### Dosya Sistemi

### Bilgilerin Uzun Vadeli Saklanması

- saklanacak veriler çok fazla olabilir
- veriler proses sonlandıktan sonra da kaybolmamalı
- bilgiye prosesler ortak olarak ulaşabilmeli

## Dosya Sistemi Görevleri

- · dosya isimlendirme
- · dosyalara erişim
- dosyaların kullanımı
- · koruma ve paylaşım
- · gerçekleme

# Dosya Sistemi Özellikleri

- kullanıcı açısından
  - dosyaların içerikleri
  - dosya isimleri
  - dosya koruma ve paylaşma
  - dosya işlemleri

- tasarımcı açısından
  - dosyaların gerçeklenmesi
  - boş alanların tutulması
  - mantıksal blok boyu
  - **–** ....

⇒ Kullanıcı arayüzü

⇒ Dosya sistemi gerçeklemesi

### Dosya Tipleri

- Dosyalar
  - ASCII dosyalar
  - ikili dosyalar
- Kataloglar
  - çoğu işletim sisteminde katalog = dosya

### Dosya İçi Erişim

- sıralı erişim
- · rasgele erişim

## Dosyaların Özellikleri (Attribute)

- erişim hakları
- parola
- yaratıcı
- sahibi
- salt oku bayrağı
- saklı bayrağı
- sistem bayrağı
- arşiv bayrağı
- ASCII/ikili dosya bayrağı
- rasgele erişim bayrağı
- geçici bayrağı

- kilit bayrakları
- kayıt uzunluğu
- anahtar konumu
- anahtar uzunluğu
- yaratılma zamanı
- son erişim zamanı
- son değişiklik zamanı
- dosya boyu
- · maksimum dosya boyu

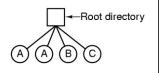
# Dosya İşlemleri

- yaratma / silme
- · isim değiştirme
- · açma / kapama
- yazma / okuma / ekleme
- · dosya işaretçisi konumlandırma
- özellik sorgulama / değiştirme
- ⇒ sistem çağrıları ile (*open, creat, read, write, close,* .....)

8

#### Tek Seviyeli Katalog Sistemleri

- · tek seviyeli katalog
- hızlı erişim
- çok kullanıcılı sistemlere uygun değil
  - farklı kullanıcılar aynı isimli dosya yaratırsa sorun olur
- günümüzde gömülü sistemlerde uygulanabilir
  - örneğin arabada kullanıcı profilleri saklanması

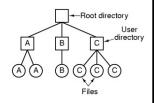


(**Not:** örnekte dosya isimleri değil sahipleri gösterilmiştir- 3 farklı sahibe ait 4 dosya.)

9

### İki Seviyeli Katalog Sistemleri

- her kullanıcıya ayrı katalog
  - kullanıcıların aynı isimli dosya sorunu çözülür
- örneğin çok kullanıcılı kişisel bilgisayarlarda
- sisteme kullanıcı adı ve parola ile girme kavramı uygulanabilir

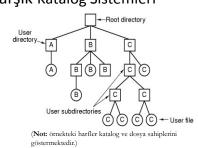


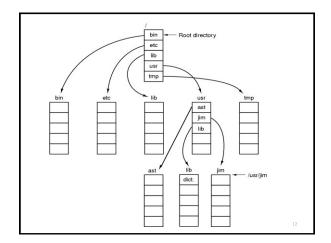
(**Not:** örnekteki harfler katalog ve dosya sahiplerini göstermektedir.)

10

# Hiyerarşik Katalog Sistemleri

- kullanıcıların dosyalarını mantıksal olarak gruplama isteği
- · katalog ağacı
- modern
   işletim
   sistemlerindek
   i yapı





### Katalog İşlemleri

- · yaratma / silme
- kataloğu açma / kapama
- · kataloğu okuma
  - örneğin dosyaların listelenmesi
  - okumadan önce açılması lazım
- · isim değiştirme
- bağla / kopar (link/unlink)
  - UNIX'te dosya silmeye özdeş

Dosya Sistemi Gerçeklemesi

Entire disk

Partition table

Disk partition

Disk partition

Root dir Files and directories

Örnek Dosya Sistemi Yapısı

#### Dosya Sistemi Gerçeklemesi (1)

- sürekli yer ayırma ile gerçekleme
  - dosyanın ilk bloğunun adresi ve blok sayısı tutulur
  - avantajları
    - · basit gerçekleme
    - daha etkin *okuma* işlemi
  - sorunları
    - diskte parçalanma (fragmentation)
    - sıkıştırma maaliyeti yüksek
    - · boşluk listesi tutulmalı
      - dosya boyu en baştan bilinmeli ve sonradan değişemez
      - dosyaların maksimum boyları kısıtlı
  - CD-ROM dosya sistemlerine uygun (tek yazımlık)

Dosya Sistemi Gerçeklemesi (1)

File A
(4 blocks)
(6 blocks)
(12 blocks)
(3 blocks)
(3 blocks)
(3 blocks)
(6 blocks)
(6 blocks)
(8 blocks)
(9 blocks)
(10 blocks)
(11 blocks)
(12 blocks)
(12 blocks)
(13 blocks)
(14 blocks)
(15 blocks)
(16 blocks)
(17 blocks)
(18 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)
(19 blocks)

,

#### Dosya Sistemi Gerçeklemesi (2)

#### · bağlantılı listeler kullanarak gerçekleme

- her bloğun ilk sözcüğü bir sonraki bloğa işaretçi
- parçalanma yok (yanlız son blokta iç parçalanma)
- yanlız dosyanın ilk bloğunun adresi tutulur
- dosyadaki verilere erişim
  - sıralı erişim kolay
  - rasgele erişim zor-zaman alır
- bloktaki veri boyu 2'nin kuvveti olmayabilir
  - okumada bloklar genelde 2'nin kuvveti boyunda

Dosya Sistemi Gerçeklemesi (2)

File A

File A

File B

Physical 4 7 2 10 12

Physical 6 3 11 14

Dosya bloklarının bağlantılı liste yapısında tutulması

#### Dosya Sistemi Gerçeklemesi (3)

#### · bellekte dosya tabloları ile gerçekleme

- işaretçiler diskteki bloklarda değil bellekte tabloda tutulur
- FAT (File Allocation Table)
- rasgele erişim daha kolay
  - tablo bellekte-erişim hızlı
- başlangıç bloğunun bilinmesi yeterli
- tüm tablo bellekte yer almalı-bellekte alan kaplar
- tablo boyu disk boyuna bağlı
  - örnek: 20 GB disk ve blok boyu 1K olsun: tabloda 20 milyon en az 3 sekizli boyunda kayıt gerekli (20MB)

5

#### Dosya Sistemi Gerçeklemesi (4)

- her dosyaya ilişkin bir i-node (index-node)
  - dosyanın özellikleri
  - dosyanın bloklarının disk adresleri
- sadece açık dosyaların i-node yapıları bellekte tutulur
  - toplam bellek alanı bir anda açık olmasına izin verilen maksimum dosya sayısı ile orantılı
- basit bir yapıda dosyanın maksimum blok sayısı üzerinde kısıtlama söz konusu
  - çözüm: son gözü blok işaretçileri içeren ek tabloya işaretçi

File Attributes
Address of disk block 0
Address of disk block 1
Address of disk block 2
Address of disk block 3
Address of disk block 4
Address of disk block 5
Address of disk block 6
Address of disk block 6
Address of disk block 7
Address of disk block 7
Address of disk block 7
Address of disk block 7

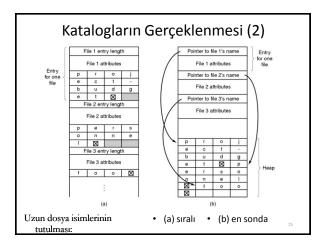
## Katalogların Gerçeklenmesi (1)

- dosya adı ile diskteki bloklar arasında ilişki kurulması
- dosya özellikleri nerede tutulmalı?
  - katalog yapısı içinde?
    - (a) katalog yapısı: sabit boyda dosya adı, dosya özellikleri, disk blokları adresleri
      - MS-DOS / Windows
    - (b) katalog yapısı: dosya adı, i-node numarası
      - dosya özellikleri i-node içinde
      - UNIX
- güncel işletim sistemlerinde dosya isimleri uzun olabilir
  - dosya ismi için sabit bir üst sınır tanımla
    - · gereksiz yer kaybı

Katalogların Gerçeklenmesi (1)

games attributes games mall attributes work attributes (a)

(a) Basit katalog yapısı sabit uzunluklu dosya adı disk adres ve dosya özellik bilgileri (b) Her kayıt bir i-node yapısını gösterir



## Katalogların Gerçeklenmesi (2)

- dosya isimlerinin sıralı tutulmasının sorunları
  - dosya isimleri sözcük başlarında başlamalı
  - sona boşluk ekle
  - dosya silinince değişken boylu bölgeler boşalır
    - sıkıştırmak kolay
  - bir kayıt birden fazla sayfa boyunda olabilir
    - dosya adını okurken olası sayfa hatası
- · dosya isimlerinin en sonda tutulması
  - tüm dosyalar için eş boylu kayıt alanı
  - yine sayfa hatası olabilir
  - dosya isimleri sözcük başında başlamak zorunda değil

### Disk Alanı Yönetimi

- · dosyalar sabit boylu bloklara bölünür
- · blok boyu ne olmalı?
  - sektör, iz, silindir boyu?
    - aygıta bağlı
  - boy seçimi önemli
    - başarım ve etkin yer kullanımı çelişen hedefler
    - ortalama dosya boyuna göre seçmek iyi
    - çoğu sistemde çok önceden belirlenmiş
      - UNIX: çoğunlukla 1K

Boş Blokların Yönetimi 1001101101101100 0110110111110111 136 162 234 210 612 897 1010110110110110 97 342 422 0110110110111011 214 140 1110111011101111 41 63 160 223 1101101010001111 21 664 223 0000111011010111 48 216 160 1011101101101111 262 320 126 1100100011101111 142 516 141 1101111101110111 1 KB disk block can hold 256 32-bit disk block numbers (b)

97

#### UNIX V7 Dosya Sistemi (1) kök kataloğundan başlayan ağaç yapısı dosya özellikleri i-node yapısında dosya adı max. 14 dosya boyu, yaratılma, erişim ve değişim zamanı, sahibi. grubu, koruma bitleri, bağlantı sayısı karákter - / ve NUL olamaz NUL = 0 (isimleri 14 karaktere tamamlamak için) 14 katalog yapısında dosya başına bir kayıt File name dosya adı (14 karakter) i-node numarası (2 sekizli) I-node number UNIX V7 katalog kaydı

