

# Sistem Programlama

## Ders 2

Doç. Dr. Mehmet Dinçer Erbaş  
Bolu Abant İzzet Baysal Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

# UNIX

- UNIX sistemine genel bakış
  - UNIX, çoklu kullanıcıya ait çoklu programları aynı anda yönetebilmek amacıyla oluşturulmuş etkileşimli bir sistemdir.
    - Programcılar tarafından programcılar için oluşturulmuştur.
  - Bir programcının beklentilerine cevap verecek şekilde geliştirilmiştir.
    - Basit, zarif ve tutarlıdır.
      - Bir dosya sadece byte topluluğudur.
    - Güçlü ve esnektir.
      - Bir sistem gerektiğinde birbirine bağlantılı şekilde karmaşık sorunları çözebilecek basit parçalardan oluşmalıdır.

# UNIX

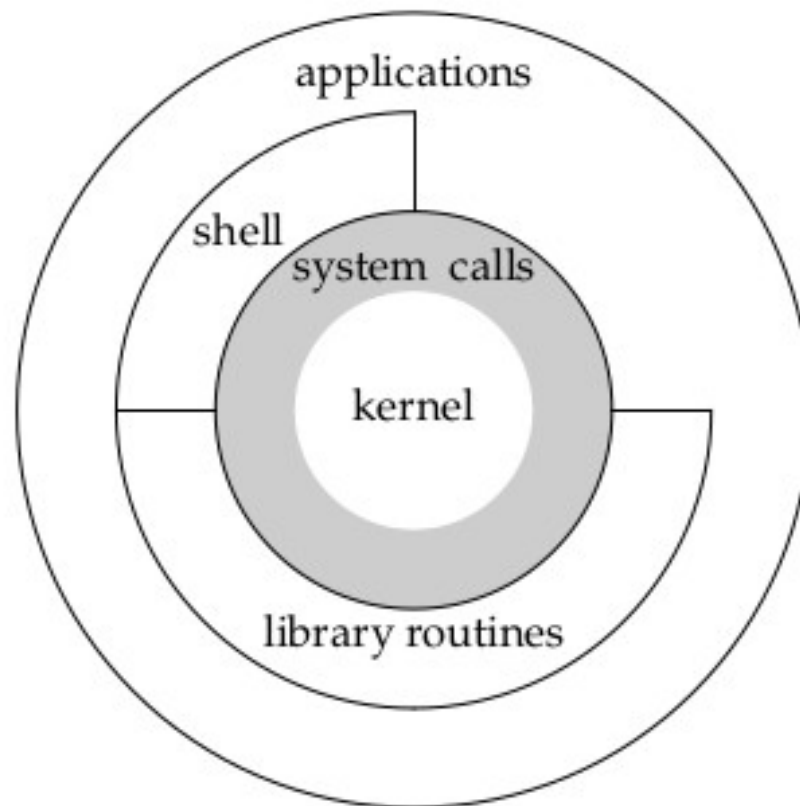
- UNIX sistemine genel bakış
  - Bir programcının beklentilerine cevap verecek şekilde geliştirilmiştir.
    - Güçlü ve esnektir.
      - Sistemin parçası olan programlar sadece bir işe odaklanmalıdır ve o işi en iyi şekilde yapmalıdır.
    - Gereksiz ayrıntı olmamalıdır.
      - Kopyalama işlemi için “cp” yazmak yeterli ise “copy” yazmaya gerek yoktur.
      - grep ard f

# UNIX

- UNIX programları
  - Her biri basit işlemler yapar.
  - Kullanma kılavuzları bulunur.
    - `man ls`
  - Ne yaptıkları kolaylıkla tahmin edilebilir.
    - `ls A*`
    - `rm A*`
  - `stdin`'den girdi alırlar.
  - `stdout`'a çıktı oluştururlar.
  - Anlamlı hata mesajları oluştururlar.
  - Anlamlı çıkış kodları vardır.

# UNIX

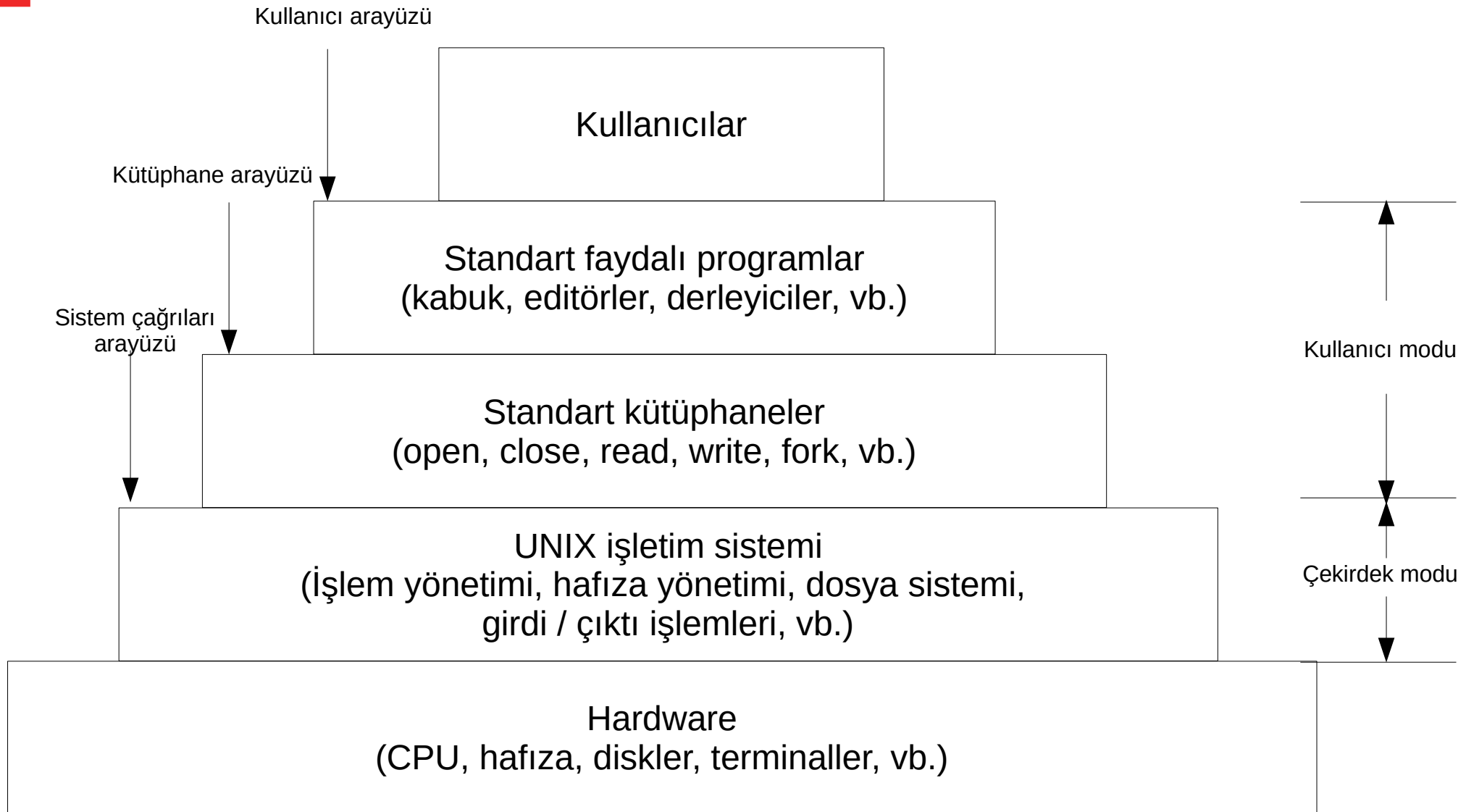
- UNIX sisteminin mimari yapısı



# UNIX

- UNIX mimarisi
  - Genel anlamıyla işletim sistemleri bilgisayara ait donanım kaynaklarını yönetir ve programların çalışabileceği ortamı sağlar.
    - Genellikle bu işlemleri yapan yazılıma çekirdek (kernel) ismini veriyoruz.
      - İşletim sisteminin bu parçası görece olarak küçüktür ve sistemin merkezinde bulunur.
  - Çekirdekte işlem yapmak için sistem çağrıları arayüz olarak kullanılır.
  - Çok kullanılan fonksiyonların bulunduğu kütüphaneler sistem çağrıları arayüzünü kullanır.
    - İstenirse direk olarak sistem çağrıları veya kütüphanlere kullanılabilir.
  - Kabuk (shell) başka programları çalıştırabilmek için arayüz sağlayan özel bir programdır.

# UNIX



# UNIX

- Giriş işlemi
  - UNIX sistemine giriş yapabilmek için sistem üzerinde tanımlanmış bir hesaba sahip olmanız gerekir.
  - Her hesaba ait bir kullanıcı adı ve parola bulunmaktadır.
  - Sisteme girerken önce kullanıcı ismi daha sonra parola girilmelidir.
  - Bunun sonrasında sistem parola dosyasını kontrol eder
    - Genellikle bu dosya */etc/passwd* dosyasıdır.
  - Bu dosya içeriğindeki her girdi sekiz farklı alan içerir.
    - Giriş ismi, şifrelenmiş parola, kullanıcı numarası, grup numarası, bir yorum alanı, giriş dizini, ve kabuk programı.
      - mderbas:x:1000:1000:Mehmet Dincer Erbas,,,:/home/nderbas:/bin/bash
  - Bütün modern sistemler şifrelenmiş parolaları başka bir dosyaya taşımıştır.



# UNIX

- UNIX kabuğu
  - Kabuk kullanıcıdan girdileri alan ve verilen emirleri gerçekleştiren bir komut satırı yorumlayıcısıdır.
  - Kullanıcı girdisi terminal yoluyla (etkileşimli kabuk) veya bir dosyadan (İng: script) alınabilir.

İsim	Yol	FreeBSD 5.2.1	Linux 2.4.22	Mac OS X 10.3	Solaris 9
Bourne Shell	<i>/bin/sh</i>	+	bash'e yönlendirilmiş	bash'e yönlendirilmiş	+
Bourne-again shell	<i>/bin/bash</i>	opsiyonel	+	+	+
C shell	<i>/bin/csh</i>	tcsh'ye yönlendirilmiş	tcsh'ye yönlendirilmiş	tcsh'ye yönlendirilmiş	+
Korn shell	<i>/bin/ksh</i>				+
TENEX C shell	<i>/bin/tcsh</i>	+	+	+	+



# UNIX

- Unix kabuğu
  - Sistem parola dosyasından hangi kabuk versiyonun kullanılacağını öğrenir.
  - Bourne kabuğu versiyon 7'den beri kullanılmaktadır ve neredeyse her Unix versiyonunda bulunur.
    - Bell laboratuvarlarında Steve Bourne tarafından geliştirilmiştir.
  - C kabuğu bütün BSD yayımlarında bulunur
    - Kontrol akışı C diline benzer.
    - Bazı ek özellikleri bulunur: işlem kontrolü, işlem tarihçesi ve emir satırı düzenleme



# UNIX

- Unix kabuğu
  - Korn kabuğu Bourne kabuğunun devamı niteliğindedir ve SVR4 versiyonu ile geliştirilmiştir.
    - Belirttiğimiz diğer iki kabuk kadar yaygın değildir.
    - Bourne kabuğu ile uyumludur ve işlem kontrolü, emir satırı düzenleme gibi fonksiyonlar içerir.
  - Bourne-again kabuğu bütün Linux versiyonlarında bulunan GNU kabuğudur.
    - Bourne kabuğu ile uyumludur ve POSIX standartlarına uyar.
    - C kabuğunun ve Korn kabuğunun bazı özelliklerini içerir.

# UNIX

- Unix kabuğu
  - Tenex C kabuğu C kabuğunun geliştirilmiş bir versiyonudur.
    - TENEX işletim sisteminden birçok özelliği bünyesinde bulundurur (örneğin emir tamamlama).
  - Linux Bourne-again kabuğunu varsayılan kabuk olarak kullanır.
  - FreeBSD ve Mac OS X Tenex C kabuğunu kullanırlar, ancak yönetici kabuk işlemleri için Bourne kabuğunu kullanırlar.
  - Solaris belirtilen bütün kabukları sağlar.
  - İnternette belirtilen kabukların hemen hemen hepsinin serbest bağlantı noktaları bulunabilir.

# UNIX

- Unix kabuğu
  - Kabuk çalıştırıldığında kendisini başlatır, daha sonra bir sorgu karakteri ekrana basarak kullanıcı girdisi bekler.
    - Sorgu karakteri genellikle % veya \$ karakterleridir.
  - Kullanıcı emir satırına bir şeyler yazdığında
    - Kabuk ilk kelimeyi alır.
    - İlk kelime çalıştırılacak programın ismi olarak kabul edilir.
    - Bu programı arar.
    - Program bulunursa çalıştırılır.
  - Program çalıştığında kabuk program bitene kadar kendisini durdurur ve program bittiğinde bir sonraki komutu bekler.

# UNIX

- Komutlar argüman alabilirler. Bu argümanlar character dizisi olarak programa gönderilir.
  - `cp src dest`
- Argümanlar her zaman dosya ismi olmak zorunda değildir.
  - `head -20 file`
- Çizgi karakteri muallaklığı gidermek için kullanılır.
  - `head 20 file`
- Kabuk özel karakterleri kabul eder.
  - `ls *.c`
- Köşeli parantez içerisinde bulunan karakterlerin biri seçilir.
  - `ls [ape]*`

# UNIX

- Kabuk gibi programlar terminal açmadan yazma ve okuma yapabilirler.
- Kabuk standart girdi (okuma için), standart çıktı (yazma için) ve standart hata (hata mesajları yazabilmek için) dosyalarına erişim sahibidir.
  - Kabuk terminali için bu üç dosya varsayılan olarak bulunur.
- Birçok Unix programı standart girdiden okur ve standart çıktıyı yazar.
  - sort
    - Terminalden girilen girdileri alır, CTRL+D ile girdi sonlanır, girdiler sıralanır ve ekrana yazdırılır.
- Standart girdiden okuyan, girdiler üzerinde bazı işlemler yapan ve sonuçları standart çıktıya yazan programlara filtre denir.

# UNIX

- Standart girdi ve standart çıktı başka dosyalara yönlendirilebilir.
  - Standart girdiyi yönlendirmek için “küçüktür” işareti kullanılır (<).
  - Standart çıktıyı yönlendirmek için “büyüktür” işareti kullanılır (>).
  - `sort < in > out`
- Aynı satırda birden fazla işlem yapılabilir.
  - `sort <in > temp; head -30 <temp; rm temp`
- Boru (ing: pipe) sembolü ile birbirini takip eden işlemler yapılabilir.
  - `sort <in | head -30`
  - `grep ter *.t | sort | head -20 | tail -5 >foo`





# UNIX

- Yararlı Unix programları 6 kategoriye ayrılabilir.
  - Dosya ve klasör düzenleme komutları
  - Filtreler
  - Editörler ve derleyiciler gibi programlama araçları
  - Metin işleme
  - Sistem yönetimi
  - Diğer
- POSIX 1003.2 standartlar yüzden az komutun yazım ve anlamını tanımlar. Bu komutların büyük çoğunluğu ilk üç kategoridedir.

# UNIX

- Bazı örnek komutlar
  - \$ date
    - Sistemin zamanını hafta günü, ay, gün, saat, saat dilimi ve yıl olarak yazdırır.
  - \$ who
    - Sisteme giriş yapmış kullanıcıları listeler.
  - \$ whoami
    - Kendi hesabınız hakkında bilgi verir.
  - \$ ls
    - Bulunduğunuz klasördeki dosyaları listeler.

# UNIX

- Bazı örnek komutlar
  - \$ cd
    - Klasör değiştir.
  - \$ mkdir
    - Klasör oluşturur.
  - \$ rmdir
    - Klasör siler.
  - \$ cp
    - Dosya kopyalar.
  - \$ rm
    - Dosya siler.

# UNIX

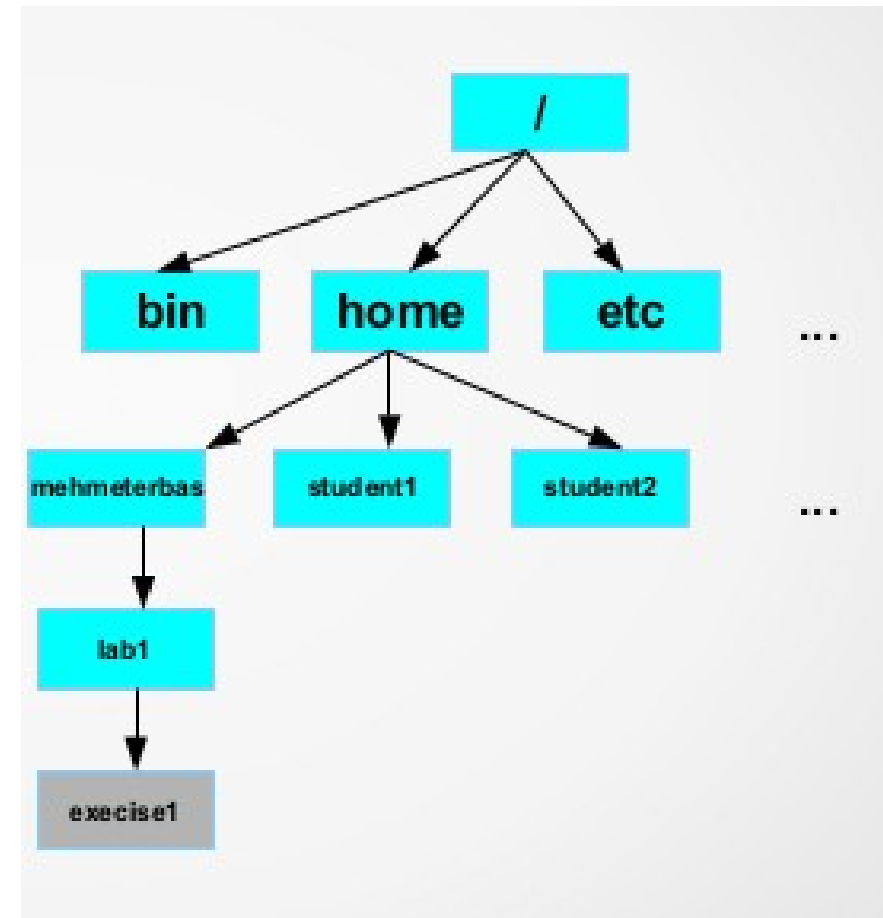
- Bazı yararlı programlar
  - \$ cat
    - Dosya içeriğini gösterir.
  - \$ head
    - Dosyanın ilk 10 satırını gösterir.
  - \$ less
    - Dosyanın son 10 satırını gösterir.
  - \$ grep
    - Dosya içerisinde metin arar, metnin bulunduğu satırları gösterir.

# Unix sistemine genel bakış

- Dosya sistemi
  - Unix sistemi dosyaları ve klasörleri hiyerarşik olarak düzenler.
    - Her şey kök klasörü ile başlar: /
  - Klasör, diğer dosyaları içerebilen özel bir dosya tipidir.
    - Bir klasörü ismi olan ve içerdiği diğer dosyalara dair bilgiler içeren özel bir yapı olarak düşünebiliriz.
  - Dosyanın içerdiği özelliklerden bazıları şunlardır:
    - Dosya tipi (normal dosya, klasör gibi).
    - Dosyanın büyüklüğü
    - Dosyanın sahibi
    - Dosya üzerindeki izinler
    - Dosyanın son değişme zamanı
  - stat ve fstat komutları dosyanın özellikleri ile ilgili bilgi verir.

# Dosyalar ve klasörler

- Bir klasör için dosya isimleri bulunur.
  - Bir dosya ismi içerisinde / karakteri ve null karakteri bulunamaz.
- Yeni bir klasör oluşturulduğunda iki dosya otomatik olarak oluşturulur.
  - . (dot) ve .. (dot-dot)
- Günümüzde neredeyse tüm Unix sistemleri 255 karakterlik dosya isimlerine izin verir.



# Dosyalar ve klasörler

- Yoladı
  - Bir veya daha fazla birbirini takip eden dosya ismi, aralarında / karakteri ile, ve başlangıçta opsiyonel olarak / karakteri olan bir metin yoladı (ing: pathname) olarak adlandırılır.
    - / karakteri ile başlayan yoladları mutlak yoladı olarak adlandırılır.
      - `cd /home/mehmeterbas/lab1`
    - Diğerleri ise görece yoladı olarak adlandırılır,
      - `cd mehmeterbas/lab1`

# Dosyalar ve klasörler

- Örnek1: ls1.c
- Örnek program hakkında notlar
  - apue.h: Bu dosya ders boyunca kitabımızda göreceğimiz örnekler için gerekli standart kütüphaneleri ve sabit değerleri içerir.
  - Programdaki main fonksiyonunun deklarasyonu ISO C standartlarına uygundur.
  - Program kullanıcıdan bir argüman almakta (argv[1]) ve bu argümanı dosyaları listelenecek klasör adı olarak kullanmaktadır.
  - Klasör işlemleri yapabilmek için opendir, readdir ve closedir fonksiyonlarını kullanıyoruz.



# Dosyalar ve klasörler

- Örnek program hakkında notlar
  - opendir fonksiyonu dir yapısında bir işaretçi döner.
    - Daha sonra readdir fonksiyonu ile bir döngü içerisinde klasördeki kayıtları okur.
    - readdir fonksiyonu bütün kayıtları okuduğunda null işaretçi döner ve bu sayede döngü sonlanır.
    - Her kayıttaki dirent yapısında bulunan d\_name bilgisi kullanıcıya gösterilir.
  - Ayrıca programda uygun hata mesajları tanımlanmıştır.
  - Program sonlandığında exit fonksiyonu 0 argümanı ile çağırılmıştır.

# Dosyalar ve klasörler

- Çalışma klasörü
  - Her işleme ait bir çalışma klasörü mevcuttur.
    - Bu işlem için bütün görece yolları bu klasörden hareketle hedefine gider.
  - Bir işlem chdir fonksiyonunu kullanarak çalışma klasörünü değiştirebilir.
  - doc / memo / joe
    - Bu yoladı joe isminde bir dosyayı hedeflemektedir. Bu dosya memo isimli bir klasördedir, memo klasörü doc isimli bir klasördedir ve doc klasörü çalışma klasöründe bulunmalıdır.
  - / usr / lib / lint
    - Bu yoladı lint isimli bir dosyayı hedeflemektedir. lint dosyası lib isimli bir klasördedir, lib klasörü ise usr klasöründedir. usr klasörü ise işletim siseminin kök (root) klasöründedir.
  - Giriş dizini
    - Sisteme giriş yaptığımızda çalışma klasörü olarak giriş dizinine gireriz.
      - Giriş dizini parola dosyasında bulunan kaydımızdan öğrenilir.