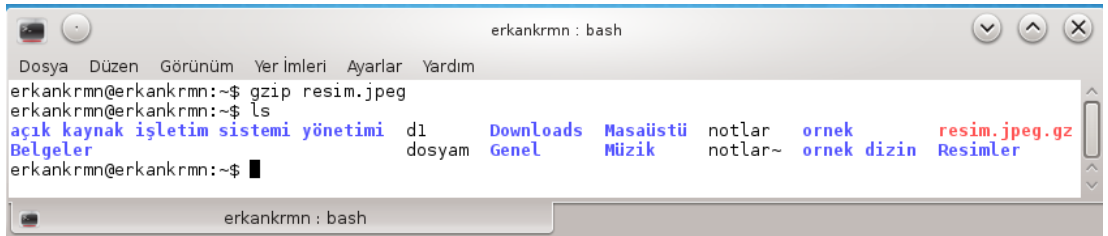
```

erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ df
Dosyasistemi          Tür          Boy  Dolu  Boş  Kull%  Bağlanılan yer
rootfs                rootfs      8,4G  5,3G  2,8G   66%  /
udev                  devtmpfs    10M      0  10M    0%  /dev
tmpfs                 tmpfs      101M  428K  101M    1%  /run
/dev/disk/by-uuid/cd4c7184-dc3c-4efa-809c-47f4661c2b49 ext4        8,4G  5,3G  2,8G   66%  /
tmpfs                 tmpfs        5,0M      0  5,0M    0%  /run/lock
tmpfs                 tmpfs       507M   80K  507M    1%  /run/shm
erkankrnm@erkankrnm:~$ █

```

Resim 1.31: Df komutu örnek kullanımı

- **Gzip komutu:** Dosyaları veya dizinlerisıkıştırmak ve arşivlemek için kullanılır.



```

erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ gzip resim.jpeg
erkankrnm@erkankrnm:~$ ls
açık kaynak işletim sistemi yönetimi d1 Downloads Masaüstü notlar ornek resim.jpeg.gz
Belgeler dosyam Genel Müzik notlar~ ornek dizin Resimler
erkankrnm@erkankrnm:~$ █

```

Resim 1.32: Gzip komutu örnek kullanımı

- **-d** seçeneği ile sıkıştırılmış olan dosyaları veya dizinleri açmak için kullanılır.

```

erkankrnm@erkankrnm:~$ gzip -d resim.jpeg.gz
erkankrnm@erkankrnm:~$ ls
açık kaynak işletim sistemi yönetimi  dosyam      Masaüstü   notlar~    resim.jpeg  Videolar
Belgeler                               Downloads  Müzik     ornek     Resimler
d1                                      Genel      notlar    ornek dizin Şablonlar
erkankrnm@erkankrnm:~$

```

Resim 1.33: Gzip komutu -d seçeneği örnek kullanımı

- **-r** seçeneği, dizin adı ile kullanıldığında dizinin tüm alt dizinlerini inceler ve bulunduğu tüm dosyaları ayrı ayrı sıkıştırır.

```

ornek1 : bash
erkankrnm@erkankrnm:~$ gzip -r ornek1
erkankrnm@erkankrnm:~$ ls
açık kaynak işletim sistemi yönetimi  Downloads  Müzik     resim.jpeg  Videolar
Belgeler                               Genel      ornek     Resimler
dosyam                                  Masaüstü   ornek1    Şablonlar
erkankrnm@erkankrnm:~$ cd ornek1
erkankrnm@erkankrnm:~/ornek1$ ls
ornek2  ornek.gz  resim.jpeg.gz
erkankrnm@erkankrnm:~/ornek1$

```

Resim 1.34: Gzip komutu -r seçeneği örnek kullanımı

- **Gunzip komutu:** Sıkıştırma işlemi uygulanan dosyaları, dizinleri açmak için kullanılır.

```

ornek1 : bash
erkankrnm@erkankrnm:~/ornek1$ ls
ornek2  ornek.gz  resim.jpeg.gz
erkankrnm@erkankrnm:~/ornek1$ gunzip ornek.gz
erkankrnm@erkankrnm:~/ornek1$ ls
ornek  ornek2  resim.jpeg.gz
erkankrnm@erkankrnm:~/ornek1$

```

Resim 1.35: Gunzip komutu örnek kullanımı

- **Tar komutu:** Sistemde bulunan dosyaların veya dizinlerin birarada paketlenmesini yapıp arşiv oluşturmayı ve var olan bir arşivden dosyaların geri alınmasını sağlar. Oluşturulan arşivlerin uzantısı “.tar” şeklindedir. Genellikle oluşturulan arşivler yer kazanmak amacı ile sıkıştırılıp saklanmaktadır. Kullanımı: Tar [seçenekler] arşiv_ismi arşiv_yapılacak_dosyalar Tar komutu seçenekleri ile birlikte kullanılır. Bunlar:
 - **-c (create):** Tar uzantılı bir arşiv dosyası oluşturulacağını belirtir.
 - **-x (extract):** Tar uzantılı arşiv dosyasının açılacağını belirtir.

- **-t (tabel of contents):**Tar uzantılı arşiv dosyasının içeriğini görüntülemek için kullanılır.
- **-v (verbose):** Tar uzantılı bir arşiv dosyası oluşturulurken veya açılırken arşiv içindeki dosyaların isimlerini ekrana listeler.
- **-f (file):** Tar uzantılı arşiv dosyası oluşturulurken, açılırken veya içindekiler listelenirken tar isimli dosyanın komut satırında kullanılacağını belirtir.
- **-z:** Tar uzantılı dosyanın gzip veya gunzip komutları ile kullanılabileceğini belirtir.

```

ornek1 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ cd ornek1
erkankrnm@erkankrnm:~/ornek1$ ls
ornek ornek2 resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ tar -cf ornek.tar ornek resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ ls
ornek ornek2 ornek.tar resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ █

```

Resim 1.36: Tar komutu ile arşivleme

```

ornek1 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek1$ ls
ornek2 ornek.tar
erkankrnm@erkankrnm:~/ornek1$ tar -xf ornek.tar
erkankrnm@erkankrnm:~/ornek1$ ls
ornek ornek2 ornek.tar resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ █

```

Resim 1.37: Tar komutu ile arşiv dosyasını açmak

```

ornek1 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek1$ ls
ornek ornek2 ornek.tar resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ tar -czf ornek.tar.gz ornek
erkankrnm@erkankrnm:~/ornek1$ ls
ornek ornek2 ornek.tar ornek.tar.gz resim.jpeg
erkankrnm@erkankrnm:~/ornek1$ █

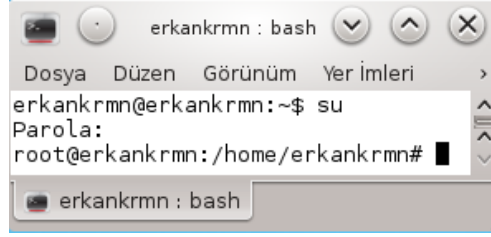
```

Resim 1.38: Tar komutu ile dizin arşivleme

- **Su (switch user) komutu:** İşletim sisteminde kullanıcı değiştirmek için kullanılır.

Kullanımı *su [kullanıcı_adi]*

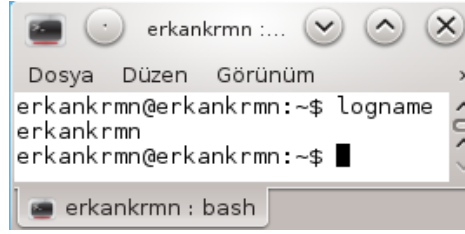
Herhangi bir kullanıcı adı belirtilmediğinde sistem yöneticisi (root) olarak oturum açılır.**Exit** komutu ile açılan oturumdan çıkılır.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri >
erkankrnm@erkankrnm:~$ su
Parola:
root@erkankrnm: /home/erkankrnm#
erkankrnm : bash
```

Resim 1.39: Su komutu örnek kullanımı

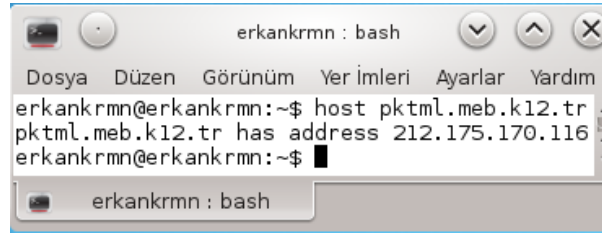
- **Logname komutu:** İşletim sisteminde oturum açan kullanıcıyı gösterir.



```
erkankrnm : ...
Dosya Düzen Görünüm >
erkankrnm@erkankrnm:~$ logname
erkankrnm
erkankrnm@erkankrnm:~$ █
erkankrnm : bash
```

Resim 1.40: Logname komutu örnek kullanımı

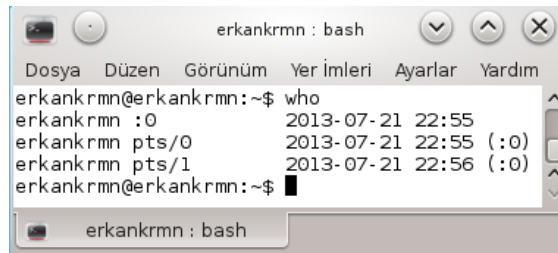
- **Host komutu:** Girilen internet adresinin(Domain Name) IP karşılığını verir.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım >
erkankrnm@erkankrnm:~$ host pktml.meb.k12.tr
pktml.meb.k12.tr has address 212.175.170.116
erkankrnm@erkankrnm:~$ █
erkankrnm : bash
```

Resim 1.41: Host komutu örnek kullanımı

- **Who komutu:** İşletim sisteminde kayıtlı olan kullanıcıların oturum açma saat ve tarihi listelemek için kullanılır.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım >
erkankrnm@erkankrnm:~$ who
erkankrnm :0          2013-07-21 22:55
erkankrnm pts/0      2013-07-21 22:55 (:0)
erkankrnm pts/1      2013-07-21 22:56 (:0)
erkankrnm@erkankrnm:~$ █
erkankrnm : bash
```

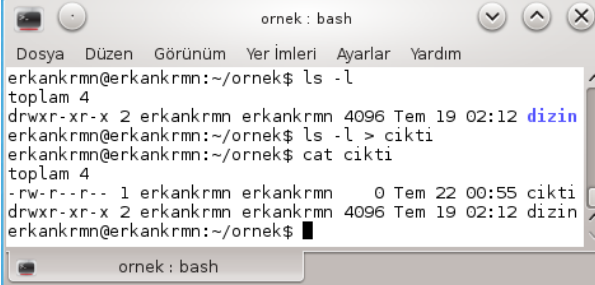
Resim 1.42: Who komutu örnek kullanımı

1.3.2. Girdi ve Çıktıların Yönlendirilmesi

Açık kaynak işletim sisteminde komutların en büyük özelliklerinden biri girdilerin veya çıktıların bir dosyaya yönlendirilmesi ya da başka bir birime yönlendirilmesidir. Komut girdilerinin standart giriş biriminin, klavyeden değil de başka birimden alınması da komut sisteminin özelliklerindedir.

Standart çıktıları yönlendirmek için

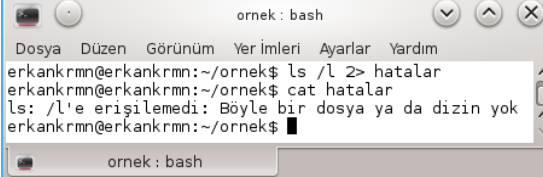
- > operatörü çıktıyı bir dosyaya yönlendirir ancak hatalar ekranda görüntülenmeye devam eder. Bu operatör dosyanın içeriğini temizleyerek çıktıyı dosyaya ekler.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ ls -l
toplam 4
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 19 02:12 dizin
erkankrnm@erkankrnm:~/ornek$ ls -l > cikti
erkankrnm@erkankrnm:~/ornek$ cat cikti
toplam 4
-rw-r--r-- 1 erkankrnm erkankrnm 0 Tem 22 00:55 cikti
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 19 02:12 dizin
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.43: Standart çıktıyı bir dosyaya yönlendirme

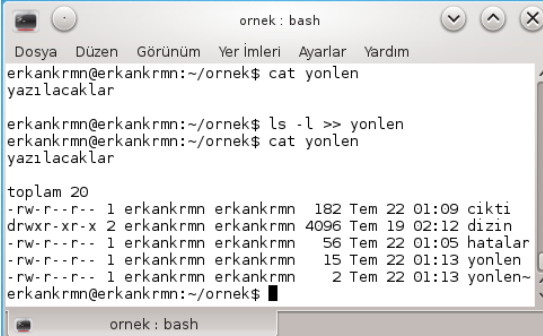
- 2> operatörü hataları dosyanın sonuna ekler.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ ls /l 2> hatalar
erkankrnm@erkankrnm:~/ornek$ cat hatalar
ls: /l'e erişilemedi: Böyle bir dosya ya da dizin yok
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.44: Hataları yönlendirme

- >>standart çıktıyı dosyanın içeriği silinmeden dosyanın sonuna ekler.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ cat yonlen
yazılacaklar
erkankrnm@erkankrnm:~/ornek$ ls -l >> yonlen
erkankrnm@erkankrnm:~/ornek$ cat yonlen
yazılacaklar
toplam 20
-rw-r--r-- 1 erkankrnm erkankrnm 182 Tem 22 01:09 cikti
drwxr-xr-x 2 erkankrnm erkankrnm 4096 Tem 19 02:12 dizin
-rw-r--r-- 1 erkankrnm erkankrnm 56 Tem 22 01:05 hatalar
-rw-r--r-- 1 erkankrnm erkankrnm 15 Tem 22 01:13 yonlen
-rw-r--r-- 1 erkankrnm erkankrnm 2 Tem 22 01:13 yonlen~
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.45: Standart çıktıyı dosya sonuna ekleme

- <<girdi birimini değiştirmek için kullanılır.

1.4. Temel Filtreleme Komutları

1.4.1. Düzenli Deyimler

Düzenli deyimler, metni düzenlemek veya metin içerisine belirli kurallarla alt metinler eklemek için kullanılır. Açık kaynak işletim sisteminde düzenli deyimler kullanılarak filtreleme işlemleri gerçekleştirir.

Bu düzenli deyimler:

Sembol	Anlamı
.	Herhangi tek bir karaktere karşılık gelir.
*	Kendisinden önceki karakterin sıfır veya daha fazla kez olduğunu gösterir.
[]	Küme içerisindeki karakterlerden herhangi birine karşılık gelir.
[^]	Küme içerisindeki karakterlerin dışında herhangi bir karaktere karşılık gelir.
^	Satır başını ifade eder.
\$	Satır sonunu ifade eder.
\+	Kendisinden önceki karakterin 1 ya da daha fazla olduğunu gösterir.
\?	Kendisinden önceki karakterin 0 ya da 1 kez bulunduğunu gösterir.
\	Kendisinden bir önceki veya bir sonraki karaktere karşılık gelir.
\ (\)	Grup olarak düzenli deyimleri tanımlar.
\	Özel karakterlerin normal karakter olarak algılanmasını sağlar.

Tablo 1.2: Düzenli deyimlerde kullanılan semboller

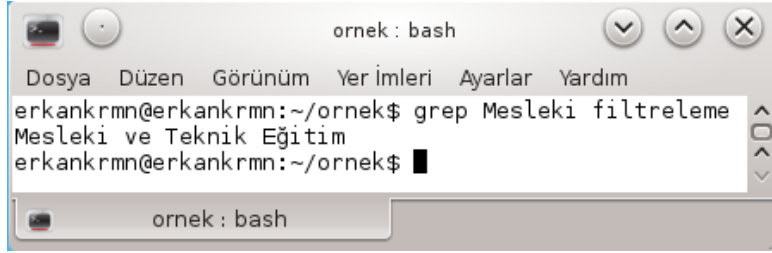
Düzenli deyimler için aşağıda kullanım örnekleri verilmiştir.

- `^a`: Düzenli deyim yalnızca "a" ile başlayan satırları bulacaktır.
- `a$`: Düzenli deyim yalnızca satır sonlarında "a" harfinin bulunduğu satırları bulacaktır.
- `^$`: Deyimi, başı sonu bir olan, boş satırları bulmakta kullanışlıdır.
- `se[vz]gi`: Düzenli deyim "s, peşinden e, peşinden bir v ya da z, peşinden g, peşinden i" şeklinde olan kelimeleri bulmak için kullanılır.
- `^k{2,}`: Satır başında en az 2 adet k karakteri bulunan sözcükler için kullanılır.
- `\(3\|5\)+`: Düzenli deyim ile içerisinde 3 veya 5 sayılarından en az 1 kez geçen sözcükleri bulmak için kullanılır.
- `^K*M$`: Düzenli deyim ile satır başında 'K' karakteri ile başlayıp, satır sonunda 'M' ile biten sözcükleri bulmak için kullanılır.

1.4.2. Grep Komutu

Bir dosya içerisinde bir metni arayıp ekrana listeleyen komuttur.

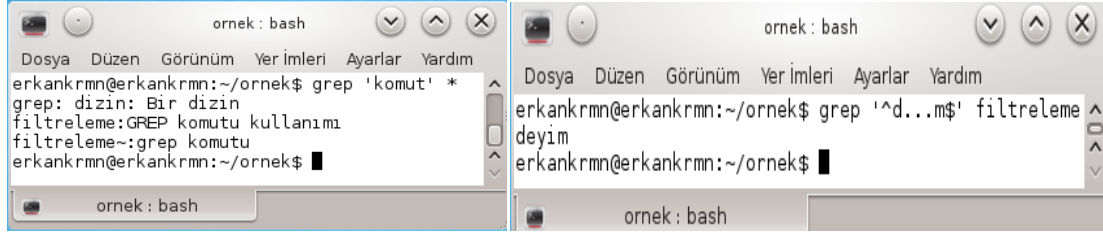
Kullanımı:grep [Araştırılacak kelime ve Düzenli Deyim] [Araştırılacak Dosya]



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep Mesleki filtreleme
Mesleki ve Teknik Eğitim
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.46. Grep komutu örnek kullanımı

Resim 1.46'da *filtreleme* dosyasında içerisinde *Mesleki* kelimesi geçen metinler listelenmiştir.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep 'komut' *
filtreleme:GREP komutu kullanımı
filtreleme~:grep komutu
erkankrnm@erkankrnm:~/ornek$

ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep '^d...m$' filtreleme
deyim
erkankrnm@erkankrnm:~/ornek$
```

(a)

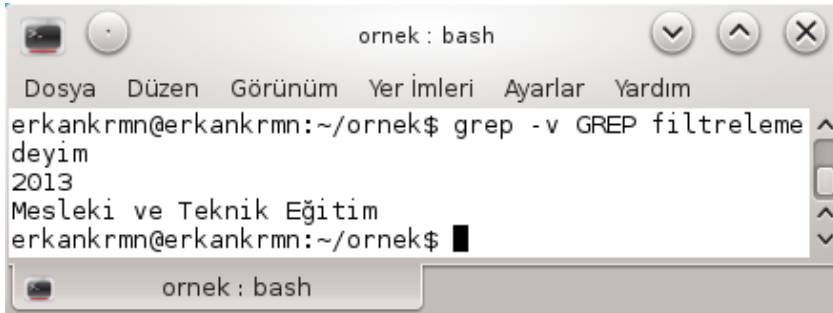
(b)

Resim 1.47: Grep komutu düzenli deyim ile kullanımı

Resim 1.47(a)'da kullanılan grep komutunda * deyimi ile dizin içerisindeki tüm dosyalarda içerisinde *komut* kelimesi geçen metinler listelenmiştir. Resim 1.47(b)'de *filtreleme* dosyası içerisinde *d* harfi ile başlayıp *m* harfi ile biten beş harfli metinler listelenmiştir.

Grep komutu seçenekleri ile de daha detaylı listeleme yapılabilir.

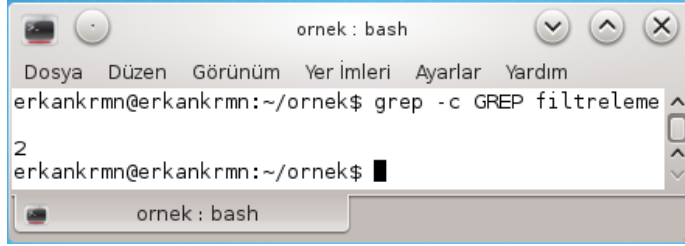
- **-v:** Seçeneği ile kullanılırsa belirli kelime dışında kalan metinler listelenecektir.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep -v GREP filtreleme
deyim
2013
Mesleki ve Teknik Eğitim
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.48: Grep komutu -v seçeneği kullanımı

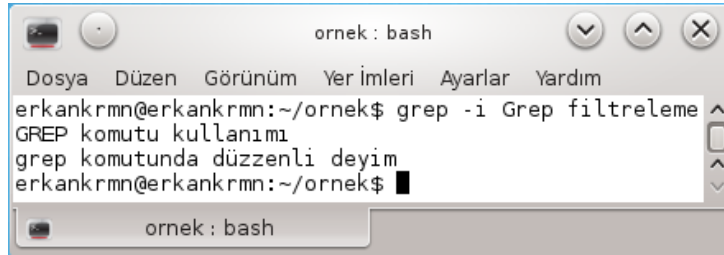
- **-c:**Seçeneği ile kullanılırsa aranan kelimenin dosya içerisinde kaç kez kullanıldığı listelenecektir.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep -c GREP filtreleme
2
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.49. Grep komutu -c seçeneği kullanımı

- **-i:**Seçeneği ile kullanıldığında, küçük/büyük harf ayırt etmeksizin metinler listelenecektir.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ grep -i Grep filtreleme
Grep komutu kullanımı
grep komutunda düzenli deyim
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.50: Grep komutu -i seçeneği kullanımı

1.4.2. Cut Komutu

Cut komutu ile bir dosyanın ya da bir girdinin içeriğinden istenen parçaları almak için kullanılabilir.

Kullanımı: *cut* seçenekler... [Dosya veya Girdi]

Seçenekler:

- **-b** --> byte cinsinden içerik alma,
- **-c** --> karakter cinsinden içerik alma,
- **-d** --> belirtilen ayraç (delimiter) dikkate alınır,
- **-f** --> belirtilen ayraça göre hangi alanların alınacağı hesaplanır.

Örnek:

```
@erkankrnm:~$: cut -c 3-5 notlar.txt
```

```
gep
yle
uss
```


Örnekte *notlar.txt* dosya içeriğindeki 3. karakterden 5. karaktere kadar *-c* seçeneği ile kesim işlemi yapar.

1.4.3. Sort Komutu

Sort komutu, bir metin dosyasındaki satırları alfabetik sıraya dizerek standart çıktıya yazar.

Kullanımı:*sort*. [Dosya adı]

Örnek:

```
linux:~ # cat deneme
a
c
b
linux:~ # sort deneme
a
b
c
linux:~ #
```

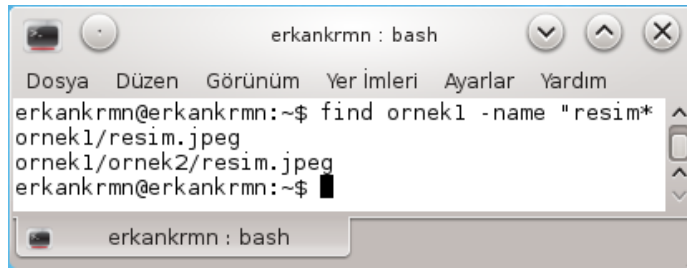
Örnekte deneme dosyasındaki karakterler alfabetik olarak sıralanmıştır.

1.5. Dosyaların Taranması

Açık kaynak işletim sistemi kabuk programında komut satırını kullanarak dosya ve dizinler aranabilir. Dosyaların içerisinde aranılan metinler görüntülenebilmektedir.

- **Find komutu:** Find komutu ile dizinleri, dosyaları sistemde bulmak mümkündür. Dosyaları türlerine göre arama işlemleri de yapılabilir.

Kullanımı:*find [aranacak_yol(dizin veya dosya)] [seçenekler] [düzenli_deyim]*

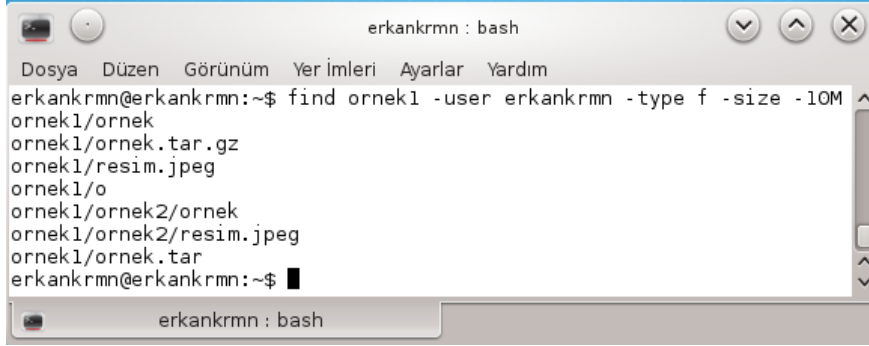


Resim 1.51: Find komutu örnek kullanımı

Resim 1.51’de *ornek1* dizininde adı *resim* ile başlayan dosyalar aranmıştır. Find komutu seçenekleri ile birlikte kullanılır.

- **-name:** Dosya veya dizinin ismine göre arama işlemi yapılır.
- **-size:** Dosya veya dizinin boyutuna göre arama işlemi yapılır.

- **-user:** Kullanıcıya göre arama işlemi yapılır.
- **-type:** Türe göre arama yapılacağını belirler.
 - o *.f* dosya türünün arama yapılacağını belirler.
 - o *d* dizine göre arama yapılacağını belirler.



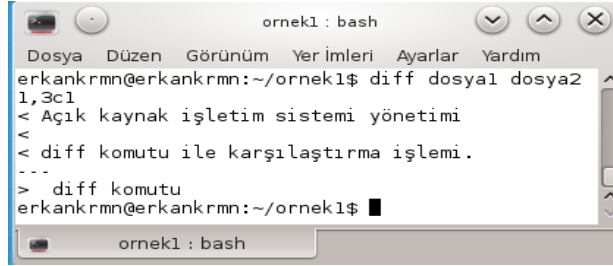
```

erkankrnm@erkankrnm:~$ find ornek1 -user erkankrnm -type f -size -10M
ornek1/ornek
ornek1/ornek.tar.gz
ornek1/resim.jpeg
ornek1/o
ornek1/ornek2/ornek
ornek1/ornek2/resim.jpeg
ornek1/ornek.tar
erkankrnm@erkankrnm:~$
  
```

Resim 1.52: Find komutu seçeneklere göre arama

Resim 1.52’de kullanıcı adı erkankrnm olan *ornek1* dizininde türü dosya olan (*f*) ve boyutu 10 megabyte’tan az olan dosyaların arama işlemi yapılmıştır.

- **Diff komutu:** Bir metin dosyasını başka bir metin dosyası ile karşılaştırma işlemi yapar, dosyanın benzerliklerini ve farklılıklarını bildirir.



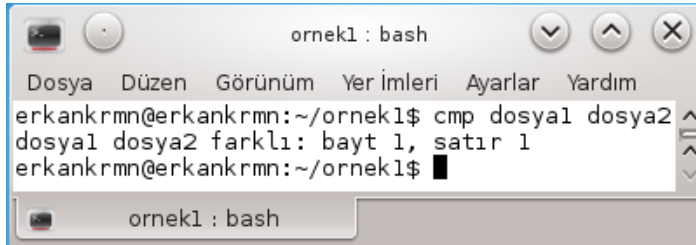
```

ornek1@erkankrnm:~/ornek1$ diff dosya1 dosya2
1,3c1
< Açık kaynak işletim sistemi yönetimi
^^
< diff komutu ile karşılaştırma işlemi.
...
> diff komutu
ornek1@erkankrnm:~/ornek1$
  
```

Resim 1.53: Diff komutu örnek kullanımı

Resim 1.53’te ‘<’ işareti *dosya1* içeriğini ‘>’ işareti de *dosya2* içeriğinin görüntülediğini bildirir.

- **Cmp komutu:** Herhangi iki tür dosyayı karşılaştırır ve sonucu ekrana yazar. Dosyaların aynı olduğu durumda öntanımlı olarak cmp komutu birşey yapmaz. Eğer farklıysalar, ilk farkın olduğu bayt ve satır sayısı gösterilir.



```

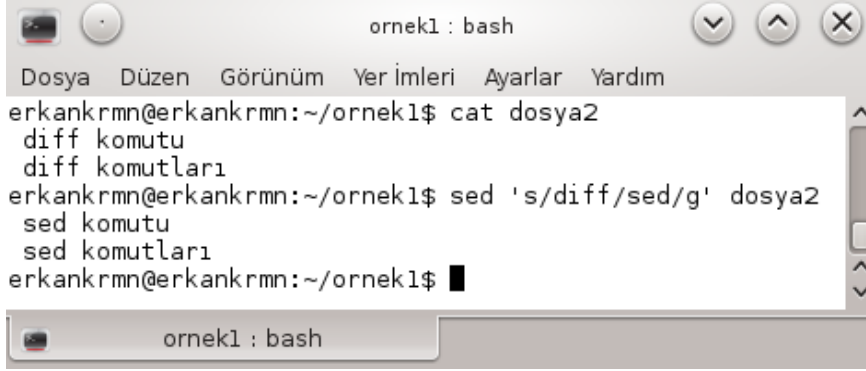
ornek1@erkankrnm:~/ornek1$ cmp dosya1 dosya2
dosya1 dosya2 farklı: bayt 1, satır 1
ornek1@erkankrnm:~/ornek1$
  
```

Resim 1.54: Cmp komutu örnek kullanımı

- **Sed komutu:** Sed komutu bir dosyadan veya klavyeden bilgi okur ve ekrana okuduğu bilgileri kullanıcının belirlediği düzene sokarak yazar. Veriyi de kullanıcı genellikle bir dosyaya yerleştirir.

Kullanımı: `sed [seçenekler] '{komut}' [dosya ismi]`

- Sed komutu kullanılırken en çok kullanılan iki argüman vardır:
 - o 's' argümanı bir karakteri veya karakter kümesini başka bir diziye çevirmeye yarar.
 - o 'g' argümanı dosyanın tüm satırlarını araştırılması için kullanılır. Eğer bu argüman kullanılmaz ise her satırın başında bulunan ilk değiştirmey yapılacak ve satırın geri kalanına bakılmayacaktır.



```
ornek1 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek1$ cat dosya2
diff komutu
diff komutları
erkankrnm@erkankrnm:~/ornek1$ sed 's/diff/sed/g' dosya2
sed komutu
sed komutları
erkankrnm@erkankrnm:~/ornek1$
```

Resim 1.55: Sed komutu örnek kullanımı

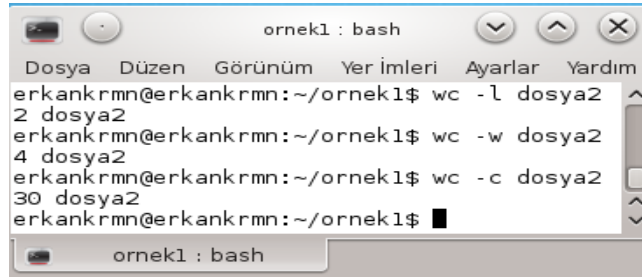
Resim 1.55'te *dosya2* içerisinde *diff* kelimesi *sed* kelimesi ile değiştirilmiştir.

- **Wc komutu:** Bir metin dosyasının satır, kelime ve karakter sayısını gösterir.

Kullanımı: `wc [seçenek] [dosya adı]`

Bu komut seçenekleri ile birlikte kullanılabilir. Seçenekler:

- **-l:** Satır sayısını
- **-w:** Kelime sayısını
- **-c:** Karakter sayısını gösterir.

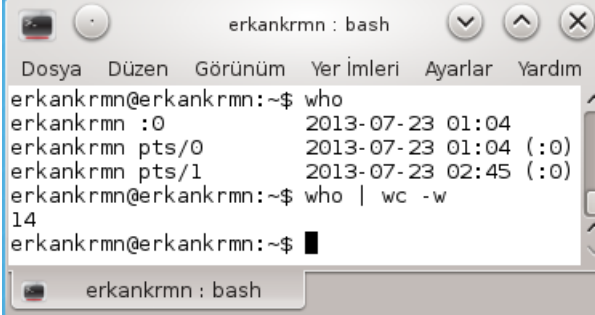


```
ornek1 : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek1$ wc -l dosya2
2 dosya2
erkankrnm@erkankrnm:~/ornek1$ wc -w dosya2
4 dosya2
erkankrnm@erkankrnm:~/ornek1$ wc -c dosya2
30 dosya2
erkankrnm@erkankrnm:~/ornek1$
```

Resim 1.56: Wc komutu örnek kullanımı

1.6. Boru (Pipe) İşlemleri

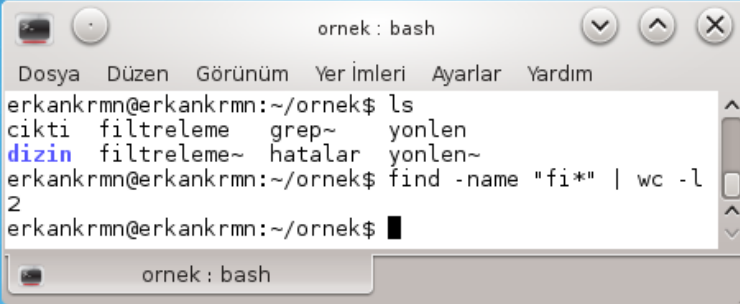
Bir komutun çıktısını, diğer bir komuta yönlendirmek için kullanılır. Yönlendirme işlemlerinde '>' ve '>>' operatörleri kullanılır. Komuta yönlendirme yapılmak için '|' operatörü kullanılır. Bu operatör, kendinden önce gelen komut veya komut serisinin çıktısını, kendinden sonra gelen komuta gönderir.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~$ who
erkankrnm :0      2013-07-23 01:04
erkankrnm pts/0  2013-07-23 01:04 (:0)
erkankrnm pts/1  2013-07-23 02:45 (:0)
erkankrnm@erkankrnm:~$ who | wc -w
14
erkankrnm@erkankrnm:~$ █
```

Resim 1.57: Boru işlemi komuta yönlendirme işlemi

Resim 1.57’de sistemdeki kullanıcı bulunmuş ve kelime sayısını bulmak için başka bir komuta yönlendirilmiştir.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ ls
cikti filtreleme grep~  yonlen
dizin filtreleme~ hatalar yonlen~
erkankrnm@erkankrnm:~/ornek$ find -name "fi*" | wc -l
2
erkankrnm@erkankrnm:~/ornek$ █
```

Resim 1.58: Boru işlemleri

Resim 1.58’de dosya adı *fi* ile başlayan dosyadaki satır sayısını gösterme işlemi yapılmıştır.

1.7. Temel Kabuk Programlama

1.7.1. Kabuk Programlamaya Giriş

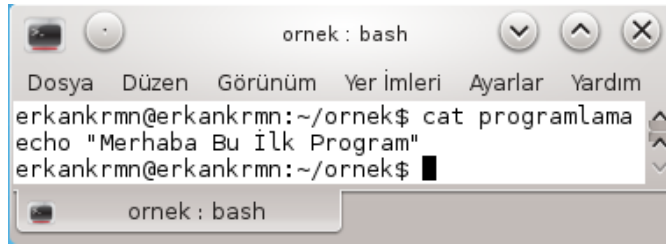
Açık kaynak işletim sisteminin son sürümünde program kodlarının yazılmasında bash kabuğu kullanılmaktadır. Bash kabuğu, güçlü programlama özellikleri ile program kodlarının yazılmasında izin vermektedir. Kabuk programlama da kabuğun programlanması değil bu kabuğa ait komutlarla, tek komutla işlerin yapılması anlaşılmaktadır. Bir dosyaya yazılmış komutlar yığını, dosya çalıştırılarak yapılır. Bu dosyalara ise genelde *script* adı verilir.

Mantıksal operatörler, döngüler, değişkenler ve modern programlama dillerinde bulunan pek çok özellik bash kabuğunda da bulunmaktadır.

Bash kabuğunun en büyük dezavantajı, derlenerek çalıştırılan programlama dillerine göre daha yavaş olması, sistem kaynaklarını biraz daha fazla tüketmesidir.

Dosya içerisine programın çalıştırılacağı kabuk belirtilir. Programın çalışacağı kabuk belirtilmezse işletim sisteminde bulunan kabuk üzerinden program çalıştırılır. Kabuk programının yazılım farklarından kaynaklanan hatalarla karşılaşmaması için programın çalıştırılacağı kabuk programı belirtilmelidir. Metin dosyası içerisinde kabuk belirtmek için **#!/bin/bash** kullanılır.

Bir metin dosyası oluşturularak içerisine program kodları yazılır. Kullanılacak ilk programlama komutu **echo** komutudur. Bu komut ekrana mesaj yazmak için kullanılır.

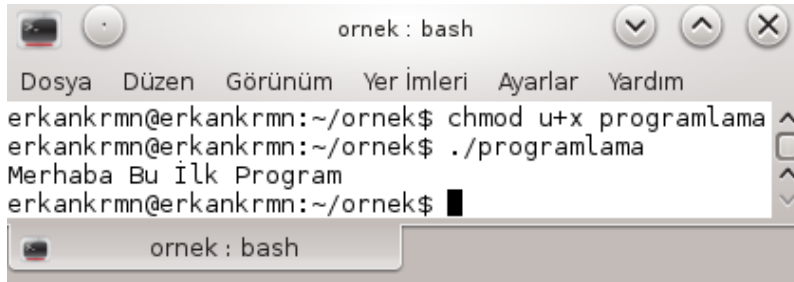


```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ cat programlama
echo "Merhaba Bu İlk Program"
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.59: Echo programlama komutu örnek kullanımı

Resim 1.59’da *programlama* metin dosyası içerisine *echo* komutu kullanılarak ekrana bir mesaj yazılmıştır.

Metin dosyasını kabuk programın komut satırında çalıştırmak için gerekli izinlerin verilmesi gerekmektedir. İzin vermek için **chmod u+x** komutu kullanılır. Bu komutun kullanımı: **chmod u+x[çalışacak dosyanın adı]**.



```
ornek : bash
Dosya Düzen Görünüm Yer İmleri Ayarlar Yardım
erkankrnm@erkankrnm:~/ornek$ chmod u+x programlama
erkankrnm@erkankrnm:~/ornek$ ./programlama
Merhaba Bu İlk Program
erkankrnm@erkankrnm:~/ornek$
```

Resim 1.60: Dosya çalıştırma

Resim 1.60’ta dosyanın çalışması için gerekli izinler verilmiştir. *./programlama* dosyasının adı yazılarak dosyanın içerisindeki program komutları çalıştırılmıştır.

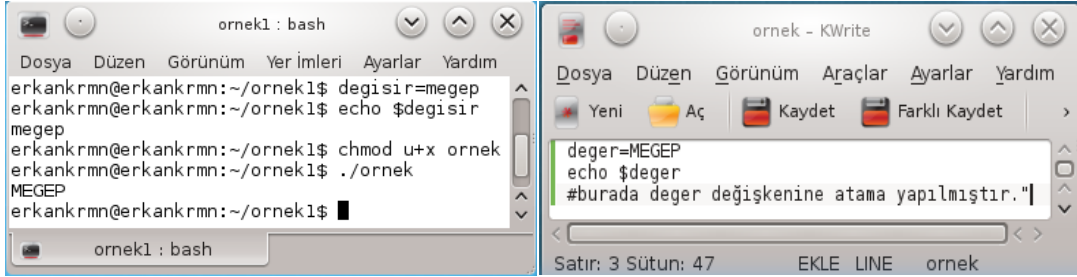
1.7.2. Değişkenler

Değişkenler kendilerine atanan, gerekli olan değerleri saklayan ve gerektiğinde bunları değiştirmeye olanak sağlayan yapılardır. Programlamada kullanılan en önemli yapılardır. Değişkenler, programda kullanılan bir sayı değerini (tam veya ondalıklı sayı) ya da bir karakter dizgesini (string) kendi içerisinde saklayabilir.

Değişkenler tanımlanırken bazı kurallara uymak gereklidir. Bunlar:

- Değişkenler sayısal bir değer ile başlayamaz. Örneğin, *1a*, *135b* olamaz.
- Değişkenlerde küçük/büyük harf duyarlılığı bulunmaktadır. Örneğin, *a* ile *A* farklı değişkenleri temsil eder.
- Değişkenler tanımlanırken Türkçe karakterler kullanılmaz. Örneğin, *sayı* değil de *sayi* olmalıdır.
- Değişkenler iki kelimeden oluşur ise araya boşluk yerine '_' işareti eklenir.
- Değişkenlere değer atanırken '=' operatörü kullanılır. Eşitliğin sağ tarafındaki değer değişkene aktarılır.

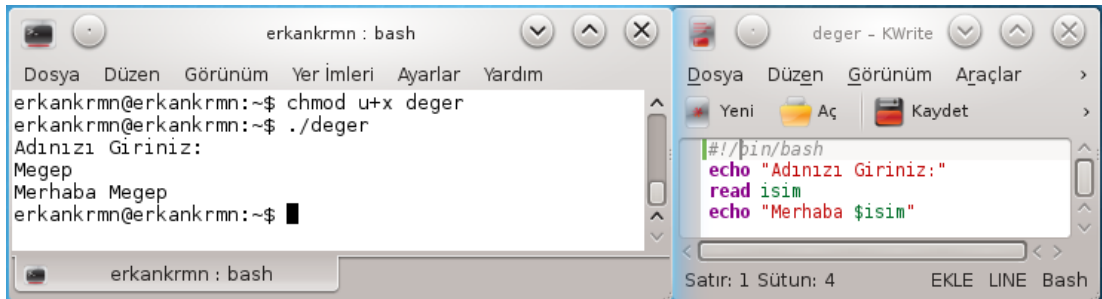
Değişkenler komut satırında kullanılabileceği gibi *script* dosyalarında da kullanılabilir.



Resim 1.61: Örnek deęişken kullanımı

1.7.3. Deęer Okuma

Dosyaların içerisinde deęişkenler tanımlanarak bu deęişkenlere komut satırından deęer verilebilir. Dosya içerisinde dışarıdan deęer okumak için *read* komutu kullanılır.

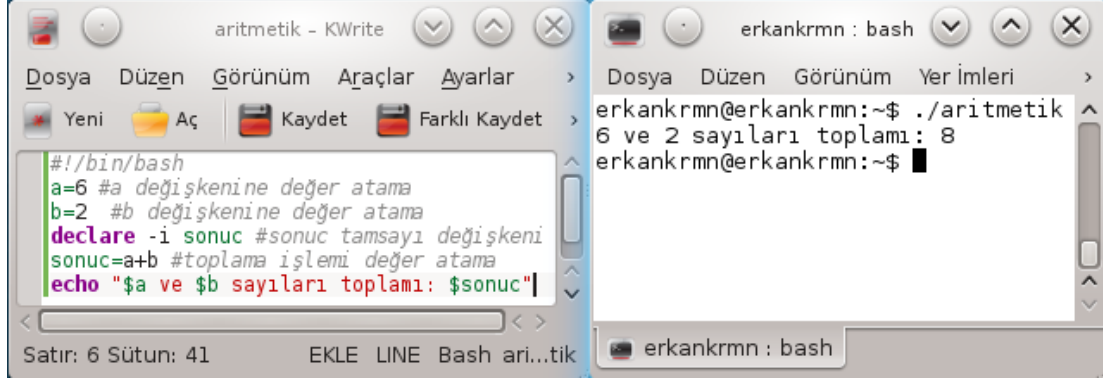


Resim 1.62: Deęişkene dışarıdan girilen deęeri okuma

Resim 1.62'de *deger* dosyasında dışarıdan isim girilmesi için mesaj yazılmıştır (*echo adınızı giriniz*). *Read* komutu ile okunan deęer *isim* deęişkenine aktarılmıştır. Okunan deęer ekrana yazılmıştır. Komut satırında ise dosya çalıştırılmış ve içerisindeki program komutlarına göre dışarıdan deęer okunmuş ve deęişkene aktarılmıştır.

1.7.4. Aritmetik İşlemler

Değişkenlerden değer okumanın yanı sıra *script* dosyalarında aritmetiksel işlemler de yapılabilir. Tam sayılar ve ondalık sayılar ile matematiksel işlemler yapılabilir. Tamsayı değişkenini tanımlamak için **declare** tanımlama komutu ve tamsayı değişkeni olacağını belirten **-i** seçeneği kullanılır.

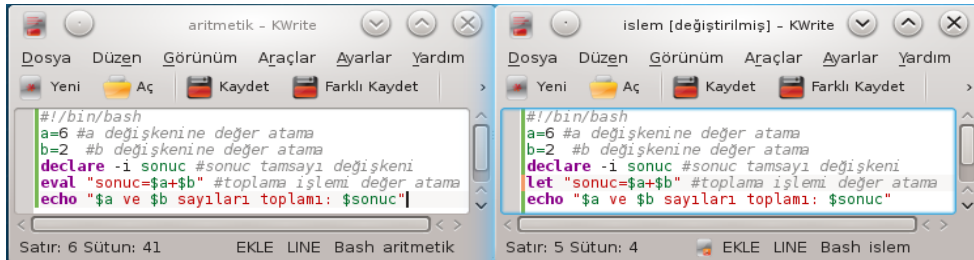


```
#!/bin/bash
a=6 #a değişkenine değer atama
b=2 #b değişkenine değer atama
declare -i sonuc #sonuc tamsayı değişkeni
sonuc=a+b #toplama işlemi değer atama
echo "$a ve $b sayıları toplamı: $sonuc"
```

Resim 1.63:Aritmetik işlemler örneği

Resim 1.63'te *aritmetik* metin dosyası içerisinde *a* ve *b* değişkenlerine tamsayı değerleri atanmıştır. *Declare* komutu **-i** seçeneği ile *sonuc* isimli değişken tanımlanmıştır. Toplama işlemi *sonuc* değişkenine aktarılmış ve *echo* komutu ile ekrana mesaj yazılması sağlanmıştır.

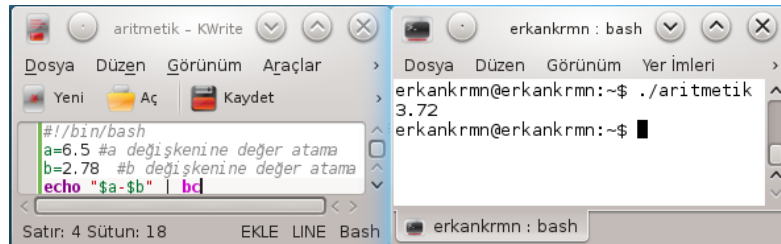
Aritmetik işlem hesaplamalarında *eval* ve *let* komutları da kullanılabilir. Bu komutlar işlem satırında kullanılır.



```
#!/bin/bash
a=6 #a değişkenine değer atama
b=2 #b değişkenine değer atama
declare -i sonuc #sonuc tamsayı değişkeni
eval "sonuc=$((a+b))" #toplama işlemi değer atama
echo "$a ve $b sayıları toplamı: $sonuc"
```

Resim 1.64:Let ve eval komutu kullanımı

Bc komutu kullanılarak ondalıklı sayı işlemleri de yapılabilir.



```
#!/bin/bash
a=6.5 #a değişkenine değer atama
b=2.78 #b değişkenine değer atama
echo "$a-$b" | bc
```

Resim 1.65:Bc komutu kullanımı

Bckomutu kullanılarak *a* ve *b* değişkenlerinde bulunan ondalıklı sayılar ile aritmetiksel işlem yapılmıştır.

1.7.5. Kontrol Yapıları

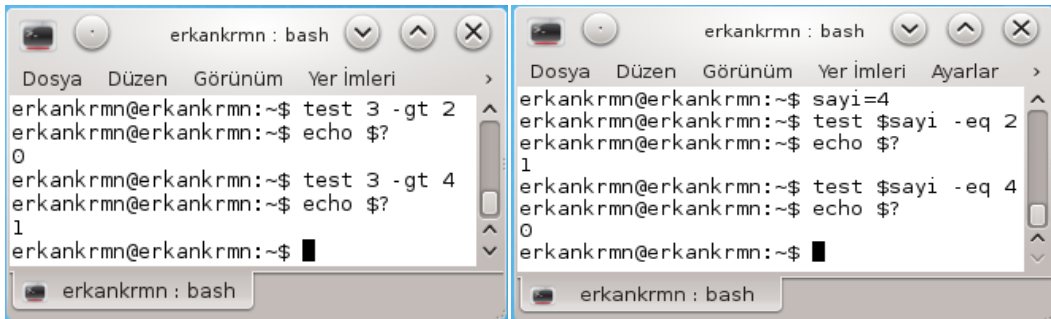
Temel kabuk programlama yapısında kontrol;*test*, *if* – *else* *vecase* deyimleri kullanılarak yapılır.

- **Test:** Mantıksal işlemleri yapan deyimdir. Bu deyim sayesinde sayılar veya diziler karşılaştırılabilir. *Test* deyiminden sonra seçenekler *ve/veya* karşılaştırılacak olan değerler yazılır. Her seçenek bir mantıksal işleme karşılık gelir.

Aritmetik karşılaştırma		Dizisel karşılaştırma		Dosya karşılaştırması		Mantıksal karşılaştırma	
-gt	büyük	-z	boş dizi	-f	Dosya var	-a	VE
-lt	küçük	-n	tanımlı dizi	-s	Dosya boş değil	-o	VEYA
-ge	büyük eşit	=	eşit diziler	-r	Dosya okunabilir	!	DEĞİL
-le	küçük eşit	!=	farklı diziler	-w	Dosyaya yazılabilir		
-eq	eşit			-x	Çalıştırılabilir dosya		
-ne	eşit değil			-h	Sembolik bağlantı		
				-c	Karakter dosyası		
				-b	Blok dosyası		

Tablo 1.3: Test deyiminde kullanılan seçenekler

Test deyiminin işletilmesinin ardından kabuğa bir değer döner. Bu değer komut başarılı olarak işletilmişse 0, değilse 1'dir. Son çalıştırılan tüm deyimlerin çıkış değeri \$? ifadesinde tutulur. *Test*deyiminin çıkış değeri de bu yolla öğrenilir.



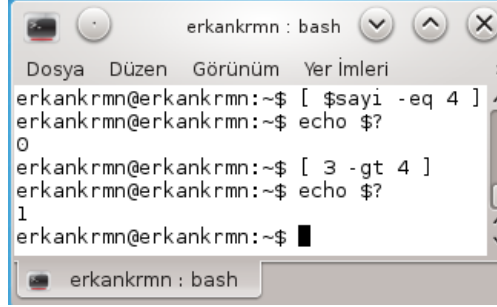
(a)

(b)

Resim 1.66: Test deyimini ile aritmetiksel karşılaştırma

Resim 1.66 (a)'da iki sayı karşılaştırılmıştır. İlk sayı ikinci sayıdan büyük ise sonuç 0 küçük ise 1'e dönmüştür. Resim 1.66 (b)'desayi değişkeni içerisindeki değer ile verilen değere eşit ise sonuç 0 eşit değil ise 1'e dönmüştür.

*Test*deyimi yerine köşeli parantezler de kullanılabilir. Dikkat edilmesi gereken bir nokta, köşeli parantez kullanırken araya boşlukların eklenmesidir. Parantezler başlı başına bir komut olarak gördüklerinden parantezin sağında ve solunda en az bir boşluk bırakılmalıdır.



Resim 1.67: Test deyimi yerine parantez kullanımı

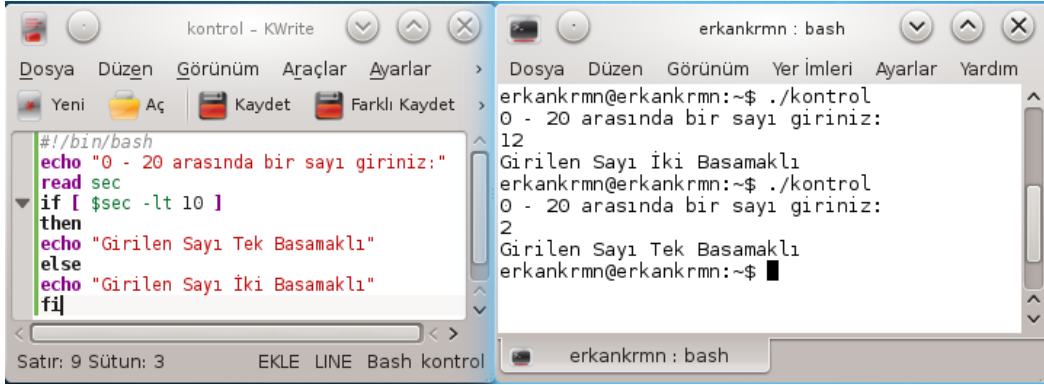
- **If – else:** If deyimi, mantıksal bir karşılaştırmayı sınyarak bir işlemin yapılmasına veya yapılmamasına yada başka bir işlemin gerçekleştirilmesine karar vermek için kullanılan deyimdir.

If – else yapısı:

```
if [ koşul-1 ]; then  
koşul-1 doğru ise yürütülecek komutlar  
elif [ koşul-2 ]; then  
koşul-2 doğru ise yürütülecek komutlar  
.....  
elif [ koşul-n ]; then  
koşul-n doğru ise yürütülecek komutlar  
else  
hiçbir koşul doğru değilse yürütülecek komutlar  
fi
```

If– else yapısı bir koşulun çalışmasını kontrol (test) eder. **If**deyiminin ardından gelen koşul çalıştırılır ve koşulun çıkış durumu alınarak ardından gelen **then** deyimiyle birlikte devamı işletilir. Genellikle koşulun iki türlü çıkış durumu olur.**Else** komutunun ardından gelen komut zinciri, diğer çıkış durumunda çalıştırılır. Her **if**, bir **fi** komutuyla bitmelidir.

Elif ve *else* durumları koşula bağlı olarak kullanılır. İlk koşul sağlanmadığında sınyanması gereken diğer koşullar *elif* kısmında kullanılır. Hiçbir koşul sağlanmadığı durumlarda gerçekleşecek bir komut var ise *else* kısmından sonra bu komutlar işletilir.



Resim 1.68:İf – else kontrol yapısı

Resim 1.69’da *kontrol* script dosyasında girilen sayının kaç basamaklı olduğu kontrol edilmekte ve sonuç ekrana yazılmaktadır.

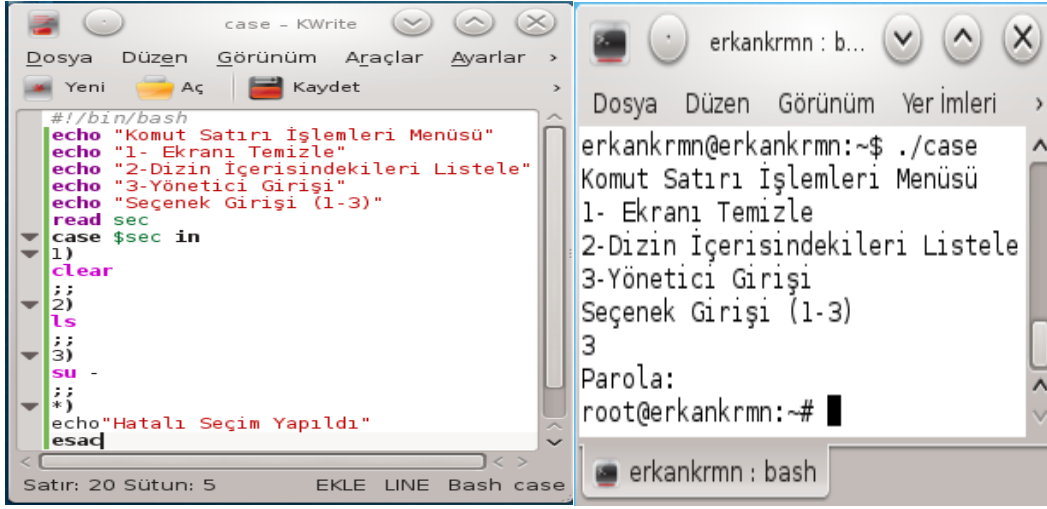
- **Case:** Bir veya birden fazla koşul arasından seçim yapmak için kullanılır. *Case* deyimi koşullardan birisi sağlandığında o koşul altındaki komutlar işleme girer. *Case* yapısı *case* deyimi ile başlar anahtar sözcük yazılır ve koşullar alt alta belirtilir. Her koşula ait olan komutlar koşulunun altına yazılır. *Case* yapısı *esac* ile sonlandırılır.

```

caseanahtar_sözcükün
koşul-1)
komutlar
;;
koşul-2)
komutlar
;;
...
*)
komutlar
;;
esac

```

Case yapısının genel kullanımında en altta bulunan * işareti hiçbir koşul gerçekleşmediğinde çalışır. Komutlar altında bulunan “;;” işareti koşul altında bulunan komutların bittiğini gösterir. Koşullar arasında özel karakterler (*, [,], ? gibi) kullanılabilir.



Resim 1.69: Case yapısı örnek kullanımı

Resim 1.69’da *case* yapısı kullanılarak *casescript* dosyasında kullanıcı işlemleri için menü hazırlanmıştır. Kullanıcı tarafından girilen seçenekler kontrol edilerek gerekli işlemler yapılmıştır. Komut satırında script dosyası çalıştırılarak kullanıcıdan menü üzerinde işlemler yaptırılmıştır.

1.7.6. Döngüler

Belirlenen sayıda veya belli bir koşula bağlı olarak tekrar etmesi istenilen komutlar için döngü yapıları kullanılır. Komutlar döngü içerisine bir kez yazılır, istenen sayı veya istenen koşul gerçekleşene kadar tekrar edilir. Döngüler kullanılarak komutlar tekrar tekrar kullanılmaz.

- **Fordöngüsü:** Belirlenen komutların, belirlenen sayıda veya belli koşulda tekrar edilmesi için kullanılır. Kontrolü yapılacak değişken *for* döngüsü üzerinde bulunduğu herhangi bir sayaç oluşturulmaz.

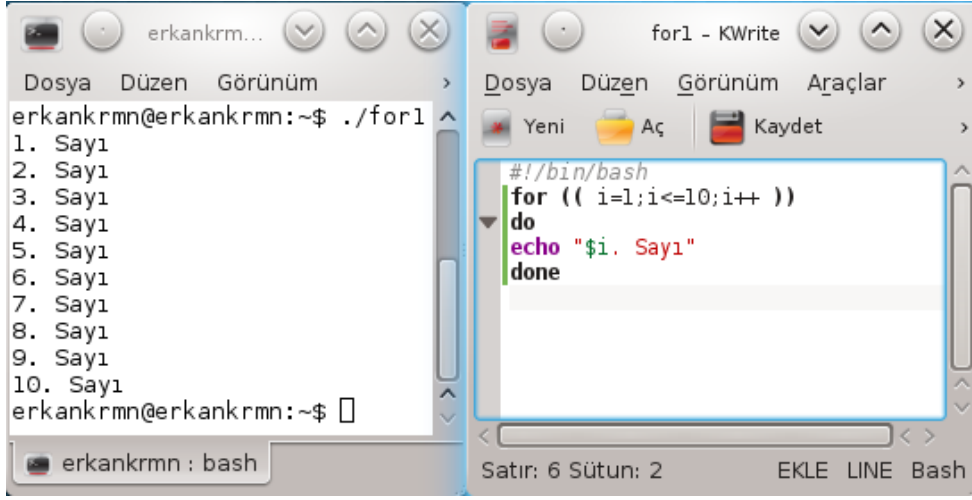
Sayaç yapısı ile kullanımı:

```
for (( sayaç başlangıç; koşul; sayaç artırımı ))
do
```

```
    Koşul yanlış olana kadar do ve done arasındaki tüm komutları tekrarlar.
done
```

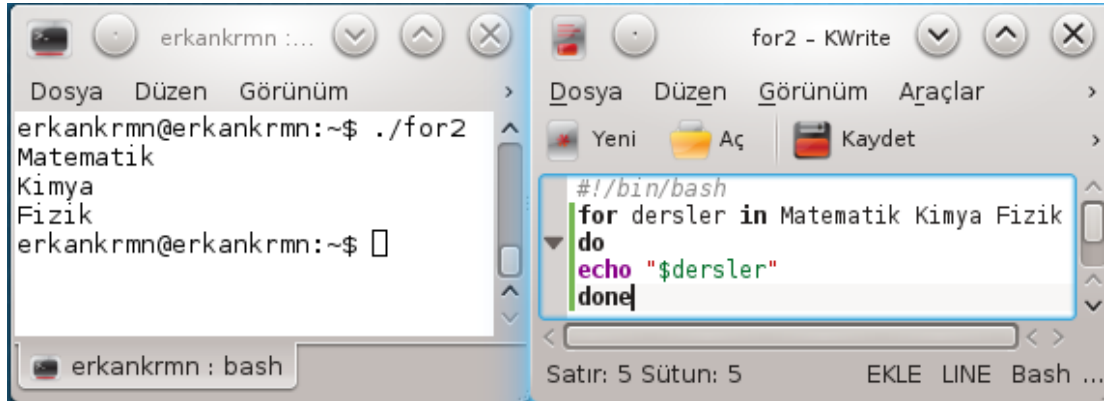
Kontrolü yapılacak değişkenin değerler belirli olduğu durumlarda kullanımı:

```
for değişken in değerler
do
    komutlar
done
```



Resim 1.70:For döngüsü sayaç kullanımlı

Resim 1.70’te *for1*script dosyasında sayacın başlangıç değeri belirtilmiştir ($i=1$). Sayacın bitiş değeri koşul olarak belirtilmiştir ($i \leq 10$). Sayacın birer birer artması sağlanmıştır ($i++$). **Do – done** deyimleri arasında sayıların sırayla ekrana yazdırılması için gerekli komut yazılmıştır (*echo “\$i. Sayı”*).



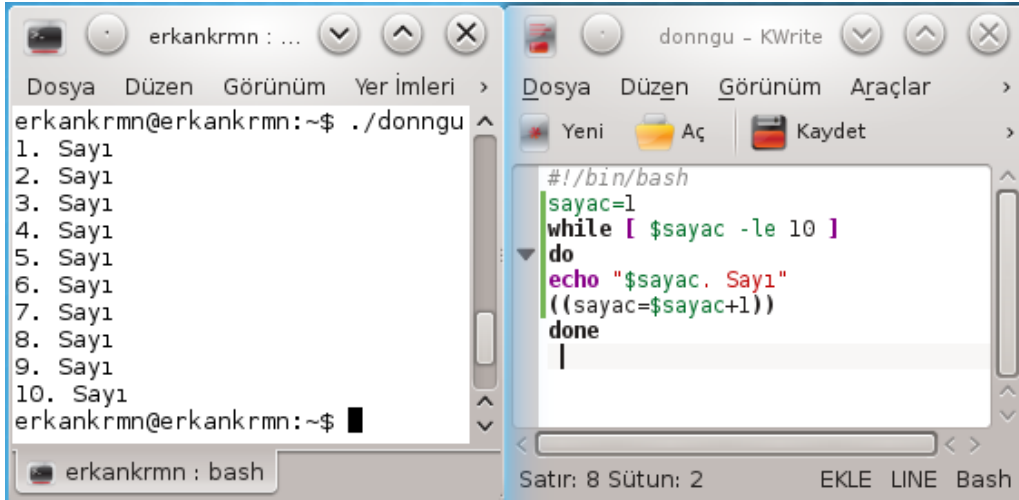
Resim 1.71:For döngüsü belirli değerlere göre çalışma yapısı

Resim 1.71’de *for2*script dosyasında değişken (*dersler*) içerisinde alınacak değerler belirlenmiştir (*Matematik Kimya Fizik*). **Do – done**deyimleri arasında belirlenen değerler tek tek yazılmıştır (*echo \$dersler*).

- **Whiledöngüsü:**Döngü bloğu *while* anahtar kelimesiyle başlar, ardından gelen koşul doğru olduğu sürece **do – done**deyimleri arasındaki komutlar tekrar edilir. Koşul yanlış olduğunda döngü sonlanır, **done** deyiminden sonraki komutlara geçilir. Döngüden çıkabilmek için mutlaka döngü içindeki koşul ifadesinin değerini yanlış yapacak bir durum oluşmalıdır, aksi halde sonsuz döngü oluşur döngüdeki komutlar sürekli çalışır.

While döngüsü kullanımı:

```
while koşul
do
komutlar
done
```



Resim 1.72: While döngü yapısı

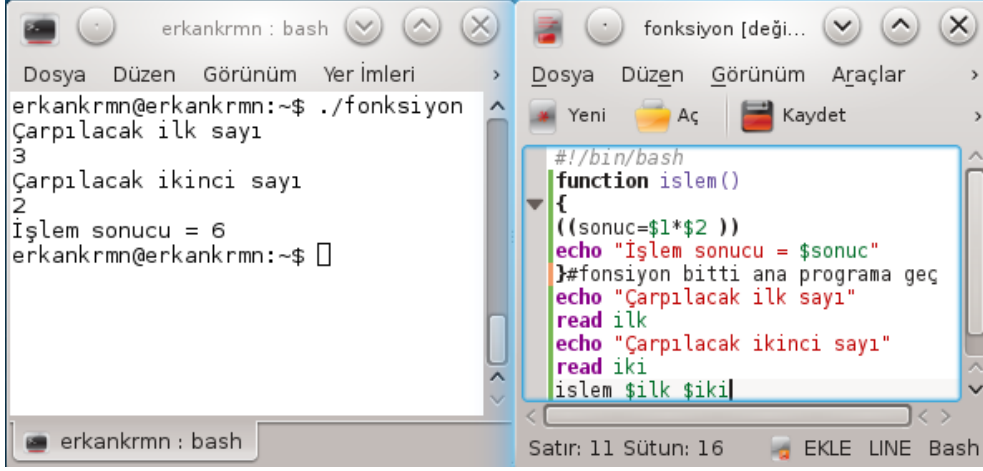
Resim 1.72’te *donnguscript* dosyasında döngü işlemi sayaç kullanılarak gerçekleştirilmiştir. *Sayac* değişkeni başlangıç değeri “1” olarak belirlenmiştir. Eğer *sayac* değişkeni değeri belirlenmeseydi *while* yapısındaki koşul sınavasında değişken içeriği boş (null) olduğu için hata verilecektir. *While* deyiminden sonra koşul (*sayac değışkeni 10’dan küçük olduğu sürece*) sağlandığı sürece *do – done* arasındaki komutları tekrar edilmesi sağlanmıştır. Döngü içerisinde “(())” parantezler arasında matematiksel olarak *sayac* değişkeni değeri “1” arttırılarak tekrar *sayac* değişkenine aktarılmıştır. Döngü içerisinde *sayac* değişkenin değeri her defasında ekrana yazılmıştır. Burada *sayac* değişkeni koşulu sağlamadığında döngü dışarısına çıkılacaktır.

1.7.7. Kabuk Fonksiyonları

Fonksiyonlar, bir program parçasını blok haline getirerek isimlendirmeye ve gerektiği yerde ismi ile çağırarak kullanmaya yarar. Programların kısa ve anlaşılır olmasında çok yararlı ve kullanışlı yapılardır. Program içinde sık kullanacak bazı yapıları bir düzen altında tanımlayarak devamlı kullanılabilir. Örneğin program içerisinde kullanıcıdan veri alacak bir bölüm olsun. Bu bölümü birden fazla yerde kullanmak gerektiğinde her kullanılacak yerde aynı şeylerin yazılması gereklidir. Bu blok fonksiyon olarak tanımlanırsak kullanılacak yerde fonksiyonu adıyla çağırılacaktır. Bu sayede daha az kod yazılacaktır.

```
fonksiyon_adi()
{
komutlar
}
.....
```

Tanımlanmış bir fonksiyonu çağırırken *fonksiyon_adi* yazılır.



```
erkankrnm : bash
Dosya Düzen Görünüm Yer İmleri
erkankrnm@erkankrnm:~$ ./fonksiyon
Çarpılacak ilk sayı
3
Çarpılacak ikinci sayı
2
İşlem sonucu = 6
erkankrnm@erkankrnm:~$ █

fonksiyon [deği...
Dosya Düzen Görünüm Araçlar
Yeni Aç Kaydet
#!/bin/bash
function islem()
{
  ((sonuc=$1*$2 ))
  echo "İşlem sonucu = $sonuc"
}#fonksiyon bitti ana programa geç
echo "Çarpılacak ilk sayı"
read ilk
echo "Çarpılacak ikinci sayı"
read iki
islem $ilk $iki
Satır: 11 Sütun: 16 EKLE LINE Bash
```

Resim 1.73: Fonksiyon yapısı

Resim 1.73'te *fonksiyon* script dosyası içerisinde *islem* adı ile bir program bloğu tanımlanmıştır. Ana programda kullanıcı tarafından iki sayı girilmiştir (*ilk* - *iki*). Ana programda *islem* fonksiyonu çağırılarak girilen sayıların çarpımı hesaplanarak ekrana yazılmıştır.

UYGULAMA FAALİYETİ

Açık kaynak işletim sisteminde komut satırını (*konsole - uçbirim*) uygulamasını çalıştırarak aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
➤ Kullanıcının ev dizininde ~ “a” harfi ile başlayan “m” harfi ile biten dosyaları ve dizinleri listeleyiniz.	➤ ls komutunu ve seçeneklerini kullanabilirsiniz. ➤ Özel sembollerden yararlanabilirsiniz.
➤ Kullanıcının ev dizini içerisinde ~ ornek/ornek1 isimli iç içe dizinler oluşturunuz.	➤ Mkdir komutunu ve seçeneklerini kullanabilirsiniz.
➤ Oluşturulan ornek1 dizini içerisine metin ve resim dosyalarını kopyalayınız.	➤ Cp komutunu ve seçeneklerini kullanabilirsiniz.
➤ Ekranda bulunan komutları temizleyiniz.	➤ Clear komutunu kullanabilirsiniz.
➤ Kullanıcının ev dizininden ornek klasörüne giriş yapınız.	➤ Cd komutunu kullanabilirsiniz.
➤ Örnek dizini içerisinde bulunan metin dosyalarından birinin içeriğini görüntüleyiniz.	➤ Cat komutunu kullanabilirsiniz.
➤ Dizin içerisinde bulunan metin dosyalarını siliniz.	➤ rm komutunu kullanabilirsiniz.
➤ Dizin içerisinde bulunan “jpeg” uzantılı dosyaları “ornek.tar” dosyası altında arşivleyiniz.	➤ Tar komutunu ve seçeneklerini kullanabilirsiniz.
➤ Bir scriptdosyası oluşturarak içerisinde dört işlem yapan (toplama, çıkartma, çarpma, bölme) basit bir hesap makinesi yapınız.	➤ Kontrol yapılarından case deyimini kullanabilirsiniz.
➤ Ekranı 10 kez isminizi yazan bir script dosyası oluşturarak komut satırında çalıştırınız.	➤ Döngü yapılarından forveya while yapılarını kullanabilirsiniz.
➤ Kullanıcı tarafından girilen iki sayının toplamını bulan fonksiyonu hazırlayınız.	➤ Kabuk fonksiyonlarını kullanarak hazırlayabilirsiniz.

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. “a” harfi ile başlayan “m” harfi ile biten dosyaları listeleyebildiniz mi?		
2. İç içe dizin oluşturabildiniz mi?		
3. Dosya kopyalama işlemini yapabildiniz mi?		
4. Ekran temizleme işlemini yapabildiniz mi?		
5. Dosya içeriğini görüntüleme işlemini yapabildiniz mi?		
6. Dosya silme işlemini yapabildiniz mi?		
7. Dosya arşivleme işlemini yapabildiniz mi?		
8. Basit bir hesap makinesi hazırlayabildiniz mi?		
9. İsminizi ekrana 10 kez yazdırabildiniz mi?		
10. Toplama işlemi fonksiyonunu hazırlayabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. () Kullanıcı tarafından verilen komutlar, kabuk tarafından algılanır ve sistem çağrılarını yardımı ile çekirdeğe iletilir.
2. () Bilgisayar ilk açıldığında masaüstü görüntülenir.
3. () İşletim sistemlerinde açılan her pencerede kullanıcının yazdığı her karakteri okuyup bilgisayarın donanımının anlayacağı gerçek komutlara çeviren kabuk çalışır.
4. () Açık kaynak işletim sisteminde kabuk programlama, komut satırı uygulamasında gerçekleşir.
5. () Komut satırında, komutlar ve dosya isimlerinde büyük/küçük harf kullanımı önemli değildir.

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

6. Aşağıdaki komutlardan hangisi yardım almak için **kullanılmaz**?
A) help
B) info
C) man
D) ls
7. Aşağıdaki komutlardan hangisi dosya ve dizin taşımak için kullanılır?
A) cp
B) rm
C) mv
D) cd
8. Aşağıdaki komutlardan hangisi dosya veya dizin aramak için kullanılır?
A) grep
B) find
C) cat
D) who
9. Boru (*pipe*) işlemi ne amaçla kullanılır?
A) Bir dosyayı diğer bir dosyaya bağlamak için
B) Dosyaları birbirine eklemek için
C) Bir komutun çıktısını başka bir komuta yönlendirmek için
D) Komutları aynı anda çalıştırmak için

10. Aşağıdaki aritmetik işlem seçeneklerinden hangisi “eşit değil” anlamında kullanılır?

- A) –gt
- B) –lt
- C) –eq
- D) –en

11. Aşağıdaki deyimlerden hangisi program bloğu oluşturmak için kullanılır?

- A) for
- B) if
- C) function
- D) while

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları, faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Açık kaynak işletim sisteminde dosya ve izin paylaşımı için ağ servislerini kullanabileceksiniz.

ARAŞTIRMA

- İşletim sistemlerinde yerel ağ ayarları hakkında araştırma yapınız.
- Dosya ve izin paylaşımı hakkında araştırma yapınız.
- İşletim sistemlerinde dosya ve izin paylaşımlarının ortak noktalarını araştırınız.
- İnternet üzerinden dosya paylaşım yöntemlerini araştırınız.

2. DOSYA VE DİZİN PAYLAŞIMI

2.1. Dosya ve Dizin Paylaşımı

Ağa bağlı olan bir bilgisayarda, kullanıcının dosyaları ve izinleri ağa bağlı olan başka bir kullanıcı tarafından kullanılabilir. Kullanıcı dosyalarını ve izinlerini işletim sistemi üzerinden paylaşımına açarak ağa bağlı başka bir kullanıcı tarafından paylaşılmış dosyalara ve izinlere erişim sağlanabilir.

2.1.1. Dosya ve Dizinlerin Paylaşımına Açılması

Açık kaynak işletim sisteminde, disk üzerinde bulunan kullanıcının dosyaları ve izinleri ağa bağlı olan diğer kullanıcıların erişim sağlaması için paylaşımına açılabilir. Aynı zamanda açık kaynak işletim sistemi üzerindeki dosyalar lisanslı bir işletim sistemindeki kullanıcılara paylaşımına açılabilir.

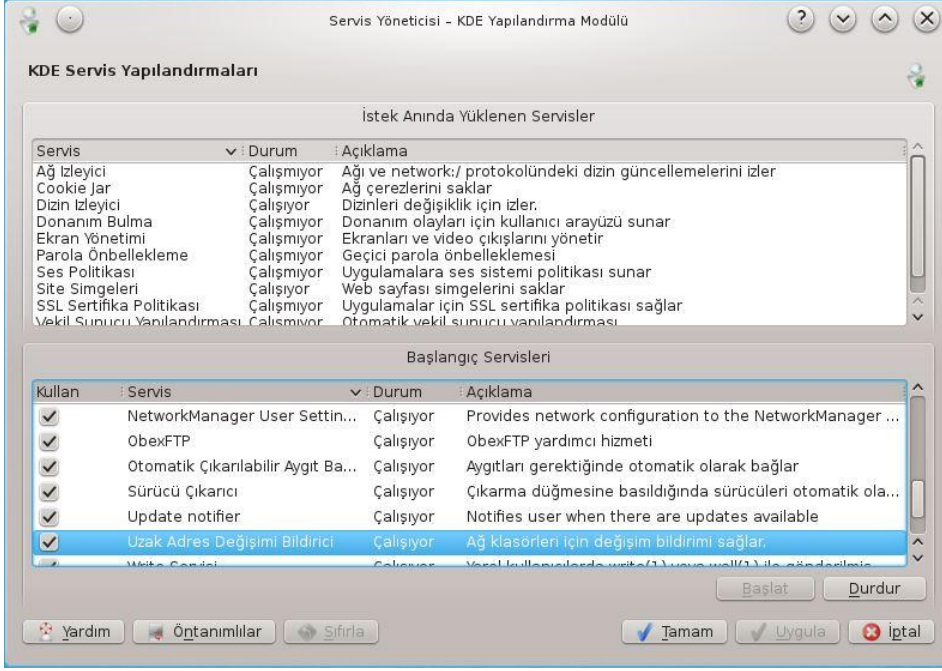
Dosyaların ve izinlerin özellikleri üzerinden paylaşım mümkündür. Paylaşım sırasında herhangi bir sorun ile karşılaşılır ise *Servis Yöneticisi* ayarları kontrol edilmelidir.

Servis Yöneticisini başlatmak için *Uygulama Yöneticisinden* Araçubuğuna *Servis Yöneticisi* yazmak gerekir.



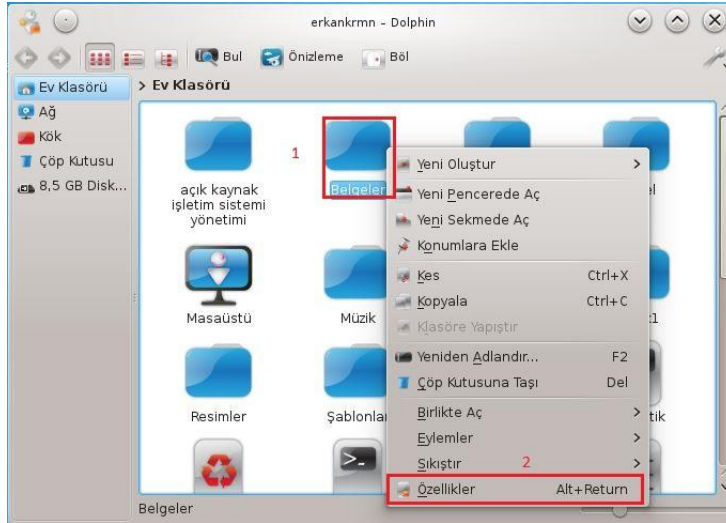
Resim 2.1: Servis yöneticisi başlatma

Servis Yöneticisi penceresinde tüm servislerin çalıştığından emin olunması gerekmektedir. Çalışmayan servisler *Başlat* butonuna basılarak çalıştırılabilir.

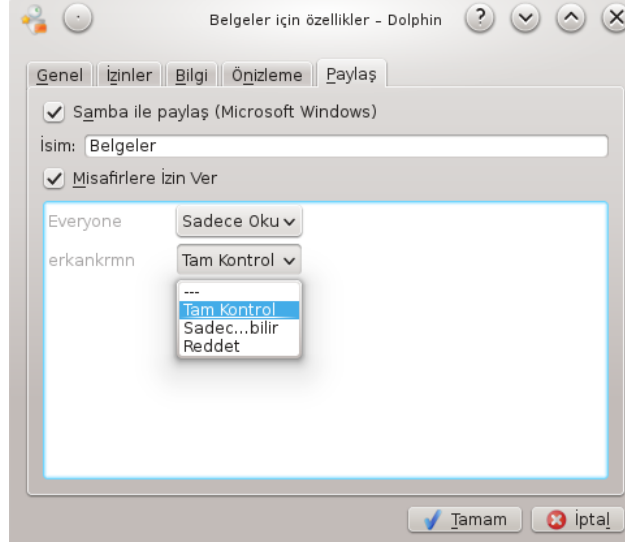


Resim 2.2: Servis yöneticisi penceresi

Dosya veya dizinleri paylaşım açmak için fare ile üzerine sağ tıklanarak özellikler sekmesine tıklanır.



Resim 2.3: Dosya özellikleri

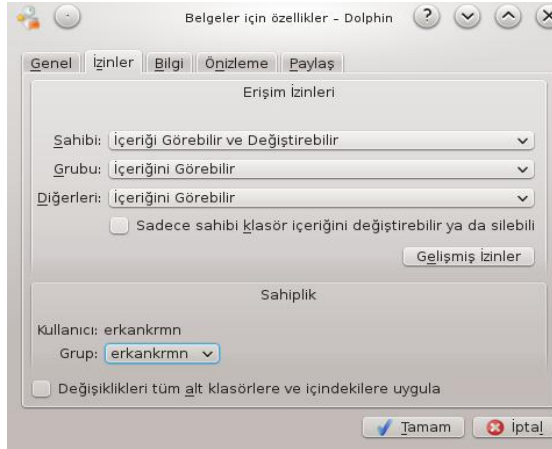


Resim 2.4: Dosya / dizin paylaşım ayarı

Özellikler penceresinde “*Paylaş*” sekmesine tıklanır. Dosyayı başka bir işletim sistemi ile paylaşmak için “*Samba*” servisi kullanılır. Bu paylaşımı sağlamak için “*Samba*” seçeneği tıklanır. Samba (SMB) protokolü lisanslı işletim sistemi ile Açık Kaynak işletim sistemi arasında dosya ve yazıcı paylaşımı için kullanılır.

Paylaş sekmesinde “*Misafirlere İzin Ver*” seçeneği işaretlenirse ağda bulunan diğer kullanıcılara erişim için izin verilir. Bu seçenekte ağdaki diğer kullanıcılara üç şekilde izin verilebilir. Dosyalara tam erişim için “*Tam Kontrol*”, sınırlı erişim için “*Sadece Oku*” ve erişim sağlanmaması için “*Reddet*” seçenekleri seçilebilir.

Açık kaynak işletim sisteminde dosya ve dizin izinleri de erişim açısından önem taşımaktadır. Bu nedenle, “*İzinler*” sekmesine geçilerek buradaki izinlerin diğerleri için görünür veya değiştirilebilir halde olduğu kontrol edilmelidir.



Resim 2.5: İzinler sekmesi ayarı

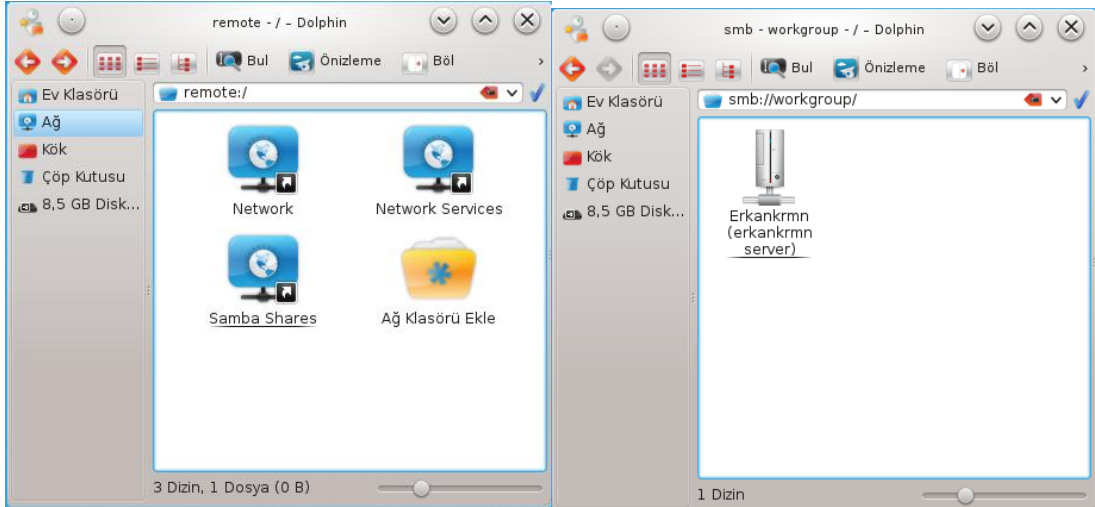
2.2. Paylaşılan Dosyalara Erişim

Paylaşılmış dosyalara erişim sağlamak için dosya yöneticisi (*Dolphin*) açılır. Dosya yöneticisi ilk açıldığında kullanıcının ev klasörü görüntülenir. Bupencerede “Ağ” bölümü tıklanır.



Resim 2.6: Dosya yöneticisi (dolphin)

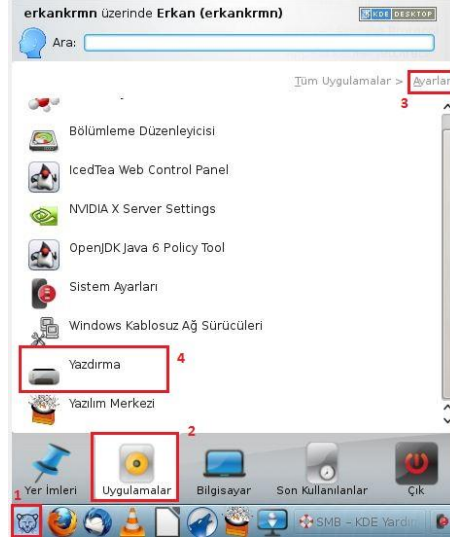
Ağdaki diğer bilgisayarlarda paylaşıma açılan dosya ve dizinlere erişim için “*Samba Shares*” seçilir. Buradan dosya paylaşımında bulunan bilgisayarlara erişim sağlanabilir.



Resim 2.7: Samba shares paylaşılan dosyalara erişim

2.3. Paylaşılan Yazıcılara Erişim

Açık kaynak işletim sisteminde paylaşılan yazıcılara erişim için *Uygulama Çalıştırıcı* → *Uygulamalar* → *Ayarlar* → *Yazdırma* seçeneğine tıklanır.



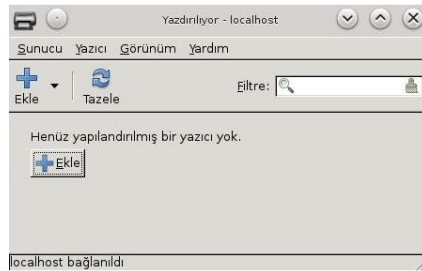
Resim 2.8: Uygulama yöneticisi yazdırma sekmesi

Yazıcı yükleme işlemi için sistem yöneticisi ile giriş yapılması için giriş penceresi ekrana gelir.



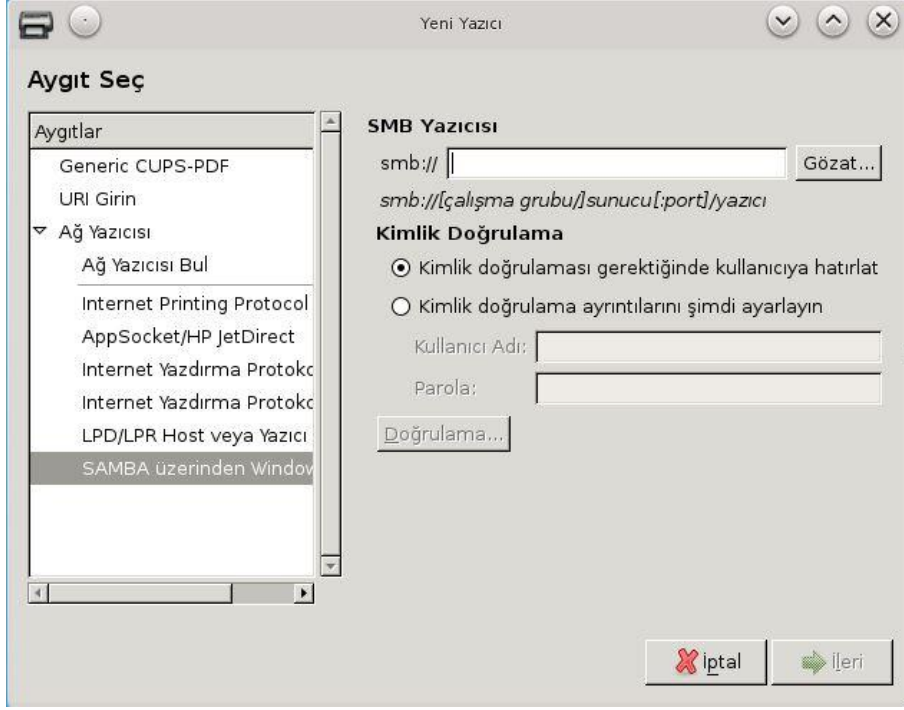
Resim 2.9: Yazıcı ekleme yönetici giriş penceresi

Sisteme tanıtılmış olan yazıcılar, ekrana gelen yazıcılar penceresinde görüntülenir. Sistemde yazıcı yüklü değil ise *Ekle* butonuna basılır.



Resim 2.10: Yazıcılar penceresi

Yeni Yazıcı penceresinden *Yerel Yazıcı* ve *Ağ Yazıcısı* eklenebilir.



Resim 2.11: Yeni yazıcı penceresi

Açık kaynak işletim sistemi yüklü olan bilgisayarlardan paylaşılmış olan yazıcıları eklemek için *Internet Yazdırma Protokolü (ipp)* sekmesi seçilir. Ağa bağlı olan lisanslı işletim sistemi yüklü olan bir bilgisayardan *SAMBA* sekmesi seçilir. SMB yazıcısını ağda bulmak için *Gözet* butonuna tıklanır.



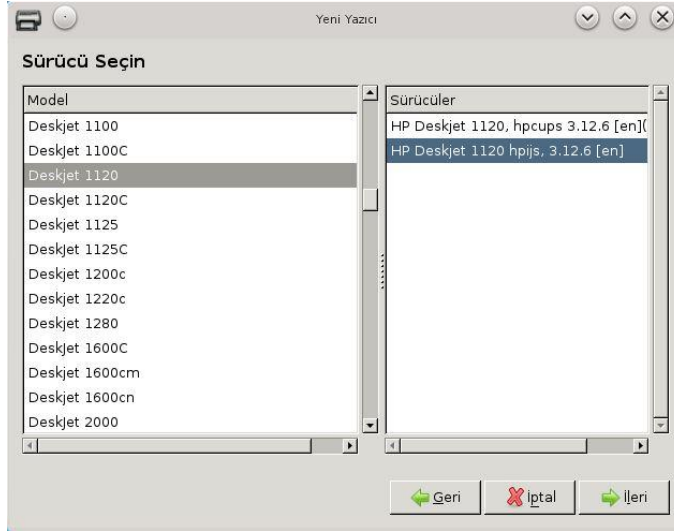
Resim 2.12: SMB gezgin penceresi

SMB Gezgin penceresinde etkin çalışma ağında olan bilgisayarlar listelenir. Bu bilgisayarlardan paylaşılmış yazıcı seçilerek *Tamam* butonuna basılır.



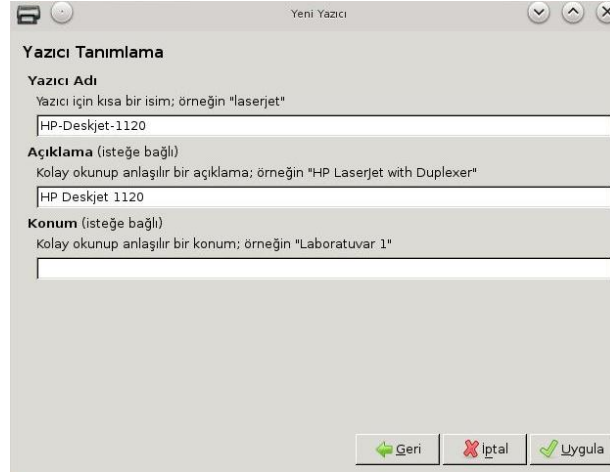
Resim 2.13:Yeni yazıcı penceresi

Ekrana gelen *Yeni Yazıcı* penceresinden yazıcının markası ve modeli seçilir.



Resim 2.14: Yeni yazıcı sürücü yükleme penceresi

Yazıcının markası ve modeli seçildikten sonra yazıcının sistemdeki adını belirlenir.



Resim 2.15: Yazıcı tanımlama penceresi

Yazıcı Tanımlama penceresinde yazıcının adı belirlenerek *Uygulabutonuna* basılır.



Resim 2.16: Yazıcılar penceresi

Uzaktaki yazıcı için sisteme herhangi bir sürücü yüklemeye gerek duyulmaz. Uzaktaki yazıcı bu noktadan itibaren tüm masaüstü uygulamaları tarafından tanınacaktır.

2.4. Dosya Paylaşım Programları

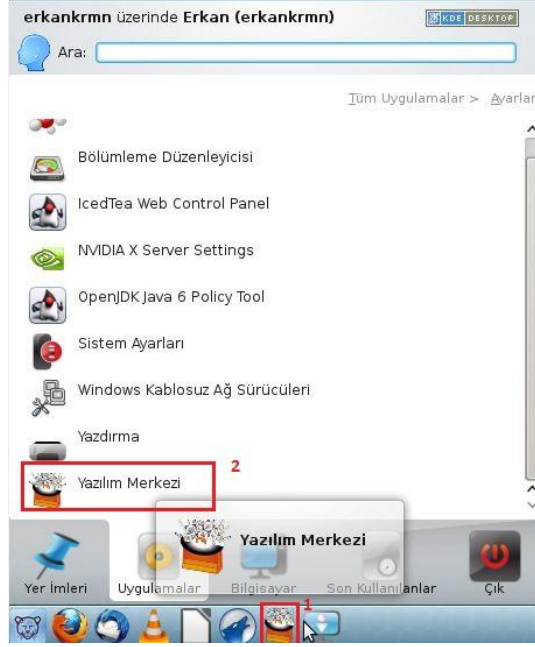
Açık kaynak işletim sistemlerinde yerel ağ üzerinde dosya paylaşım işlemlerinde herhangi bir ek yazılıma ihtiyaç duyulmaz. Geniş ağ olarak tanımlanan internet üzerinde de dosya paylaşımı sağlanabilmektedir. Ancak internet üzerinden dosya paylaşımı için ek yazılımlara gereksinim vardır. En yaygın olarak kullanılan yazılımlar Apollon ve Ktorrent programlarıdır.

Not: Bu programlar yardımıyla yasadışı dosya indirilmesi suçtur. Bu nedenle yalnızca temin edilmesi ve kullanılması suç olmayan dosyalar indirilmelidir.

2.4.1. Apollon

Apollon programı, internet üzerindeki pekçok dosya paylaşım ortamında arama yaparak dosyaların paylaşılabilceği içerisinde kendi istemcisi olan bir uygulamadır. Bu program sayesinde birçok paylaşım ağlarına erişim sağlanabilir.

Apollon programı sistemde standart olarak kurulu olan bir program değildir. Bu programı sisteme eklemek için *Yazılım Merkezi* kullanılır. *Yazılım Merkezi* görev çubuğunda standart olarak bulunur. *Yazılım Merkezi*, *Uygulama Çalıştırıcı*→ *Uygulamalar*→ *Ayarlar*→ *Yazılım Merkezi* ile de uygulamaya açılabilir.



Resim 2.17: Yazılım merkezi çalıştırma

Yazılım merkezinde arama çubuğuna Apollon yazılarak programa erişim sağlanır.



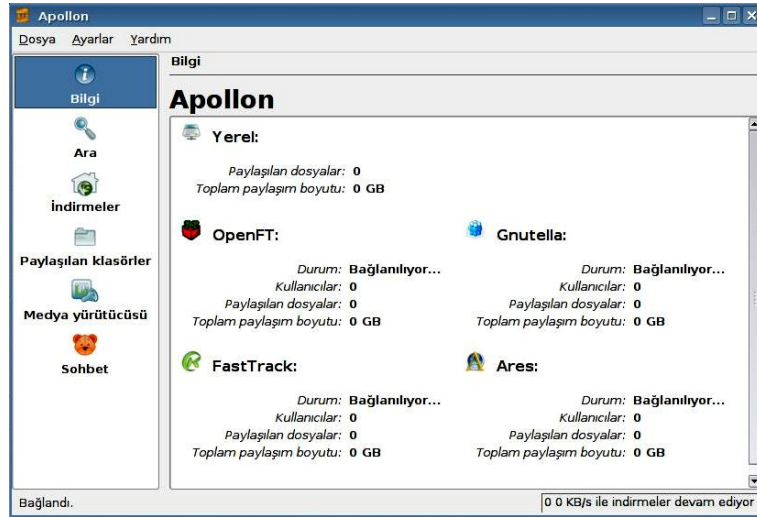
Resim 2.18: Yazılım merkezi

Sisteme yazılımı kurmak için kullanıcı kimlik doğrulaması gerekmektedir. Kullanıcı şifresi yazılarak yazılım kurulumuna izin verilebilir.



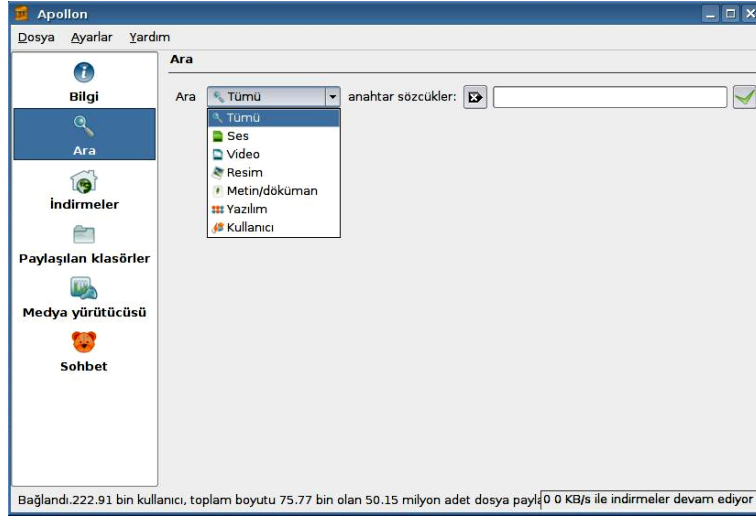
Resim 2.19: Kimlik doğrulama penceresi

Apollon programını sistemde başlatmak için *Uygulama Çalıştırıcı* → *Uygulamalar* → *İnternet* → *Apollonyolu* izlenebilir. Açılan pencerede Apollon'un hangi paylaşım kanallarına bağlı olduğu görülebilir. Ayrıca her servisin kaç kullanıcı içerdiği ya da kaç kullanıcının bağlı olduğu bilgisi de elde edilebilir.



Resim 2.20: Apollon penceresi

Apollon programında arama yapmak için pencerenin sol tarafındaki menüden *Ara* seçeneği seçilir. *Ara* kısmında dosya türü belirlenerek (ses, video, metin, resim, yazılım vb.) ve anahtar sözcük girilerek arama gerçekleştirilebilir. Sonuçlar, pencere üzerinde listelenir. Aranılan dosya seçilerek indirme işlemi başlatılabilir.



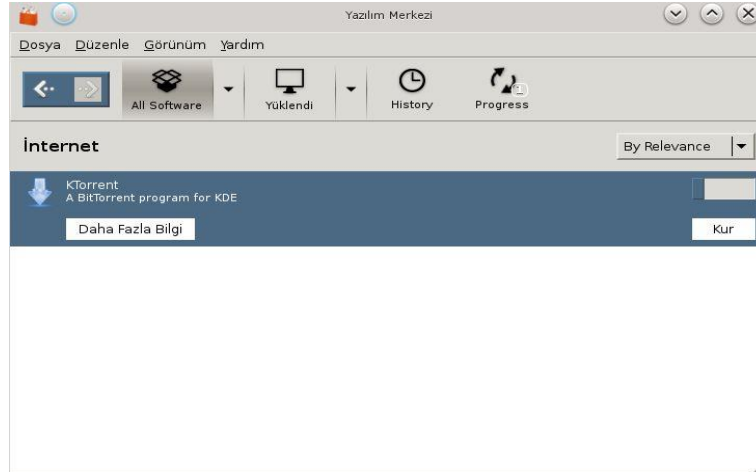
Resim 2.21: Apollon arama bölümü

İndirilen dosyalar öntanımlı olarak masaüstünde açılacak “*SharedFolder*” adlı dizine yapılır. Bu dizin *Ayarlar* menüsünden değiştirilebilir.

2.4.2. KTorrent

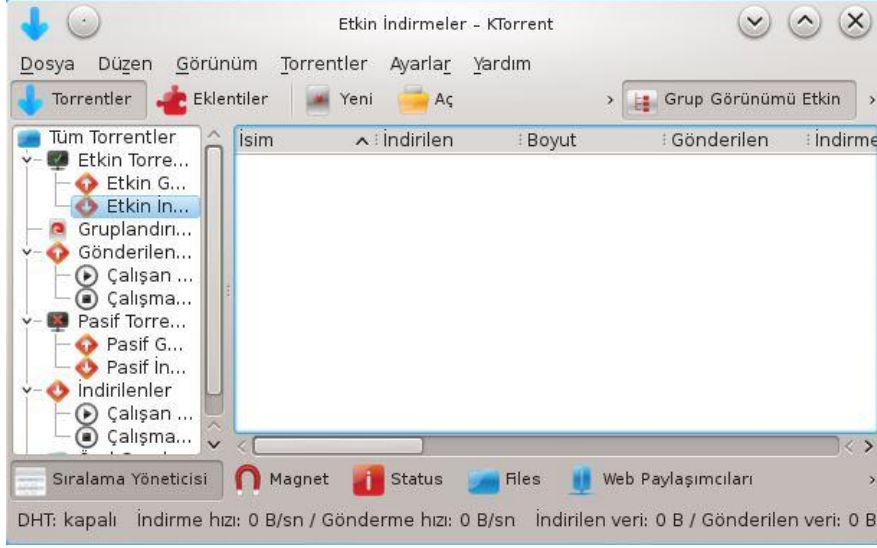
KTorrent programı en yaygın paylaşım programı olan “*bittorrent*” alt yapısını kullanarak yazılım indirilebilir. Bu programda dosya paylaşımında bulunan bir kullanıcının dosyasını birden fazla kullanıcı indirebilir. Dosya indirilirken parça parça indirilir. İndirme işlemi yapan kullanıcılar kendilerinden de dosya parçacıklarını indirilebilir. Birden fazla kaynaktan paylaşım olduğu için indirme işlemi daha hızlı olmaktadır.

KTorrent programı da *Yazılım Merkezi* kullanılarak sisteme kurulur. Yazılım kurulurken kullanıcı kimlik doğrulaması yapılır.



Resim 2.22: KTorrent program kurulum penceresi

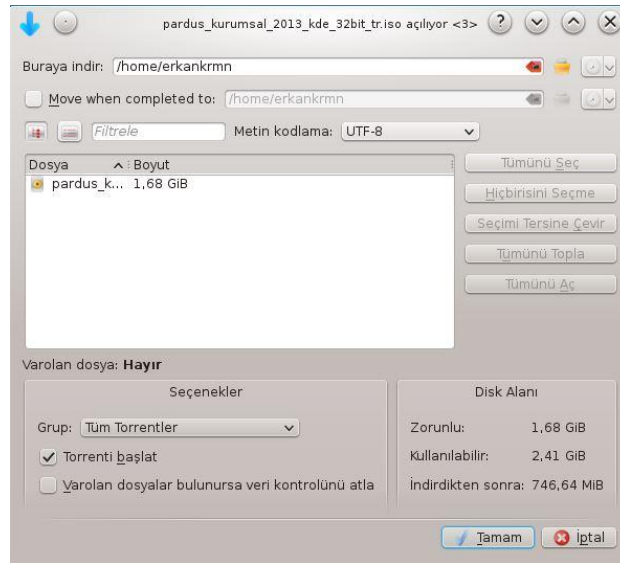
KTorrent programını sistemde başlatmak için *Uygulama Çalıştırıcı*→ *Uygulamalar*→ *İnternet*→ *KTorreyolu*nu izlenebilir.



Resim 2.23:KTorrent penceresi

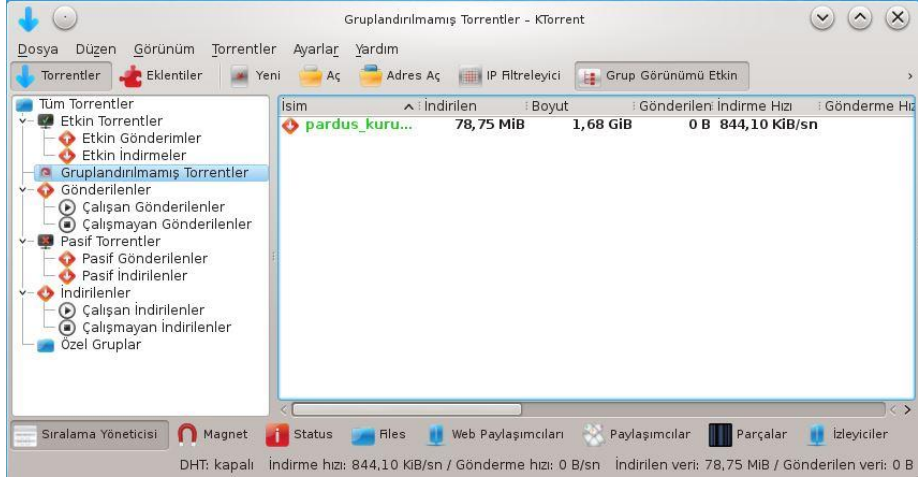
KTorrent programı, kapatılması durumunda,program sistem çubuğunda sağ alt köşede çalışmasına devam eder. Programdan tamamen çıkmak içinKTorrent simgesine sağ tıklayarak **Çık** seçeneği seçilmelidir.

KTorrent programı dosya indirilmesi için internetten Torrent dosyası indirilir. Torrent dosyasına çift tıklanarak dosya KTorrent programında çalıştırılır. Örneğin Açık kaynak işletim sisteminin son sürümünün Torrent dosyası indirilerek işletim sisteminin kurulum dosyası bilgisayara indirilebilir.



Resim 2.24: Torrent dosyası çalıştırma

KTorrent programı açık olduğun sürece diğer kullanıcılarda dosya paylaşımında bulunur. Ayrıca KTorrent programı kullanıcısı da diğer kullanıcılar ile dosya paylaşımında bulunur. Bu tür programlar, bilgisayarda bir dizini internet üzerinde paylaşımına açar. Böylece bir başka kullanıcıdan indirilmiş dosyanın kaynağı da KTorrent programı kullanıcısı olur. Dosya indirirken aynı zamanda sistemde bulunan kopyalar diğer KTorrent programı kullanıcıları tarafından alınabilir. Bu nedenle bağlantı hızı yavaşlayabilir.



Resim 2.25: KTorrent program penceresi

UYGULAMA FAALİYETİ

Bu uygulama faaliyetinde, açık kaynak işletim sistemi dosya ve izin paylaşım işlemlerini gerçekleştirmeyi öğreneceksiniz.

İşlem Basamakları	Öneriler
➤ Ev dizini içerisine yeni bir dizin oluşturunuz.	➤ Bu dizine Paylas ismini veriniz.
➤ Dizinin erişim haklarını ayarlayınız.	➤ Dizine sağ tıklayarak Özellikler seçeneğini seçin ve diğer kullanıcılar için hakları düzenleyebilirsiniz.
➤ Samba protokolü üzerinden paylaşım açtığınız dizine erişim sağlayınız.	➤ Dosya yöneticisinden Ağbölümünden dosyaya erişim sağlayabilirsiniz.
➤ Apollonpaylaşım programı ile açık kaynak işletim sistemi kurulumdosyasını indiriniz.	➤ Uygulama Çalıştırıcı→ Uygulamalar→ İnternet→ Apollonyolunu izleyerek programı çalıştırabilirsiniz. ➤ Ara çubuğunu kullanarak dosyayı aratınız.
➤ KTorrentpaylaşım programı ile açık kaynak işletim sistemi kurulumdosyasını indiriniz.	➤ İnterneti kullanarak açık kaynak işletim sistemi kurulum dosyasının Torrent dosyasını indirebilir ve çalıştırabilirsiniz.

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanmadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Kullanıcının ev klasöründe <i>Paylas</i> dizini oluşturduunuzmu?		
2. Dizinin erişim haklarını ayarladınız mı?		
3. Paylaşım açılan dizine erişim sağladınız mı?		
4. Apollonpaylaşım programı ile dosya indirme işlemi yaptınız mı?		
5. KTorrentpaylaşım programı ile indirme işlemi dosya indirme işlemi yaptınız mı?		

DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. () Ağa bağlı olan bir bilgisayarda, kullanıcının dosyaları ve dizinleri ağa bağlı olan başka bir kullanıcı tarafından kullanılamaz.
2. () Dosyayı başka bir işletim sistemi ile paylaşmak için “*Samba*” servisi kullanılır.
3. () Açık kaynak işletim sisteminde yazıcı paylaşımı **yalnızca** ağa bağlı açık kaynak işletim sistemi kurulu makineler arasında gerçekleştirilebilir.
4. () Ağ üzerinden paylaşılan yazıcı sisteme tanıtıldığında sürücü yükleme işlemi yapılmalıdır.
5. () KTorrent programı ile yerel ağda dosya paylaşımı yapılabilir.
6. () Apollon programı ile internet üzerinden dosya indirme işlemi yapılabilir.
7. () KTorrent programı ile dosya indirme ve gönderme işlemi yapılabilir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi işletim sistemi katmanlarındandır?
A) Sistem çağruları
B) Kabuk
C) Komut satırı
D) Çekirdek
2. Aşağıdakilerden hangisi komut kullanırken dikkat edilmesi gereken kurallardan değildir?
A) Komut kullanımında küçük/büyük harf ayırımına dikkat edilmelidir.
B) Komut kullanımında seçeneklerden önce “.” kullanılır.
C) Komut argümanlarında “-” işareti kullanılmaz.
D) Komut kullanımında “?” işareti tek bir karakter yerine geçer.
3. Aşağıdakilerden hangi klavye tuş kombinasyonu çalışmakta olan komutu durdurur?
A) Ctrl+C
B) Ctrl+U
C) Ctrl+S
D) Ctrl+Q
4. Aşağıdakilerden işaretlerden hangisi oturum açan kullanıcının ev dizinini gösterir?
A) -
B) /
C) ?
D) ~
5. Aşağıdaki listeleme seçeneklerinden hangisi gizli dosyaları listelemek için kullanılır?
A) -a
B) -l
C) -h
D) -R
6. Aşağıdaki komutlardan hangisi izin oluşturmak için kullanılır?
A) cd
B) ls
C) mkdir
D) find
7. Aşağıdaki komutlardan hangisi dosya veya izin silmek için kullanılır?
A) ln
B) rm
C) mv
D) cp

8. Aşağıdakilerden komutlardan hangisi dosya içeriğini liste halinde görüntülemek için kullanılır?
A) grep
B) mkdir
C) cat
D) host
9. Aşağıdakilerden komutlardan hangisi dosyaları ve dizinleri arşivlemek için kullanılmaz?
A) gzip
B) gunzip
C) rar
D) tar
10. Aşağıdakilerden komutlardan hangisi kullanıcı değiştirmek için kullanılır?
A) su
B) logname
C) host
D) who
11. Aşağıdaki komutlardan hangisi script dosyasını çalıştırmak için komut satırında izin verir?
A) ch
B) logname
C) chmod
D) who
12. Aşağıdaki değişken tanımlamalarından hangisi doğrudur?
A) 22a
B) sayı
C) ikinci sayı
D) ilk_sayi

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

13. () Açık kaynak işletim sisteminde dosya ve dizin izinleri de erişim açısından önem taşımaktadır.
14. () Paylaştırılmış dosyalara erişim sağlamak için dosya yöneticisi (*Dolphin*) açılır.
15. () Paylaştırılmış yazıcılara erişim sadece açık kaynak işletim sistemlerinden erişim izni verilebilir.
16. () Geniş ağ üzerinden (*internet*) dosya paylaşımında sadece *Apollon* ve *KTorrent* programları kullanılır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Doğru
4	Doğru
5	Yanlış
6	D
7	C
8	B
9	C
10	D
11	C

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Yanlış
4	Yanlış
5	Yanlış
6	Doğru
7	Doğru

MODÜL DEĞERLENDİRME'NİN CEVAP ANAHTARI

1	C
2	B
3	A
4	D
5	A
6	C
7	B
8	C
9	C
10	A
11	C
12	D
13	Doğru
14	Doğru
15	Yanlış
16	Yanlış

KAYNAKÇA

- VARDAR M. Ali, **Linux Altında Programlama**, Papatya Yayıncılık Eğitim A.Ş. İstanbul, 2002.
- UYARER Talat, **Her Yönüyle Pardus**, Pusula Yayıncılık, Ankara, 2007.
- ÖZBİLEN Alper, **Linux Sistem ve Ağ Yönetimi**, Pusula Yayıncılık, İstanbul, 2013.
- ÇETİN Görkem, **Linux Doktoru**, Seçkin Yayıncılık, Ankara, 2003.
- http://www.e-kitap.org/linux_unix_komutlar.html, (22.07.20013 / 09.15)