Bölüm 10: Dosya Sistemleri

İşletim Sistemleri

Dosya Sistemleri

- Birçok uygulama, bir sürecin sanal adres alanında sahip olduğundan daha fazla bilgi depolaması gerekir.
- Bilgiler, onu kullanan sürecin sona ermesinden sonra da hayatta kalmalıdır.
- Birden çok süreç aynı anda bilgilere erişebilmelidir.

Dosya Sistemleri

- Diskler dosyaları depolamak için kullanılır
- Bilgiler disklerdeki bloklarda saklanır.
- Dosya sistemi blokları okuyabilir ve yazabilir
- Dosyalar, süreçler tarafından oluşturulan, adres uzayı türüne benzer, mantıksal bilgi birimleridir.
- Dosya sistemi dosyaları yönetir: nasıl yapılandırıldıkları, adlandırıldıkları, erişildikleri, kullanıldıkları, korundukları, uygulandıkları vb.

Dosya Sistemleri

- Bir diskte bloklar halinde tutulan bilgilere erişimle başa çıkmak için dosya sistemi bir soyutlama olarak kullanılır
- Dosyalar bir süreç tarafından oluşturulur
- Bir diskte binlerce dosya bulunabilir
- İşletim sistemi tarafından yönetilir

Dosya Sistemleri

- İşletim sistemi dosyaları yapılandırır, adlandırır, korur
- Dosya sistemine bakmanın iki yolu var
  - Kullanıcı - bir dosyayı nasıl adlandırırız, koruruz, dosyaları nasıl düzenleriz
  - Uygulama - bir diskte nasıl düzenlenirler? (organize)
- Kullanıcı bakış açısıyla
  - Adlandırma (naming)
  - Yapı (structure)
  - Dizinler (directories)

Adlandırma

- Mevcut tüm işletim sistemlerinde bir ila 8 harf
- Unix, MS-DOS (FAT16) dosya sistemleri ele alındı
- İlk Windows sistemlerde FAT16 ve FAT32 kullanılmıştır.
- Son Windows sistemler yerel (native) dosya sistemi kullanır
- Tüm işletim sistemleri adın bir parçası olarak sonek (suffix) kullanır
- Unix sonekler ’in bir anlam ifade etmesini zorlamazken, DOS sistemde soneklerin bir anlamı vardır
- Sonek uzantı olarak da kullanılmaktadır.

Sonek Örnekleri

- Belgeler: .doc, .docx, .pdf, .txt, .rtf, .odt
- Görseller: .jpg, .jpeg, .png, .gif, .bmp, .tiff
- Ses: .mp3, .wav, .aiff, .m4a, .wma
- Video: .mp4, .avi, .mov, .wmv, .flv
- E-tablolar: .xls, .xlsx, .csv
- Sunumlar: .ppt, .pptx, .odp
- Web: .html, .css, .js
- Yürütülebilir: .exe, .msi
- Sıkıştırılmış: .zip, .rar, .tar, .gz, .7z

Uzantılar ve Açılımları

doc	Microsoft Word Document
docx	Microsoft Word Open XML Document
pdf	Portable Document Format
txt	Plain Text File
rtf	Rich Text Format
odt	Open Document Text
jpg	Joint Photographic Experts Group Image
jpeg	Joint Photographic Experts Group Image
png	Portable Network Graphics
gif	Graphics Interchange Format

Uzantılar ve Açılımları

bmp	Bitmap Image
tiff	Tagged Image File Format
mp3	MPEG-1 Audio Layer 3
wav	Waveform Audio Format
aiff	Audio Interchange File Format
m4a	MPEG-4 Audio
wma	Windows Media Audio
mp4	MPEG-4 Part 14
avi	Audio Video Interleave
mov	Apple QuickTime Movie

Uzantılar ve Açılımları

wmv	Windows Media Video
flv	Flash Video
xls	Microsoft Excel Spreadsheet
xlsx	Microsoft Excel Open XML Spreadsheet
csv	Comma Separated Values
ppt	Microsoft PowerPoint Presentation
pptx	Microsoft PowerPoint Open XML Presentation
odp	Open Document Presentation
html	HyperText Markup Language
css	Cascading Style Sheets

Uzantılar ve Açılımları

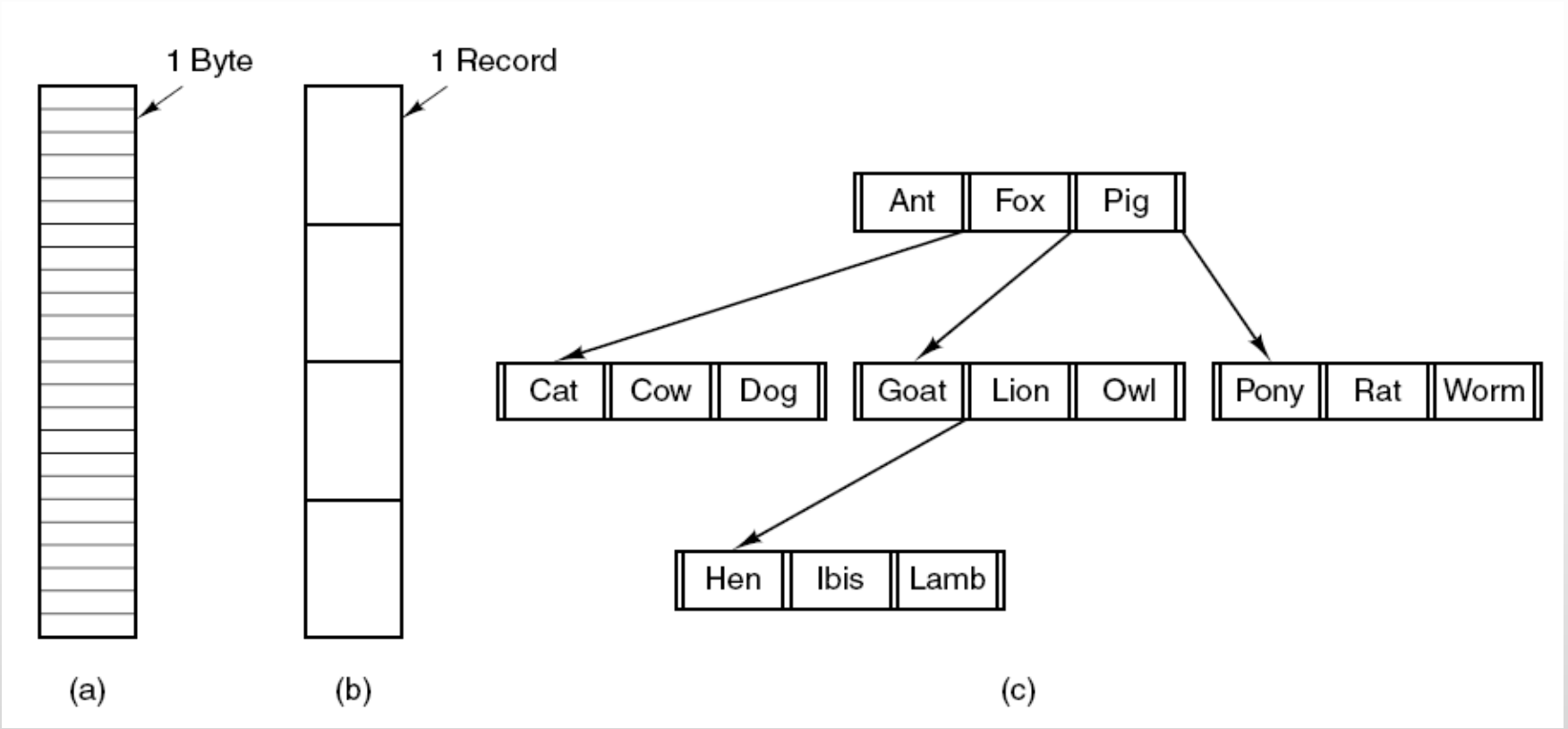
js	JavaScript
exe	Executable File
msi	Windows Installer Package
zip	Zipped File
rar	RAR Archive
tar	Tape Archive
gz	GZIP Compressed Archive
7z	7-Zip Compressed Archive

Dosya Yapısı

- Bayt dizilerinden oluşur
- Maksimum esneklik – içine her şey konabilir
- Unix ve Windows bu yaklaşımı kullanır
- Sabit uzunluklu kayıtlar (eskiden kart imajları)
- Kayıt ağacı - ağaçtaki kayıtları bulmak için anahtar alanı (key field) kullanır

Dosya Yapısı

(a) Bayt dizisi. (b) Kayıt dizisi. (c) Ağaç



Dosya Tipleri

- Normal – Kullanıcı bilgilerini içerir
- Dizinler – Dosyaların izini tutan dosyalardır
- Karakter özel dosyaları – seri (serial) model G/Ç cihazları (yazıcı)
- Blok özel dosyaları – blok tabanlı modeller (disk)

Normal (regular) Dosyalar

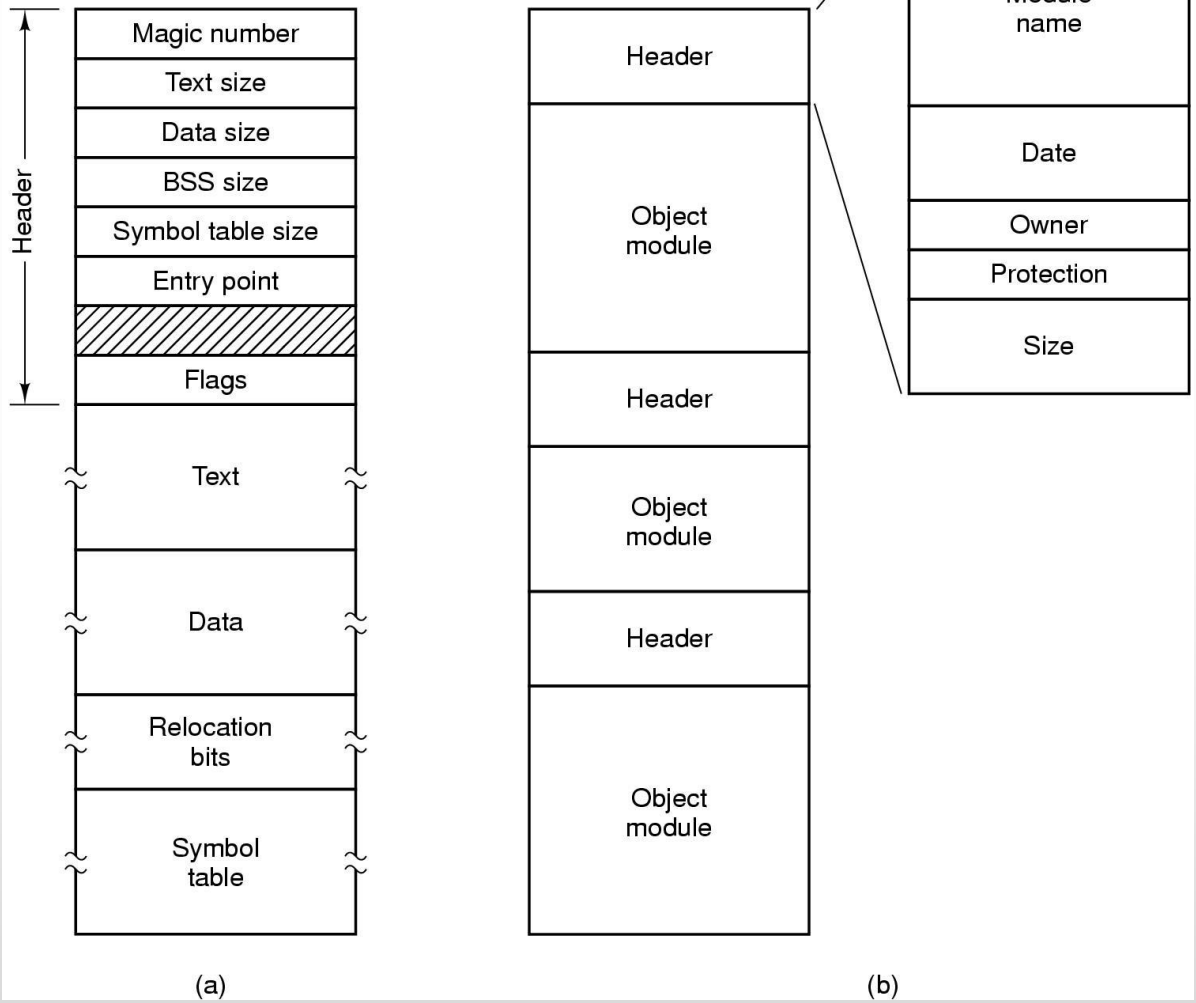
- ASCII veya ikili (binary)
- ASCII
  - Yazdırılabilir
  - Programları bağlamak için boru hattı (pipe) kullanılabilir (ASCII üretiyor/tüketiyorsa)

İkili Dosya Tipleri

- İki Unix örneği
  - Yürütülebilir (magical field, dosyayı yürütülebilir olarak tanımlar)
  - Arşiv olarak derlenmiş, bağlı (linked) kütüphane prosedürleri hariç
- Her işletim sistemi kendi yürütülebilir dosyasını tanımalıdır

İkili Dosya Tipleri

- (a) Yürütülebilir dosya (.exe)
- (b) Derlenmiş ancak bağlanmamış arşiv kütüphanesi (.o)



Dosya Erişimi

- Sıralı erişim – okumaya baştan başlanır, atlama yapılmaz
  - Manyetik banda karşılık gelir
- Rastgele erişim – okumak istenen yerden başlanır
  - Disklerle beraber devreye girdi
  - Birçok uygulama için gereklidir, (havayolu rezervasyon sistemi)
- Dosya tanımlayıcı
  - Bir dosya tanımlayıcı, bir işlemin okuyabileceği veya yazabileceği, çekirdek tarafından yönetilen bir nesneyi temsil eden küçük bir tamsayıdır.
  - Her işlemin, 0'dan başlayan özel bir dosya tanıtıcı alanı vardır.
  - Geleneksel olarak, 0 standart girdi, 1 standart çıktı ve 2 standart hatadır

Dosya Öznitelikleri

Read-only	Dosya yalnızca okunabilir, değiştirilemez veya silinemez.
Hidden	Dosya, dosya gezgininin normal görünümünde görünmez.
System	Dosya işletim sistemi tarafından kullanılır ve değiştirilmemeli silinmemelidir.
Archive	Dosya, yedeklendikten sonra değiştirilmiş, sonraki yedeklemeye dahil edilmeli.
Compressed	Dosya, disk alanından tasarruf etmek için sıkıştırılmıştır.
Encrypted	Yetkisiz erişimi önlemek için dosya şifrelenmiştir.
Temporary	Dosya geçici olarak kullanılır ve ihtiyaç kalmadığında otomatik olarak silinir.
Executable	Dosya çalıştırılabilen bir programdır.
Indexed	Dosya, daha hızlı arama için indeksleme hizmetine dahil edilmiştir.
Offline	Dosya anlık kullanım için değildir, çevrimdışı kullanım için hazır hale getirilebilir.
Not content indexed	Dosya, aramayı hızlandırmak için indeksleme hizmetinden çıkarılır.
Reparse point	Dosya, başka bir dosyaya veya dizine bir bağlantı veya referans içerir.

Dosya Öznitelikleri

Sparse file	Dosya, diskte depolanmayan büyük sıfır blokları içeren bir dosya türüdür.
Symlink	Dosya, başka bir dosya veya dizine sembolik bir bağlantıdır.
Device	Dosya, işletim sistemi tarafından kullanılan özel bir aygıt dosyasıdır.
Junction point	Dosya, başka bir birimdeki bir dizine işaret eden bir tür sembolik bağlantıdır.
Creation time	Dosyanın oluşturulduğu tarih ve saat
Last Access	Dosyaya en son erişildiği tarih ve saat
Last change	Dosyanın en son değiştirildiği tarih ve saat
Current size	Dosyadaki bayt sayısı
Max size	Dosyanın büyüyebileceği bayt sayısı
Owner	Dosyayı sahibinin kimliği
Creator	Dosyayı oluşturan kişinin kimliği

Stat Veri Yapısı

```
struct stat {
      mode_t      st_mode;    // file type and mode (permission)
      ino_t       st_ino;     // inode number
      dev_t       st_dev;     // device number
      dev_t       st_rdev;    // device number (special)
      nlink_t     st_nlink;   // number of links
      uid_t       st_uid;     // user ID of owner
      gid_t       st_gid;     // group ID of owner
      off_t       st_size;    // size in bytes
      time_t      st_atime;   // time of last access
}
```

Dosya Öznitelikleri

- drwxr-xr-x 2 root root 4096 Sep 24 2008 Unit2
- drwxr-xr-x 2 root root 4096 May 26 19:21 a
- -rwxr-xr-x 1 root root 10930 Aug 5 22:49 a.out
- -rwxrwx--T 1 root root 81 Aug 2 2008 a.txt
- -rwxr-x--- 1 root root 81 May 26 19:20 b.txt
- -rwx------ 1 root root 81 Jul 30 19:28 c.txt
- -rwxr-xr-x 1 root root 11193 Jul 30 19:27 cp

Dosya Öznitelikleri

- - rwx rw- r--

- Dosya türü: "-" bir dosya anlamına gelir. "d" bir dizin anlamına gelir.
- Dosyanın sahibi için okuma, yazma ve yürütme izinleri (rwx)
- Dosyanın sahibi olan grubun üyeleri için okuma, yazma ve yürütme izinleri (rw-)
- Diğer tüm kullanıcılar için okuma, yazma ve yürütme izinleri (r--)

Dosyalar için Sistem Çağrıları

- Oluştur - veri olmadan, bazı öznitelikleri ayarlar (create)
- Sil - Disk alanını boşaltmak için (delete)
- Aç - Oluşturduktan sonra, öznitelikleri ve disk adreslerini ana belleğe alır (open)
- Kapat - Öznitelikler ve adresler tarafından kullanılan tablo alanını boşaltır (close)
- Okuma – İşaretçinin geçerli konumundan okuma işlemi. Verilerin yerleştirileceği arabelleği belirtmek gerekir (read)
- Yazma - genellikle işaretçinin geçerli konuma yazma işlemi (write)

Dosyalar için Sistem Çağrıları

- Ekle - dosyanın sonuna ekleme işlemi (append)
- Ara - dosya işaretçisini dosyada belirli bir yere koyar. (seek) Bu konumdan okuma veya yazma yapılır.
- Öznitelikleri Al – örneğin, derleme yapılacağında dosyaların en son değişiklik zamanlarını öğrenmek için.
- Öznitelikleri Ayarla – örneğin, erişim koruma (r,w,x) ayarlama
- Yeniden adlandırmak (rename)

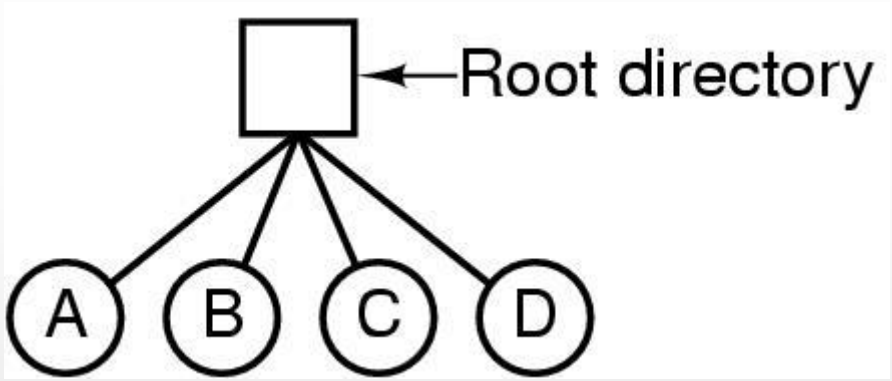
Dosya Kopyalama Örneği – copy abc xyz

- abc dosyasını xyz'ye kopyalar
- Eğer xyz varsa üzerine yazılır
- Yok ise yaratılır
- Sistem çağrıları kullanılır (okuma, yazma)
- 4K boyutunda parçalar halinde okur ve yazar
- abc dosyasından bir tampon belleğe oku (read sistem çağrısı)
- Tampondan xyz dosyasına yaz (write sistem çağrısı)

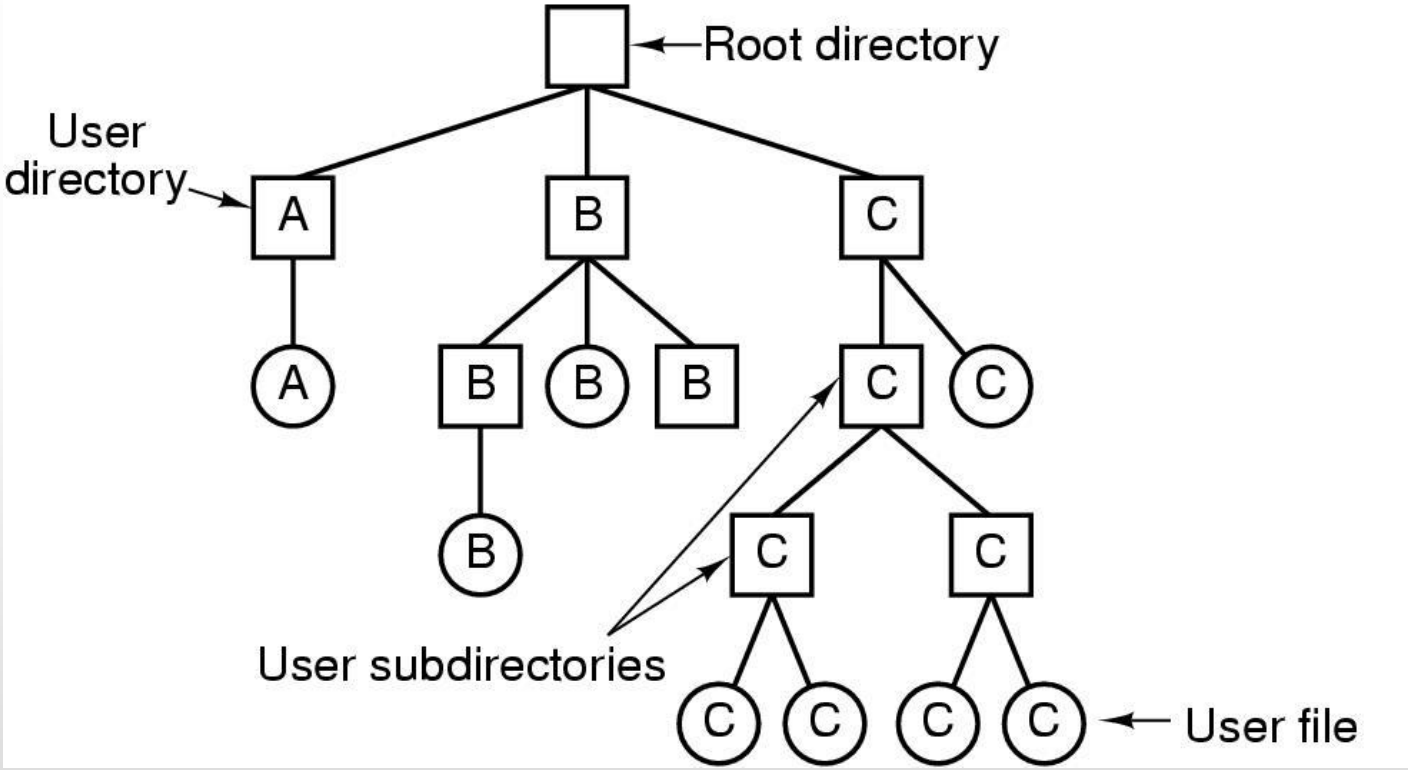
Dizinler

- Bir dosya koleksiyonunu düzenlemek için kullanılan dosyalar
- Bazı işletim sistemlerinde klasörler (folder) olarak da adlandırılır
- Katı bağlama (hard link)
  - Bağlama, bir dosyanın birden fazla dizinde görünmesini sağlar; dosyanın i-düğümündeki sayacı artırır
- Sembolik bağlama (symbolic link)
  - Başka bir dosyayı adlandıran küçük bir dosyaya işaret (point) eden bir ad oluşturulur.

Dört Dosya İçeren Tek Düzeyli Dizin



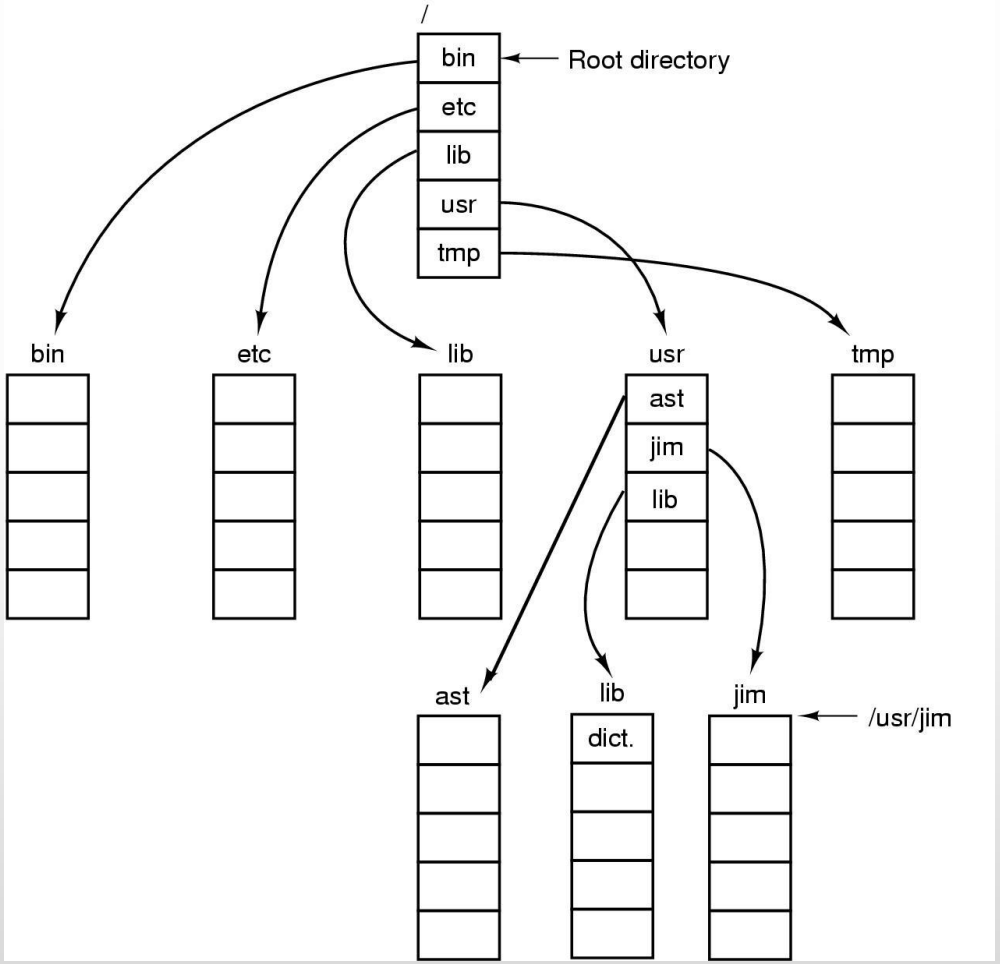
Hiyerarşik Dizin Sistemleri



Yol (path) Adları

- Mutlak /usr/sercan/os/slaytlar
- Bağıl os/slaytlar
- . Geçerli (çalışan) dizini ifade eder
- .. Geçerli dizinin ebeveynini (bir üst klasör) ifade eder

UNIX Dizin Ağacı



Dizin İşlemleri

- Create, dizin oluşturur
- Delete, dizini siler, silmek için dizin boş olmalıdır
- Opendir, dizinde bir işlem yapılmadan önce yapılmalıdır.
- Closedir, tüm işlemlerden sonra yapılır
- Readdir, açılmış dizindeki bir sonraki girişi (elemanı) döndürür
- Rename, Yeniden adlandırır
- Link, Dosyayı başka bir dizine bağlar
- Unlink, Bağlantıyı Kaldırır, Dizin girişinden kurtulur

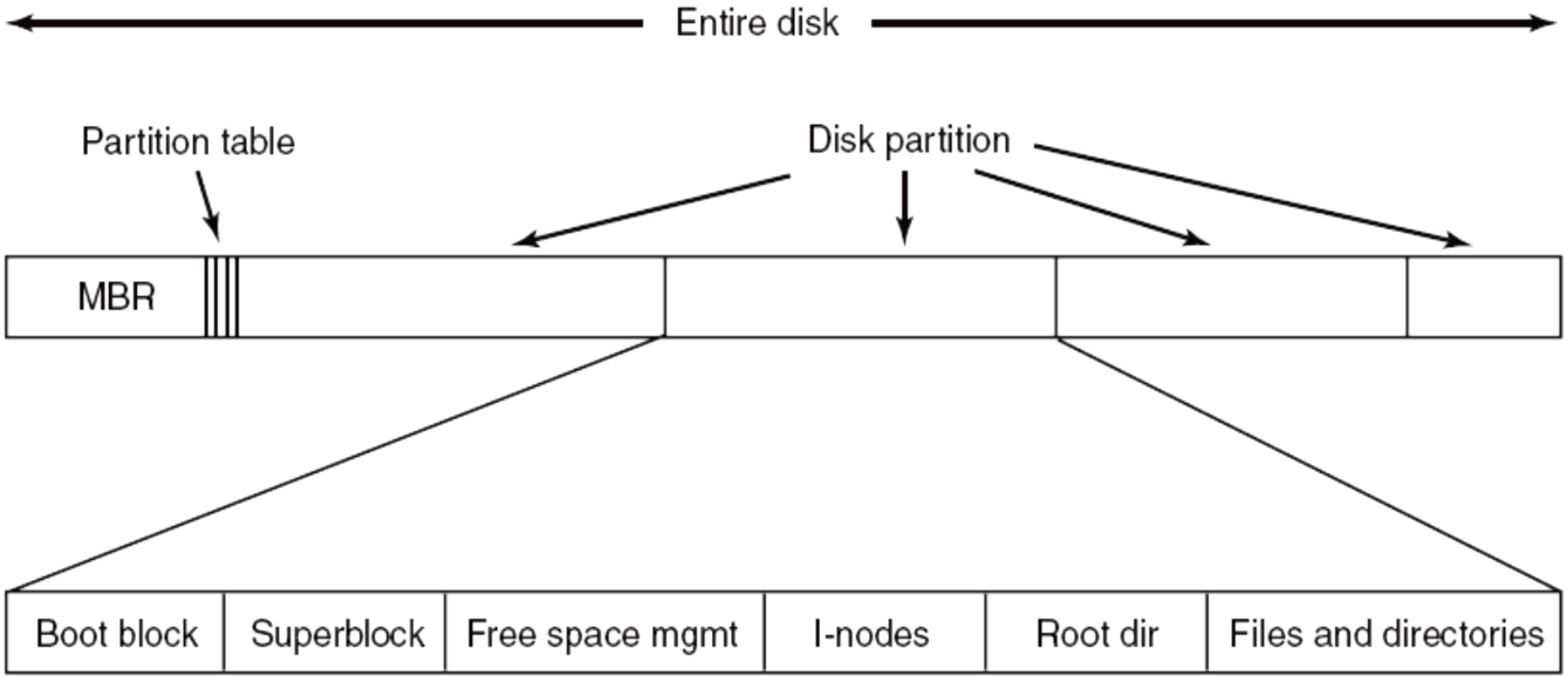
Dosya Sistemi Gerçekleme (implementation)

- Dosyalar disklerde saklanır.
- Diskler bir veya daha fazla bölümden (partition) oluşabilir.
- Her bölümde ayrı «dosya sistemi» olabilir
- Diskin 0. sektörü, ana önyükleme kaydıdır (master boot record)
- Bilgisayarın açılışı (boot) için kullanılır
- MBR'nin sonu bölüm tablosuna sahiptir.
- Tabloda her bölümün başlangıç ve bitiş adresleri bulunur.
- Bölümlerden biri, etkin (active) olarak işaretlenir

Dosya Sistemi Gerçekleme (implementation)

- Bilgisayarın açılışı => BIOS, MBR'yi okur/yürütür
- MBR aktif bölümü bulur ve ilk bloğu okur (önyükleme bloğu)
- Önyükleme bloğundaki program, o bölüm için işletim sistemini bulur ve okur.
- Tüm bölümler bir önyükleme bloğuyla başlar

Dosya Sistemi Düzeni (layout)



Dosya Sistemi Düzeni (layout)

- **Boot block**: Önyükleme yükleyici kodunu içeren diskin ilk bloğu. Çekirdeği belleğe yükleyerek işletim sistemini başlatmak için.
- **Superblock**: Dosya sistemi hakkında bilgi içeren veri yapısı. Dosya sisteminin boyutu, boş alan yönetimi ve blok boyutu gibi bilgileri depolar.
- **Free Space Management**: Boş blokları takip etme ve bunları yeni dosya ve dizinlere tahsis etme mekanizması. Dosya sistemini düzenli tutar, parçalanmayı azaltır ve disk alanı kullanımını en üst düzeye çıkarmak.

Dosya Sistemi Düzeni (layout)

- **i-nodes**: Bir dosya veya dizin hakkında bilgi içeren bir veri yapısı. Boyut, izinler, sahiplik, zaman damgaları ve dosya verilerinin konumu gibi meta verileri depolar.
- **Root directory**: Dosya sistemi hiyerarşisinde en üstteki dizin. Dosya sistemindeki dosyalara ve dizinlere erişim için başlangıç noktasıdır.
- **Files**: Kalıcı olarak saklanması gereken bir programın, belgenin veya diğer verilerin içeriğini saklar.
- **Directory**: Daha kolay gezinme ve yönetim için dosyaları yapılandırır ve kategorilere ayırır.

Dosya Sistemi Düzeni (layout)

- Dosya sistemi, bir diskteki verilerin düzenlenmesi ve işletim sistemi tarafından yönetilmesidir.
- Önyükleme bloğu, süper blok, i-düğümler ve dizinler gibi dosya sisteminin her bileşeni, diskte depolanan verilerin yönetilmesinde ve bunlara erişilmesinde belirli bir rol oynar.
- Dosya sistemi seçimi ve tasarımı, işletim sisteminin performansını, güvenilirliğini ve verimliliğini etkiler.
- Dosya sistemi, işletim sisteminin kritik bir bileşenidir ve tasarımı, diskin boyutu, depolanan veri türü ve verilere erişen kullanıcı sayısı gibi sistemin ihtiyaçlarını dikkate almalıdır.

Blokların Dosyalara Tahsisi

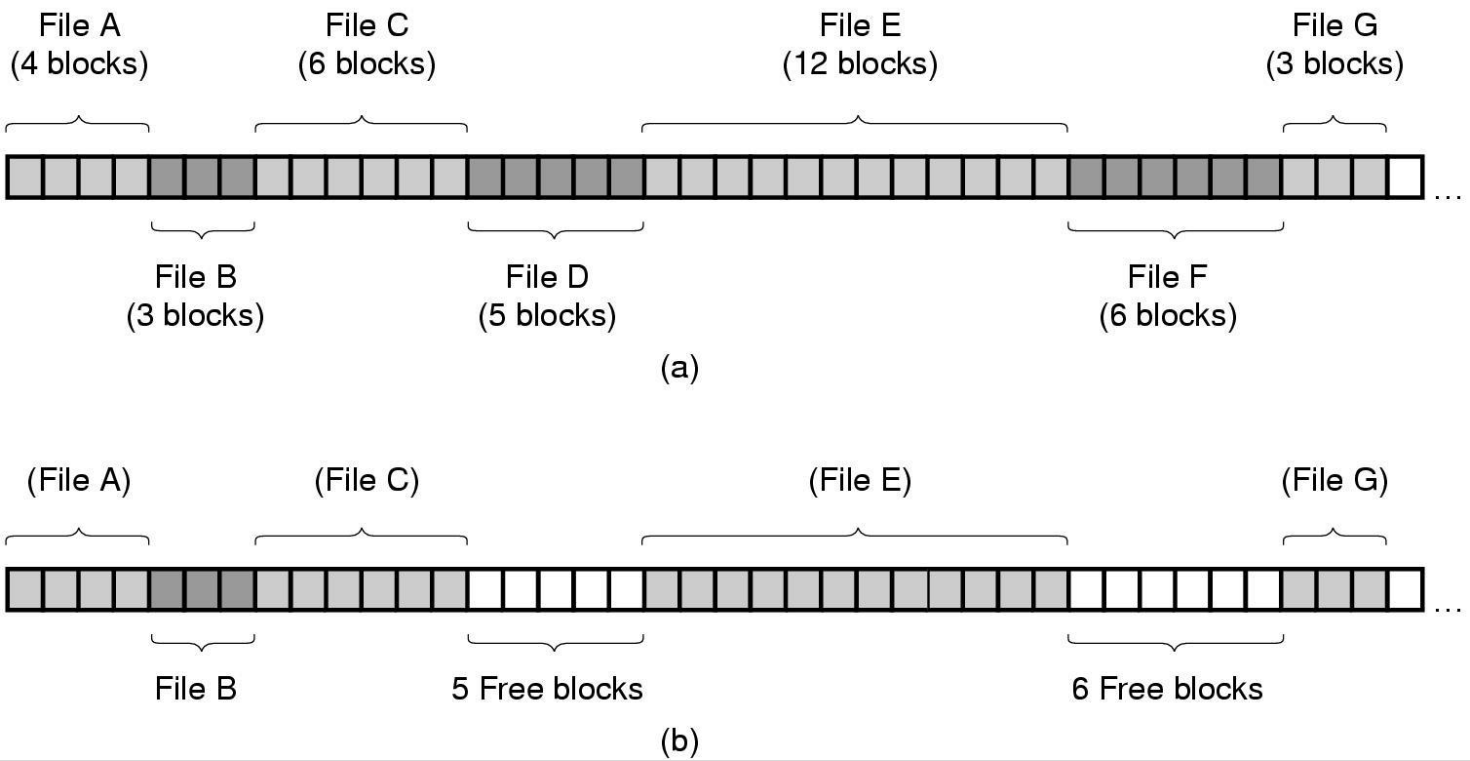
- En önemli uygulama sorunu
- Yöntemler
  - Bitişik yer tahsisi (contiguous)
  - Bağlı liste tahsisi (linked list)
  - Tablo (table) kullanılarak bağlı liste tahsisi
  - I-nodes

Bitişik Yer Tahsisi

- Bir dosyaya bitişik disk sektörleri bloğu atayarak disk alanı ayırma yöntemi.
- Basit ve verimli bir yöntem, okuma performansı harika.
- Uygulaması kolay, verilere hızlı erişim.
- Dahili parçalanma nedeniyle disk alanını boşa harcar, daha büyük dosyalara yer tahsisi zordur ve disk kullanımını sınırlar.
- Dosyadaki ilk bloğu bulmak için yalnızca bir arama (seek) yeterli
- Disk zamanla parçalanır (fragmented)
- CD-ROM'lar, dosya sistemi boyutu sabit olduğundan bitişik yer tahsisi

Bitişik Yer Tahsisi

(a) 7 dosya için bitişik disk alanı tahsisi. (b) D ve F dosyaları kaldırıldıktan sonra diskin durumu.

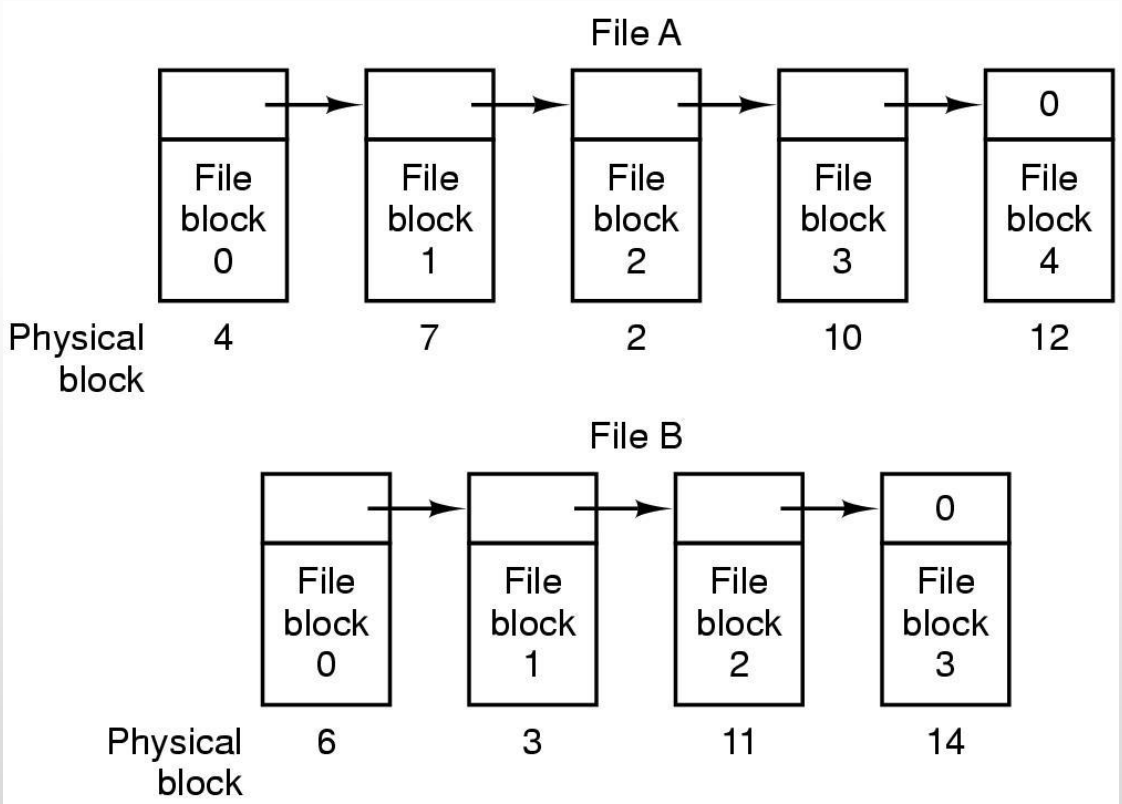


Bağlı Liste Yer Tahsisi

- İşaretçiler kullanarak disk sektörü bloklarını birbirine bağlayarak disk alanı ayırma yöntemi.
- Disk alanının dinamik tahsisine izin verir, bitişik tahsisin sınırlamalarını kaldırır. Rastgele erişim yavaştır.
- Dahili parçalanmayı en aza indirir, disk alanının dinamik olarak tahsis edilmesini sağlar ve disk kullanımını geliştirir.
- Verilere daha yavaş erişim, uygulanması daha karmaşık ve işaretçileri depolamak için ek bellek gerektirir.

Bağlı Liste Yer Tahsisi

- Bir dosyayı, disk bloklarından bağlı liste olarak saklamak.



Tablo Kullanılarak Bağlı Liste Yer Tahsisi

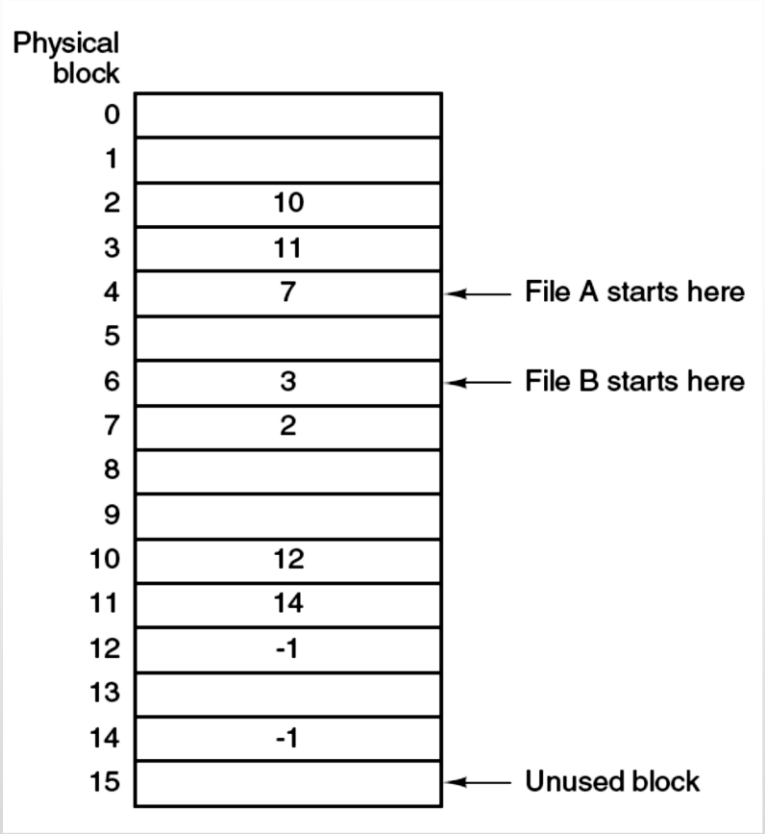
- Disk sektörü bloklarına işaretçileri depolamak için bir tablo kullanarak disk alanı ayırma yöntemi.
- Verilere erişim hızını artırmak ve bağlantılı liste tahsisinin bellek gereksinimlerini azaltmak.
- Verilere daha hızlı erişim, daha düşük bellek gereksinimleri ve gelişmiş disk kullanımı.
- Uygulaması daha karmaşıktır ve dahili parçalanmaya neden olabilir.
- File Allocation Table (FAT)

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

- Tüm tabloyu bellekte tutar!
- Ölçeklenebilir değil!
- Tablonun boyutu gerçekten büyük oluyor
- Örneğin, 1 KB bloklu 200 GB disk, 600 MB'lık bir tabloya ihtiyaç duyar
- Tablo boyutunun büyümesi, disk boyutunun büyümesiyle doğru orantılıdır

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

- Ana bellekte bir dosya tahsis tablosu kullanarak bağlantılı liste yer tahsisi.



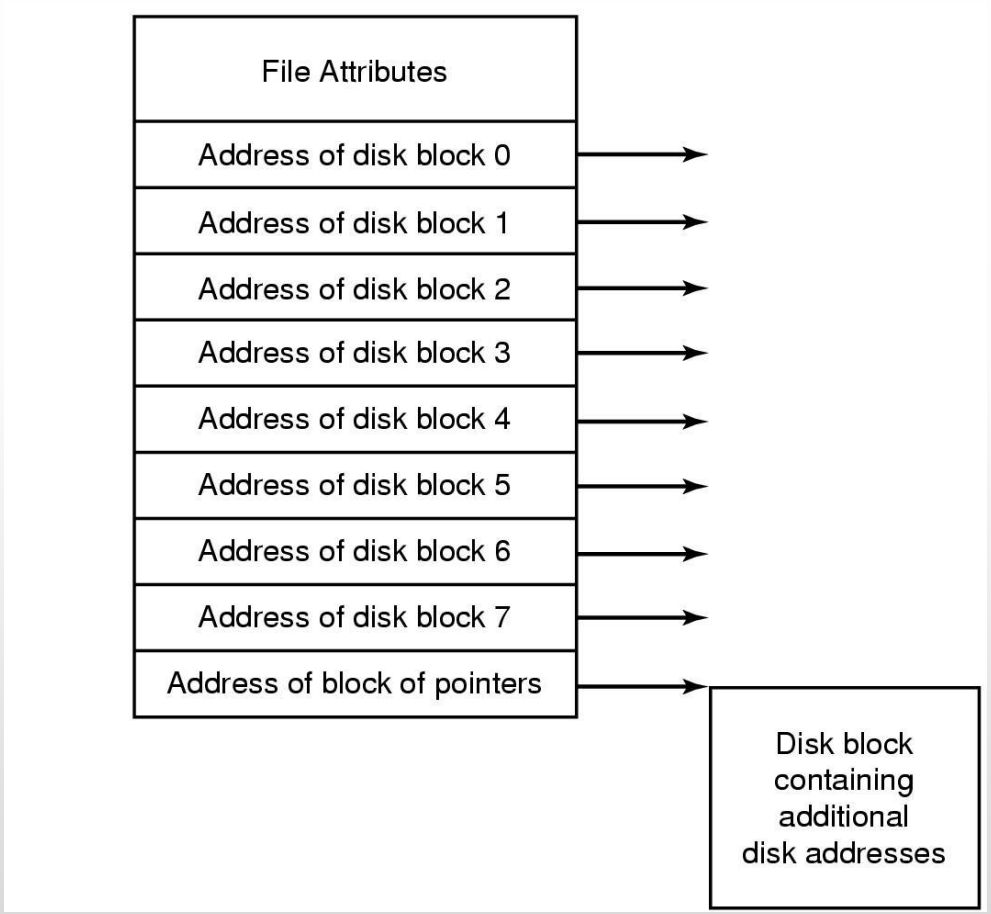
I-nodes

- Dosya sistemindeki dosyaları ve dizinleri temsil etmek için bir i-düğüm tablosu kullanarak disk alanı ayırma yöntemi.
- Disk alanı tahsis etmek ve her dosya ve dizin için meta verileri yönetmek için esnek ve verimli bir yöntem sağlar.
- Dahili parçalanmayı en aza indirir, disk alanının esnek bir şekilde tahsis edilmesini sağlar ve meta verilerin etkin yönetimini sağlar.
- Uygulanması daha karmaşıktır, i-düğümleri depolamak için ek bellek gerektirir ve disk kafasının daha fazla hareket etmesine neden olabilir.

I-nodes

- Veri yapısını yalnızca açık dosyalar için bellekte tutar
- Veri yapısı, blokların adreslerini ve dosyaların özniteliklerini listeler
- *K* aktif dosya, dosya başına *N* blok => en fazla *K***N* blok
- Dosya tablosu disk boyu ile orantılıdır, Büyüme sorununu çözer
- N ne kadar büyük olabilir?
- Tablodaki son giriş, diğer disk bloklarına işaretçiler içeren disk bloğuna işaret eder.
- i-node büyüklüğü sabittir!

Örnek I-node

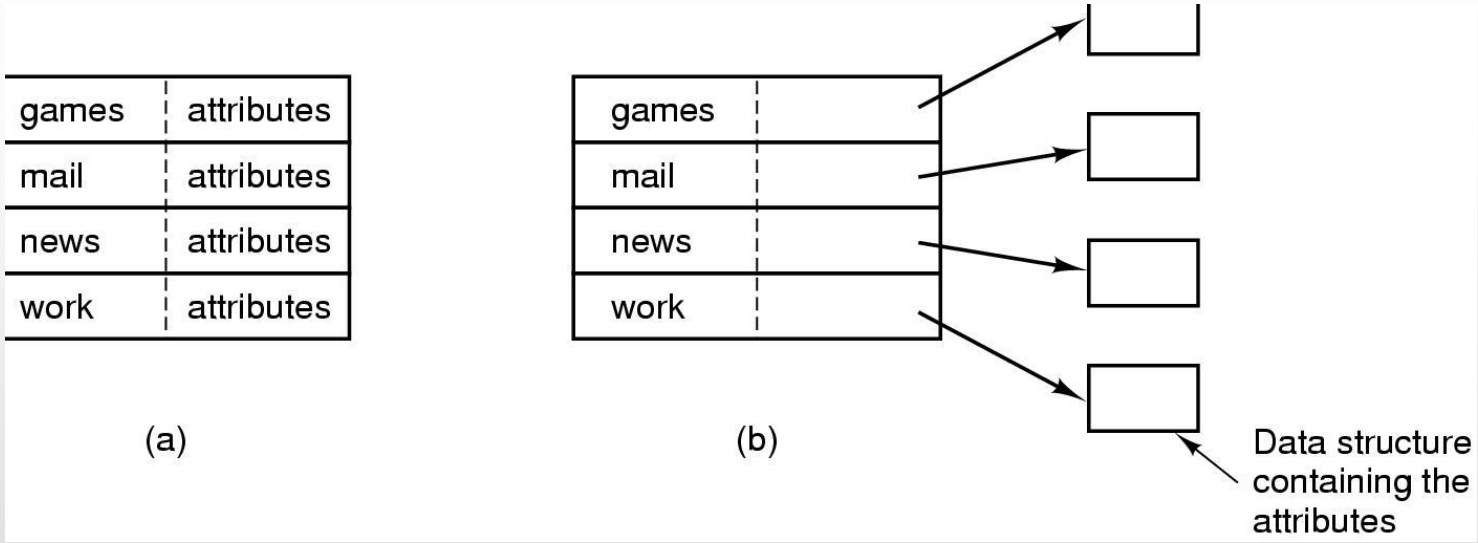


Dizinler

- Open file, dizini bulmak için kullanılan yol adı (path)
- Dizin, aşağıdakileri bilgileri kullanarak blok adreslerini belirtir
  - İlk bloğun adresi (bitişik yer)
  - İlk bloğun sayısı (bağlı liste)
  - i-node sayısı

Dizinler

(a) disk adresleri ve nitelikleri ile sabit boyutlu girişler (DOS) (b) her giriş bir i-node ifade eder. Dizin girişi öznitelikleri içerir. (Unix)

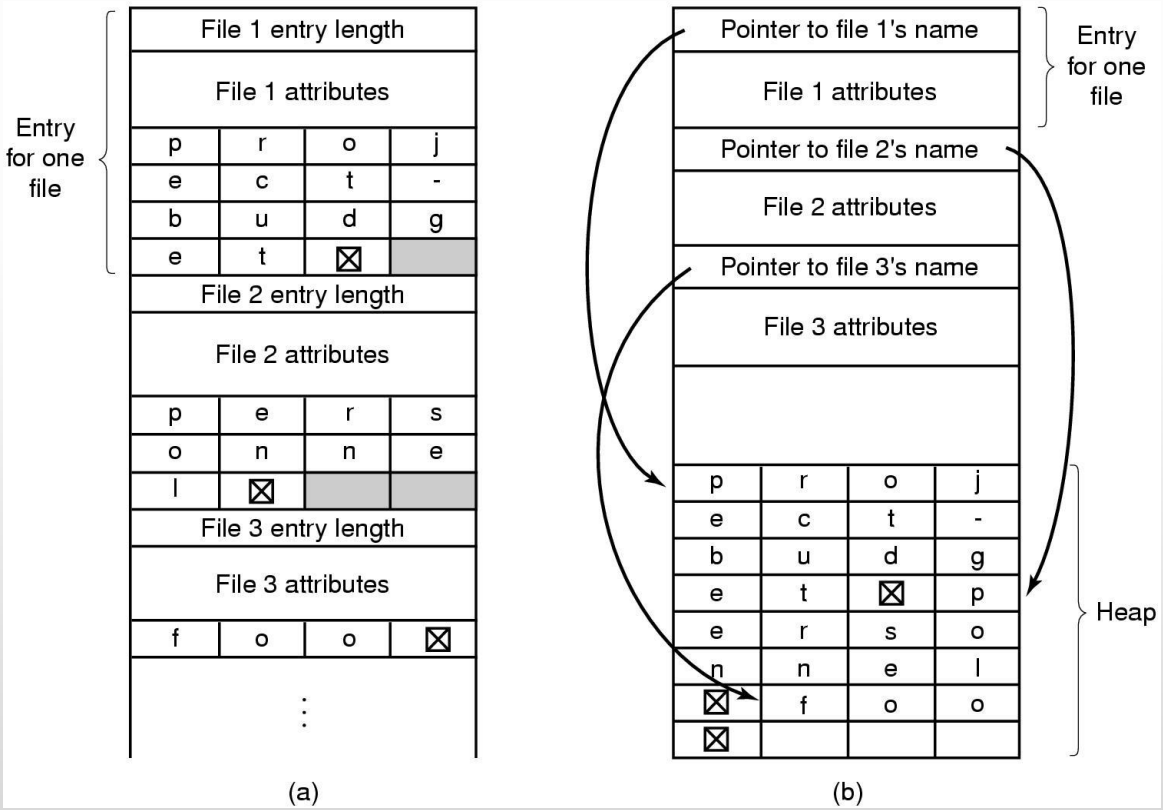


Dizinler

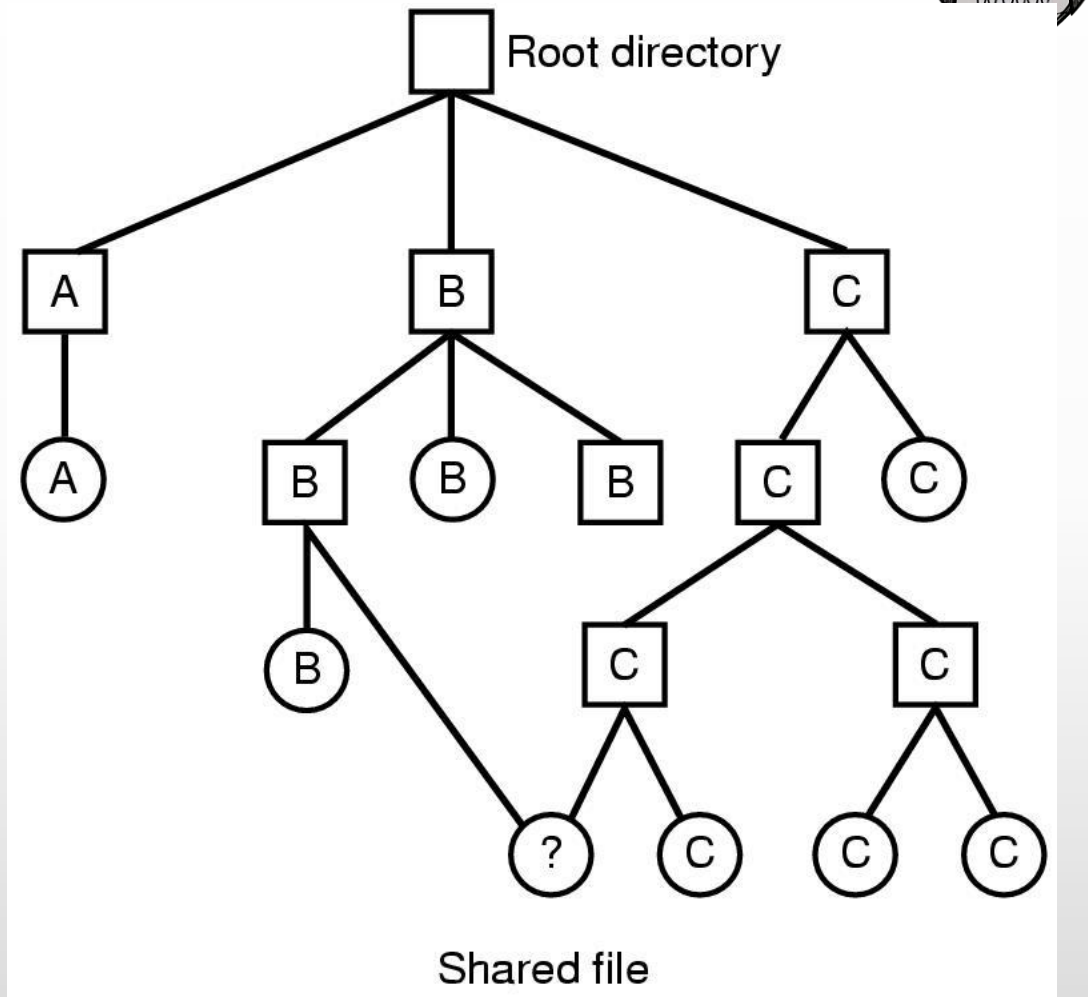
- Değişken uzunluklu adlarla nasıl başa çıkarız?
- Çok uzun adlar problem
- İki yaklaşım
  - Sabit başlık ve ardından değişken uzunluklu adlar
  - Yığın işaretçisi adları işaret eder

Dizinler

- Uzun dosya adlarını işleme. (a) Sıralı. (b) yığın içinde.



Paylaşımlı Dosyalar



- Paylaşılan bir dosya içeren dosya sistemi. Dosya sistemleri Bir yönlendirilmiş döngüsüz ağaçtır (DAG)

Paylaşımlı Dosyalar

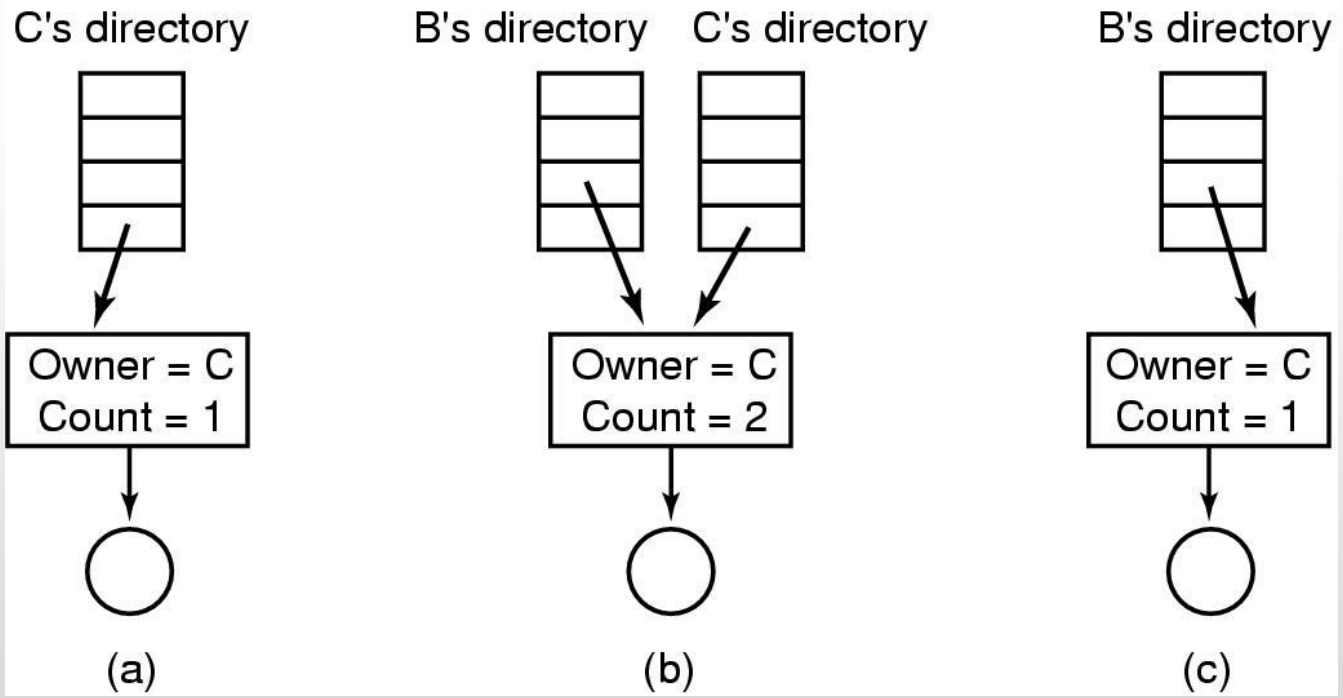
- B veya C yeni bloklar eklerse, diğer sahip nasıl öğrenir?
- Paylaşılan dosyalar için özel i-node kullan - dosyanın paylaşıldığını gösterir
- Sembolik bağlantı (symbolic link) kullanın - sahibi C ise, B'nin dizinine konulan özel bir dosya. Bağlı (linked) olduğu dosyanın yol adını içerir

I-node Problem

- C dosyayı kaldırırsa, B'nin dizini paylaşılan dosya için hala i-node'u işaret eder.
- i-node başka bir dosya için yeniden kullanılırsa, B'nin girişi noktası yanlış i-node'u gösterir.
- Çözüm, i-node'dan çıkmak ve sahip sayısını azaltmaktır.

I-node Problem

(a) Bağlamadan önceki durum. (b) Bağlantı oluşturulduktan sonra. (c) Orijinal sahibi dosyayı kaldırdıktan sonra.



Sembolik Bağlantı

- Sembolik bağlantı sorunu çözer
- Çok fazla sembolik bağlantıya sahip olabilir ve bunların takip edilmesi zaman alır.
- Büyük avantaj - diğer makinelerdeki dosyalara işaret edebilir

Günlük (log) Yapılandırılmış Dosya Sistemi

- Disk bloklarına bir günlük veya güncelleme dizisi olarak yazarak diskteki verileri düzenleme yöntemi.
- Rastgele disk erişiminden kaçınarak, büyük ölçekli depolama sistemleri için yüksek performanslı bir dosya sistemi sağlar.
- Geliştirilmiş yazma performansı, basitleştirilmiş disk alanı yönetimi ve verimli disk alanı kullanımı.
- Günlük kaydı işlemi, karmaşık tasarım ve günlük düzgün bir şekilde korunmadığı takdirde potansiyel veri kaybı nedeniyle daha yüksek ek yük.

Günlük (log) Yapılandırılmış Dosya Sistemi

- İşlemciler daha hızlı, diskler ve bellekler daha büyük ancak disk arama süresi kısa değil
- Diskteki verilerin güncellenmesi gerektiğinden yazma işlemleri optimize edilmeli
- Daha büyük önbellekler kullanılarak, önbellekten okuma yapılabilir
- i-node haritası diskte tutulur ve i-node'ları bulmak için bellekte önbelleğe alınabilir
- Bu yöntemde, disk log-collect olarak yapılandırılır ve loglar periyodik olarak diskteki bir segmente gönderir.
- Segment, içerik özetine sahiptir (i-nodes, dizinler….).

Günlük (log) Yapılandırılmış Dosya Sistemi

- Temizleyici iş parçacığı günlük dosyasını sıkıştırır.
- Segmenti mevcut i-düğümler için tarar, kullanılmayanları atar ve mevcut olanları belleğe gönderir.
- Yazıcı iş parçacığı, mevcut olanları yeni segmente yazar.
- Çoğu dosya sistemiyle uyumlu değil
- Kullanılmıyor

Günlük (journaling) Dosya Sistemleri

- Diske kaydedilmeden önce dosya sistemindeki değişikliklerin günlüğünü veya günlüğünü tutarak diskteki verileri düzenleme yöntemi.
- Çökme veya sistem arızası durumunda dosya sisteminin kurtarılmasına izin vererek sağlam ve güvenilir bir dosya sistemi sağlar.
- Geliştirilmiş güvenilirlik, sistem arızalarından sonra daha hızlı kurtarma ve azaltılmış veri kaybı.
- Günlük tutma işlemi nedeniyle artan ek yük ve ek disk G/Ç nedeniyle potansiyel performans düşüşü.

Günlük (journaling) Dosya Sistemleri

- Eylemleri gerçekleştirmeden önce bir günlük tut, günlüğü diske yaz
- NTFS (Windows) ve Linux günlük kaydı kullanır
- Bir dosyanın kaldırılması gerektiğinde neler olur
  - Dosyayı bulunduğu dizinden kaldır
  - i-node'u serbest i-node havuzuna bırak
  - Tüm disk bloklarını boş disk blokları havuzuna döndür
  - Bu süreçte bir yerde bir çökme olursa ortalık karışır

Günlük (journaling) Dosya Sistemleri

- Eylemler eşgüçlü (idempotent) olmalı. Bunu yapmak için veri yapıları düzenlenmeli
- **İdempotence**: kaç kez gerçekleştirirseniz gerçekleştirin, aynı sonucu elde edersiniz.
- Blok n'yi serbest olarak işaretle, idempotent bir işlemdir.
- Bir listenin sonuna serbest bırakılmış bloklar eklemek idempotent değildir

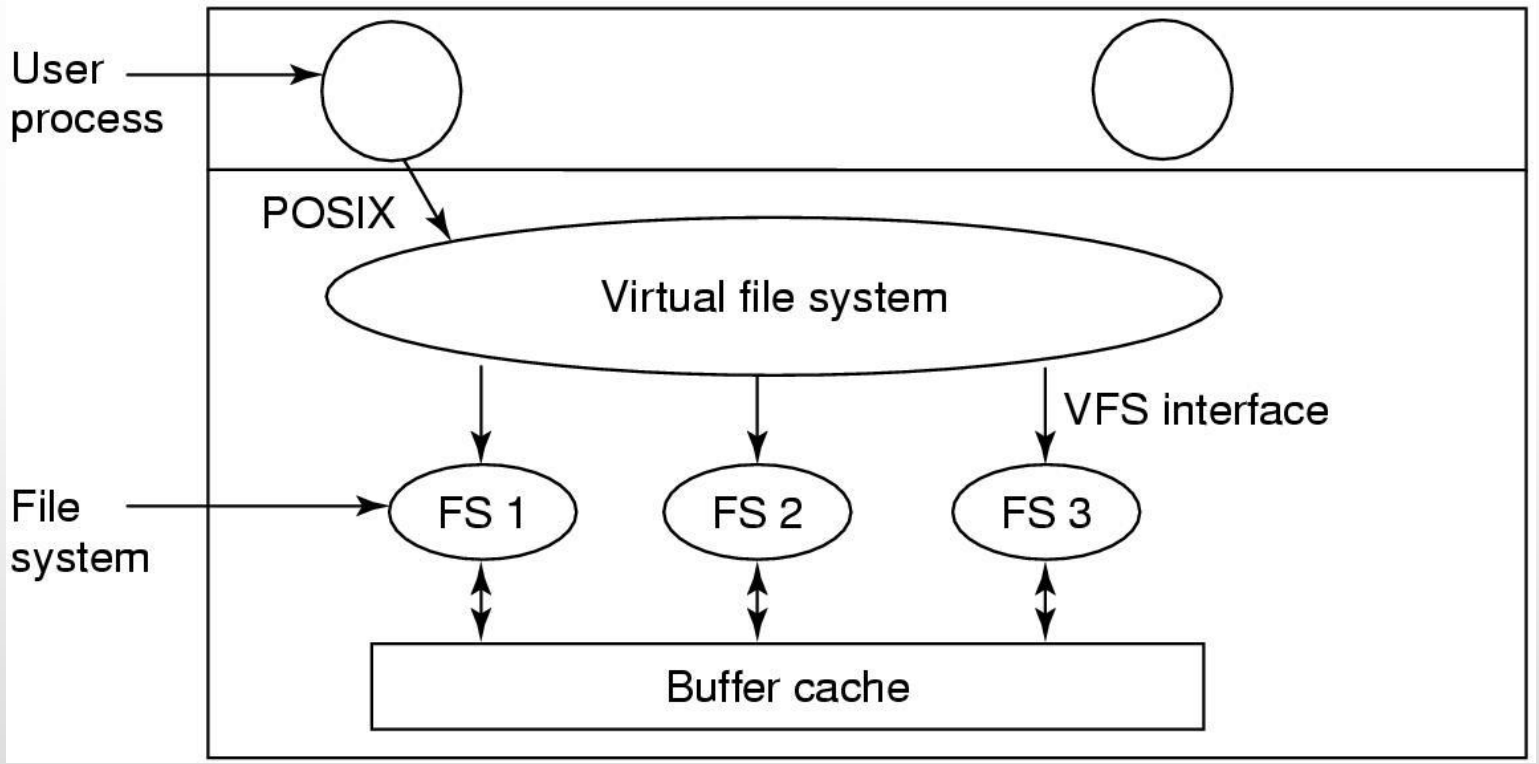
Sanal Dosya Sistemleri

- Altta yatan disk düzeninden bağımsız olarak farklı dosya sistemlerine erişim için ortak bir arabirim sağlayarak diskteki verileri düzenleme yöntemi.
- Birden çok dosya sisteminin aynı fiziksel disk üzerinde bir arada bulunmasına izin vererek ve bunlara erişim için ortak bir arabirim sağlayarak esnek ve ölçeklenebilir bir dosya sistemi sağlar.
- Geliştirilmiş esneklik, ölçeklenebilirlik ve uyumluluğun yanı sıra azaltılmış disk G/Ç.
- VFS'nin farklı disk düzenlerini ve dosya sistemi türlerini işleyebilmesi gerektiğinden artan karmaşıklık ve VFS ek bir soyutlama katmanı eklediğinden performans düşüşü.

Sanal Dosya Sistemleri

- Windows, dosya sistemi sürücüleri belirtir
- Unix, VFS'ye entegre olur
  - VFS sistem çağrıları kullanıcıdan
  - Alt seviye çağrılar gerçek dosya sistemine yapılır
- Ağ Dosya Sistemini destekler - dosya uzak bir makinede olabilir

Sanal Dosya Sistemleri

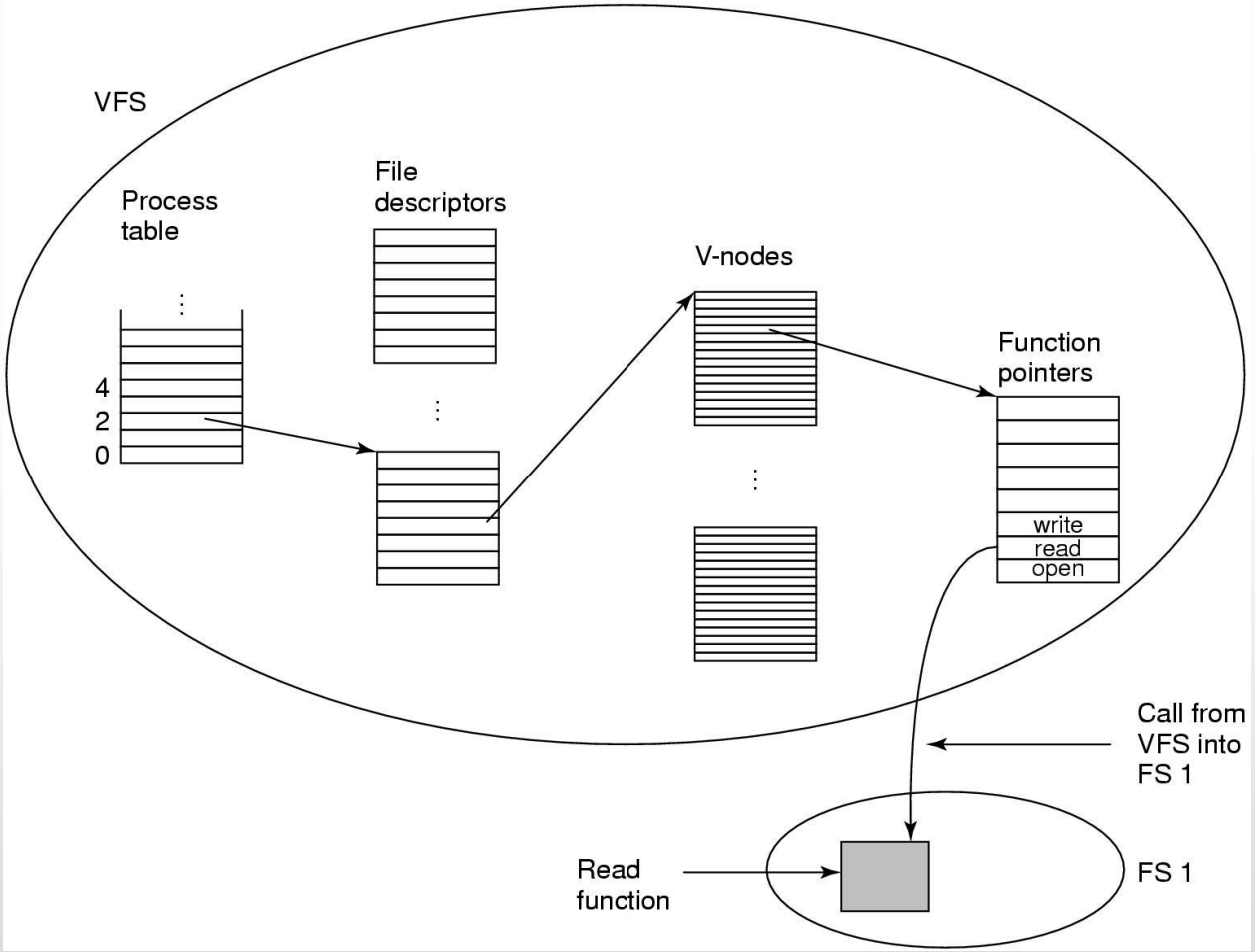


Sanal Dosya Sistemi Nasıl Çalışır

- Dosya sistemi VFS'ye kaydolur (önyükleme sırasında)
- Kayıt sırasında fs, vfs'nin istediği fonksiyon çağrılarının adres listesini sağlar.
- Vfs, yeni fs i-node'dan bilgi alır ve onu bir v-node'a yerleştirir
- Süreç için fd (file descriptor) tablosuna giriş yapar
- Süreç bir çağrı yaptığında (örn. okuma), fonksiyon işaretçileri somut fonksiyon çağrılarına işaret eder

Sanal Dosya Sistemleri

- VFS'nin kullandığı veri yapıları



Dosya Sistemi Yönetimi ve Optimizasyonu

- Disk alanı yönetimi
- Dosya sistemi yedeklemeleri
- Dosya sistemi tutarlılığı
- Dosya sistemi performansı

Disk Alanı Yönetimi

- Bitişik olması gerekmeyen sabit boyutlu bloklar kullanılmalı
- Dosyalar ardışık bayt serisi olarak saklanırsa ve dosya büyüdüğünde taşınması gerekir!
- Optimum (iyi) blok boyutu nedir?
  - Dosya boyutu dağılımı hakkında bilgiye ihtiyaç var.
- Dosya sistemi tasarlarken genel (generic) düşünülmeli

Disk Alanı Yönetimi

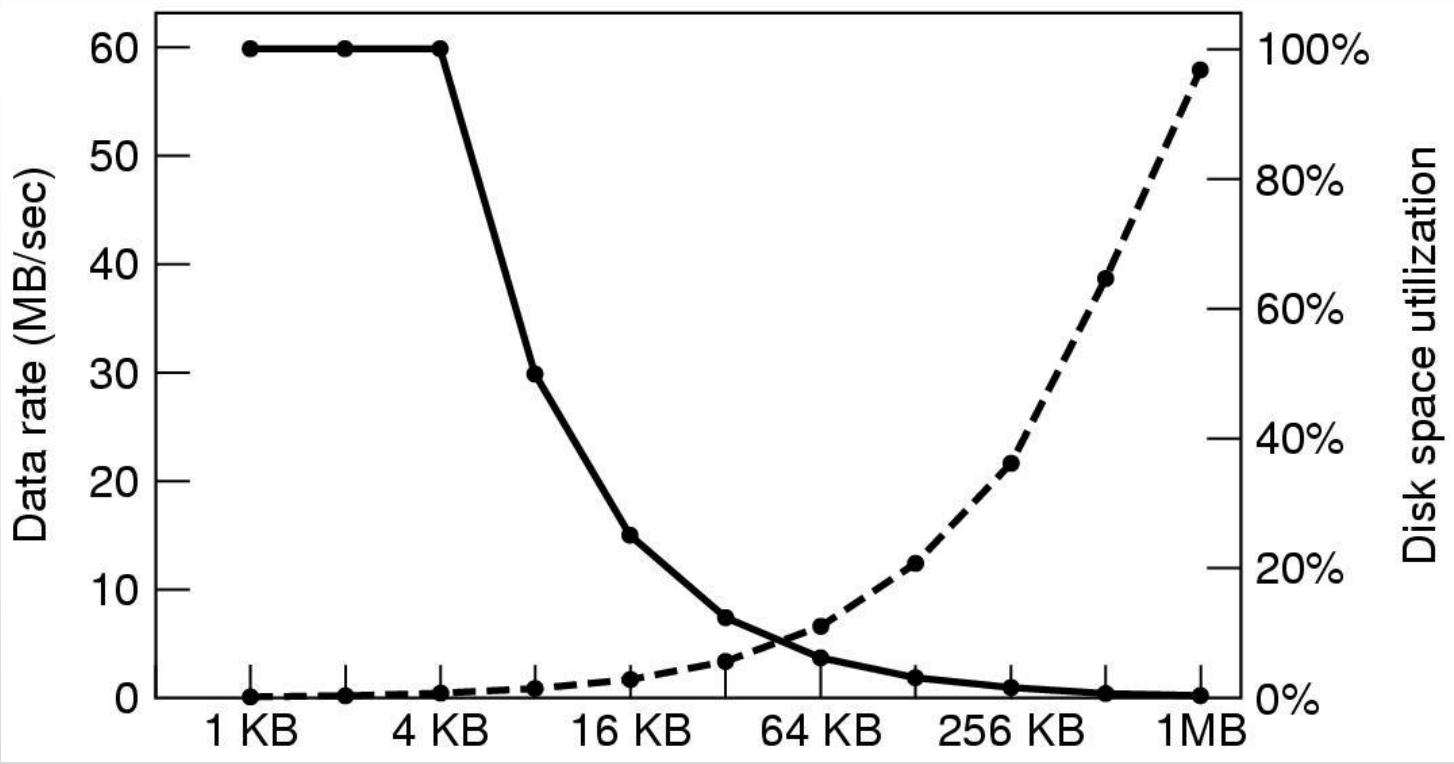
- Verilen boyuttan daha küçük olan dosyaların yüzdesi

Length	VU 1984	VU 2005	Web
1	1.79	1.38	6.67
2	1.88	1.53	7.67
4	2.01	1.65	8.33
8	2.31	1.80	11.30
16	3.32	2.15	11.46
32	5.13	3.15	12.33
64	8.71	4.98	26.10
128	14.73	8.03	28.49
256	23.09	13.29	32.10
512	34.44	20.62	39.94
1 KB	48.05	30.91	47.82
2 KB	60.87	46.09	59.44
4 KB	75.31	59.13	70.64
8 KB	84.97	69.96	79.69

Length	VU 1984	VU 2005	Web
16 KB	92.53	78.92	86.79
32 KB	97.21	85.87	91.65
64 KB	99.18	90.84	94.80
128 KB	99.84	93.73	96.93
256 KB	99.96	96.12	98.48
512 KB	100.00	97.73	98.99
1 MB	100.00	98.87	99.62
2 MB	100.00	99.44	99.80
4 MB	100.00	99.71	99.87
8 MB	100.00	99.86	99.94
16 MB	100.00	99.94	99.97
32 MB	100.00	99.97	99.99
64 MB	100.00	99.99	99.99
128 MB	100.00	99.99	100.00

Disk Alanı Yönetimi

- Düz eğri diskin veri hızını, kesikli eğri disk alanı verimliliğini gösterir. Tüm dosyalar 4 KB'dir.

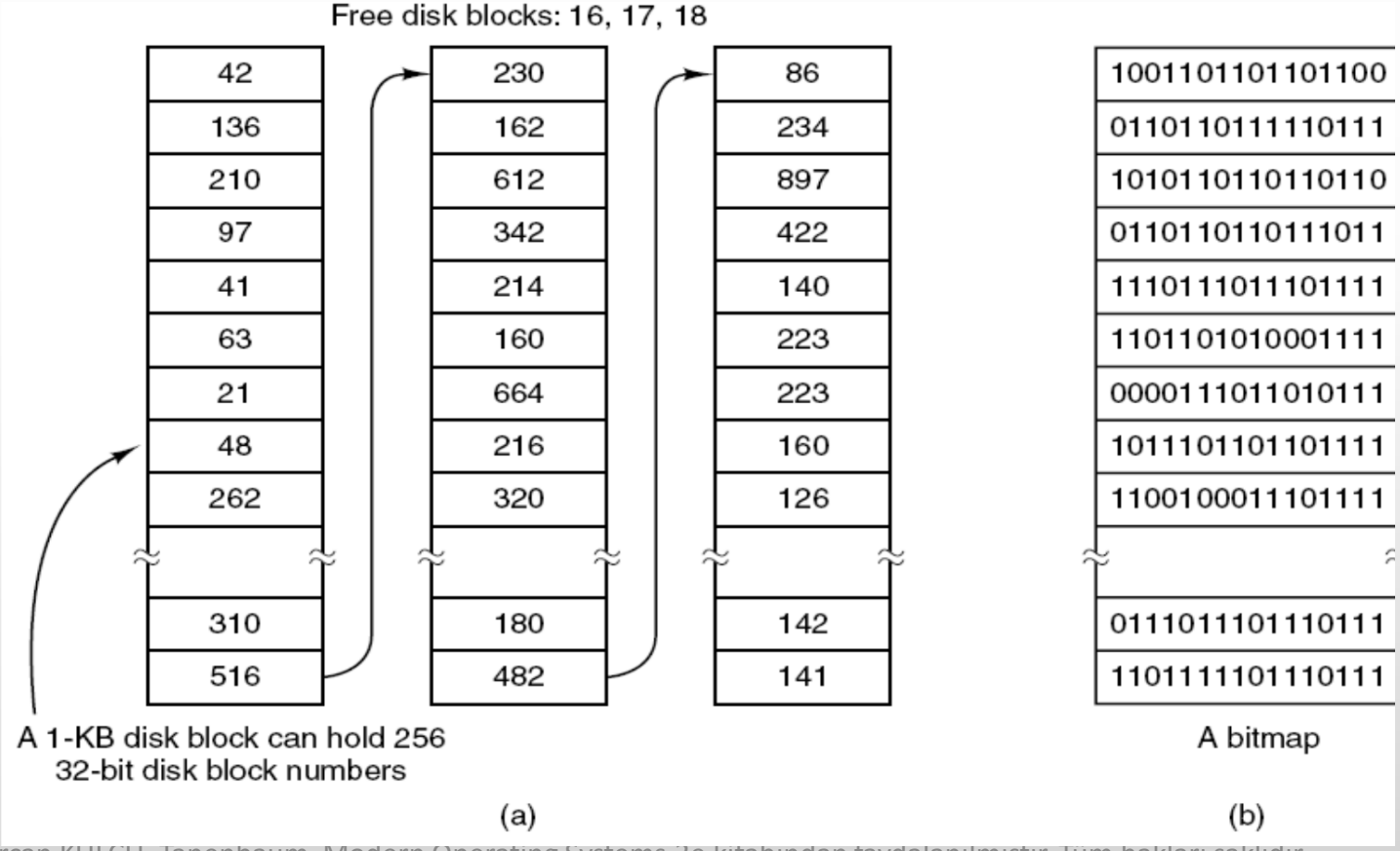


Disk Alanı Yönetimi

- Büyük blok boyutu, daha iyi alan kullanımına, ancak daha kötü aktarım (transfer) kullanımına neden olur
- Alan ve veri hızı birbiriyle ters orantı (trade-off)
- Kesin iyi bir çözüm yok (Nature wins this time)
- Disk yeterince büyükse, büyük blok boyutu (64 KB) kullan

Boş Blokların İzini Tutma

(a) Bağlı liste halinde (b) biteşlem olarak

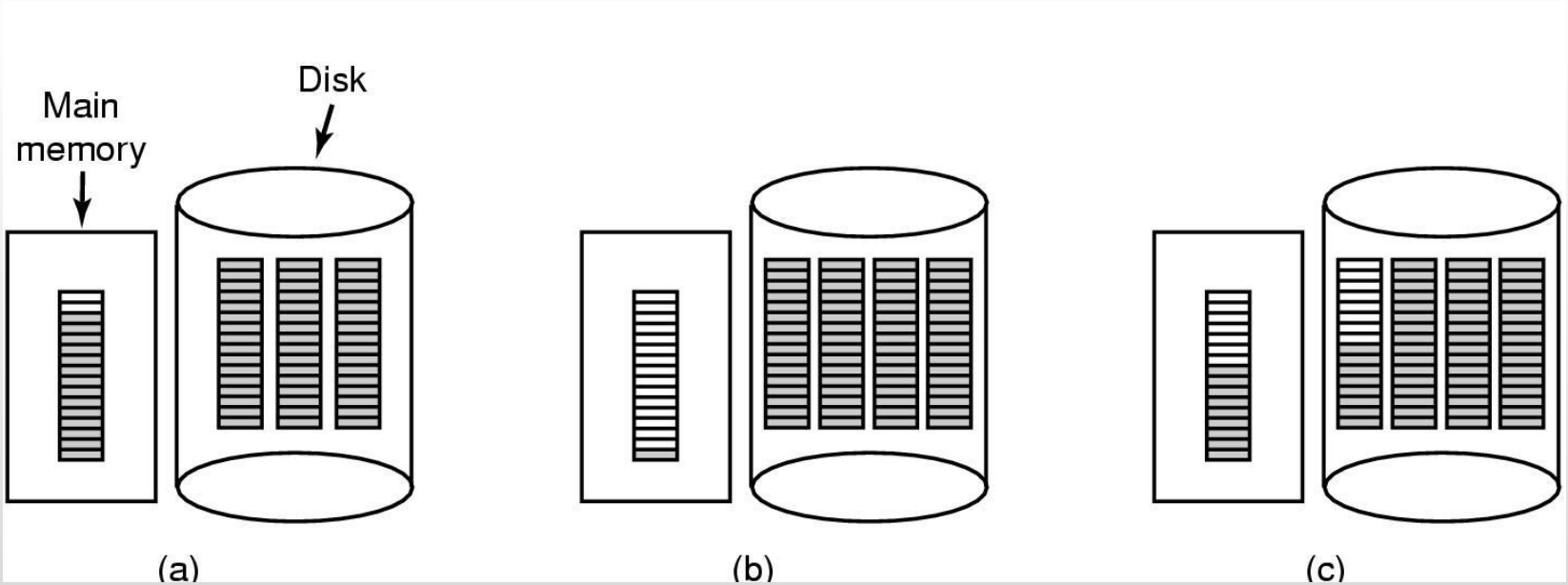


Boş Blokların İzini Tutma

- Bağlantı ihtiyacı ~1,9 milyon blok
- Biteşlem haritası ~60.000 bloğa ihtiyaç duyar
- Herhangi bir anda ana bellekte yalnızca bir işaretçi bloğuna ihtiyaç vardır. Doldur => bir tane daha al

Boş Blokların İzini Tutma

- (a) Bellekte dolmaya yakın bir işaretçiler bloğu ve diskte üç işaretçi bloğu. (b) Üç bloklu bir dosyayı serbest bıraktıktan sonra. (c) Üç boş bloğu işlemek için alternatif bir strateji.

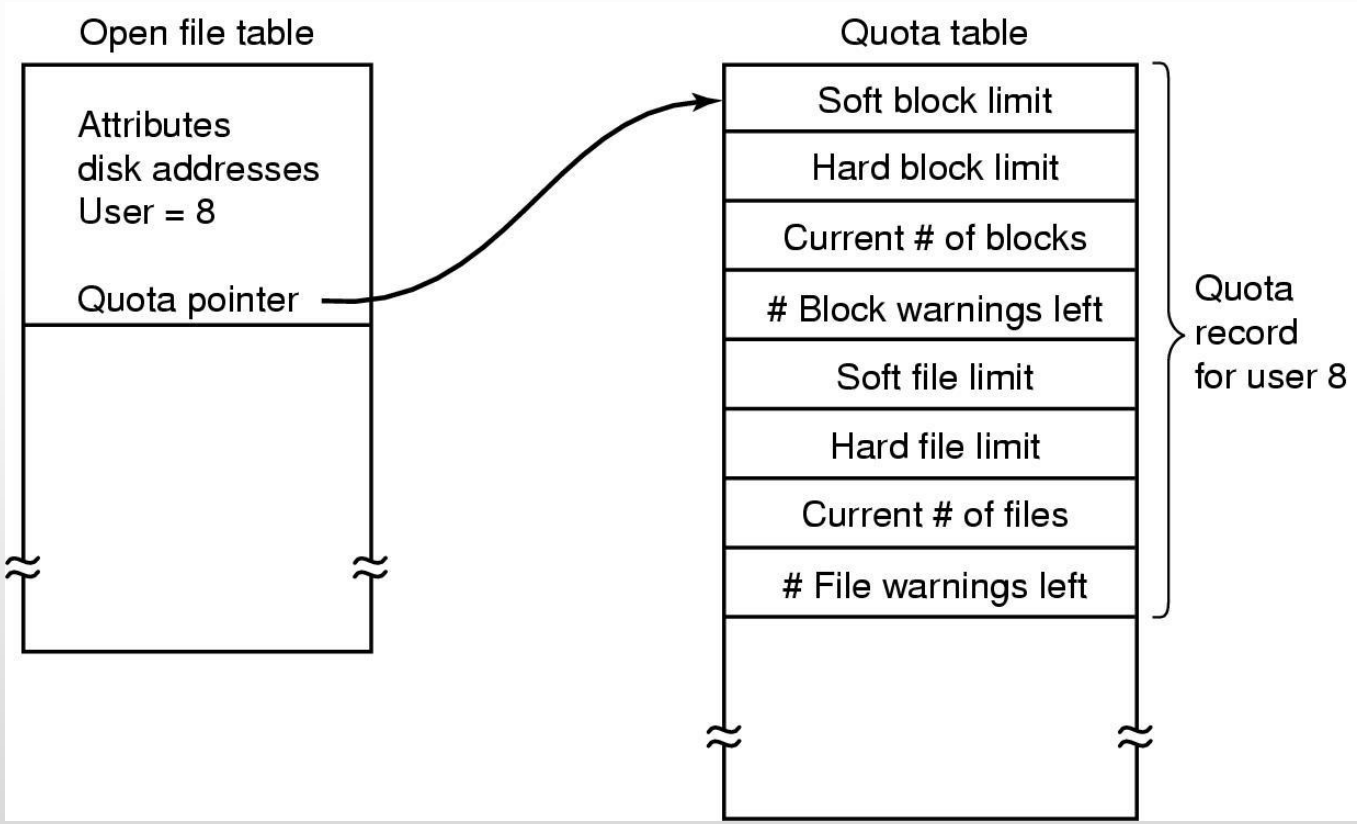


Disk Kotaları

- Açık dosyalar tablosundaki girdi (entry), kota tablosuna işaret eder
- Her açık dosya için bir girdi var
- Kullanıcıların disk kotasına sınır koyar (soft, hard)

Disk Kotaları

- Kota tablosunda kullanıcı bazında izlenir.



Dosya Sistemi Yedeklemeler

- Yedeklemeler genellikle iki olası sorundan dolayı yapılır
  - Felaketten kurtulmak için (disk çökmesi)
  - Dikkatsizlik sonucu (yanlışlıkla silinen dosya)
- Moral
  - Dikkatli ol
  - Yedekle
- Teypler yüzlerce gigabayt tutar ve çok ucuzdur

Dosya Sistemi Yedeklemeler

- Tüm dosyaları yedeklemeye gerek yok
- Üreticinin CD'lerinden, internet'den ikili dosyalar bulunabilir
- Geçici dosyaların yedeklenmesi gerekmez
- Özel dosyaların (G/Ç) yedeklenmeye ihtiyacı yoktur

Kademeli Olarak Yedekleme

- Son dökümden (dump) bu yana değiştirilen dosyaların haftalık/aylık ve günlük dökümünü tamamla
- fs'yi geri yüklemek için tam döküm gerekli
- Değiştirilmiş dosyaları dahil etmek için iyi algoritmalara ihtiyaç var
- Problem - verileri dökümden önce sıkıştırmak istiyorum, ancak bandın bir kısmı kötüyse…
- Problem - sistem kullanılırken döküm performans açısından zor. Anlık görüntü (snapshot) algoritmaları mevcut

Döküm Stratejileri - Fiziksel

- Fiziksel olarak tüm her şey dökülür.
- Uygulaması basit
- Dökülmek istenmeyenler
  - kullanılmayan bloklar: programın kullanılmayan blok listesine erişmesi ve kullanılan bloklar için blok numarasını teybe yazması gerekir
  - kötü bloklar Disk denetleyicisi kötü blokları algılamalı ve değiştirmelidir veya nerede olduklarını bilmelidir (işletim sistemi tarafından kötü blok alanında tutulurlar)

Döküm Stratejileri - Fiziksel

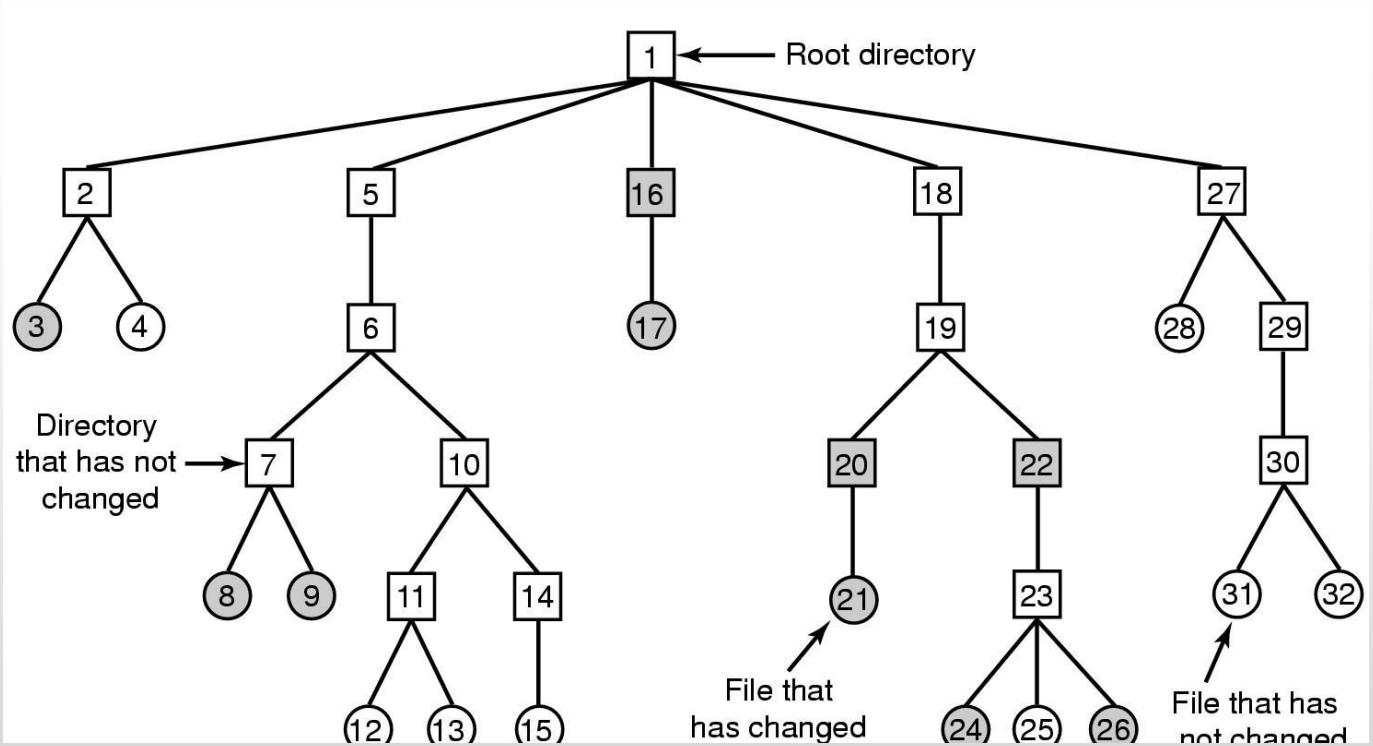
- Uygulaması kolay
- Belirli bir dizini atlayamaz (skip)
- Kademeli dökümler yapamaz (incremental)
- Dosyalar tek tek geri yüklenemez
- Mantıksal döküm stratejisi daha yaygın kullanılır

Döküm Stratejileri - Mantıksal

- Bir dizinden başlar ve verilen zamandan bu yana değişen tüm dosyaları/dizinleri özyinelemeli olarak döker.
- Dosyaları/dizinleri değiştirilmiş dosya/dizine giden yola döker
- Bu sayede yolu (path) farklı bir bilgisayarda geri yükleyebilir
- Tek bir dosyayı geri yükleyebilir

Dosya Sistemi Yedekleme

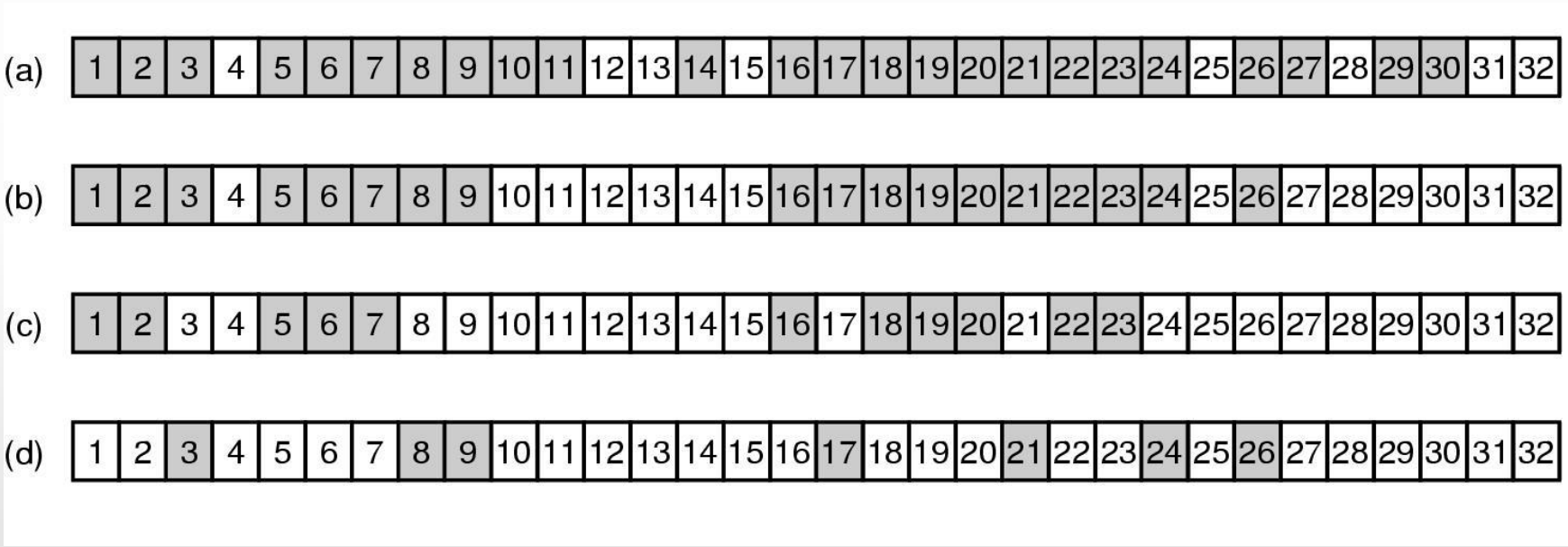
- Kareler dizinleri, daireler dosyaları gösterir. Gölgeli öğeler, son dökümden bu yana değiştirilenler. Her dizin ve dosya, i-node numarasıyla etiketlenir.



Mantıksal Döküm Algoritması

- i-node tarafından indekslenmiş biteşlem kullanır
- 4 aşamadan oluşur
- Aşama 1 - kökte başlar ve değiştirilen tüm dosyalar ve dizinler için bitleri işaretler (a)
- Aşama 2 - ağacı gezer, içinde değiştirilmiş dosya olmayan dizinlerin işaretini kaldırır (b)
- Aşama 3 - i-node'ları gözden geçirir ve işaretli dizinlerin dökümünü alır (c)
- Aşama 4 - döküm dosyaları (d)

Mantıksal Döküm Algoritması



Diskteki Dosyayı Geri Yükleme

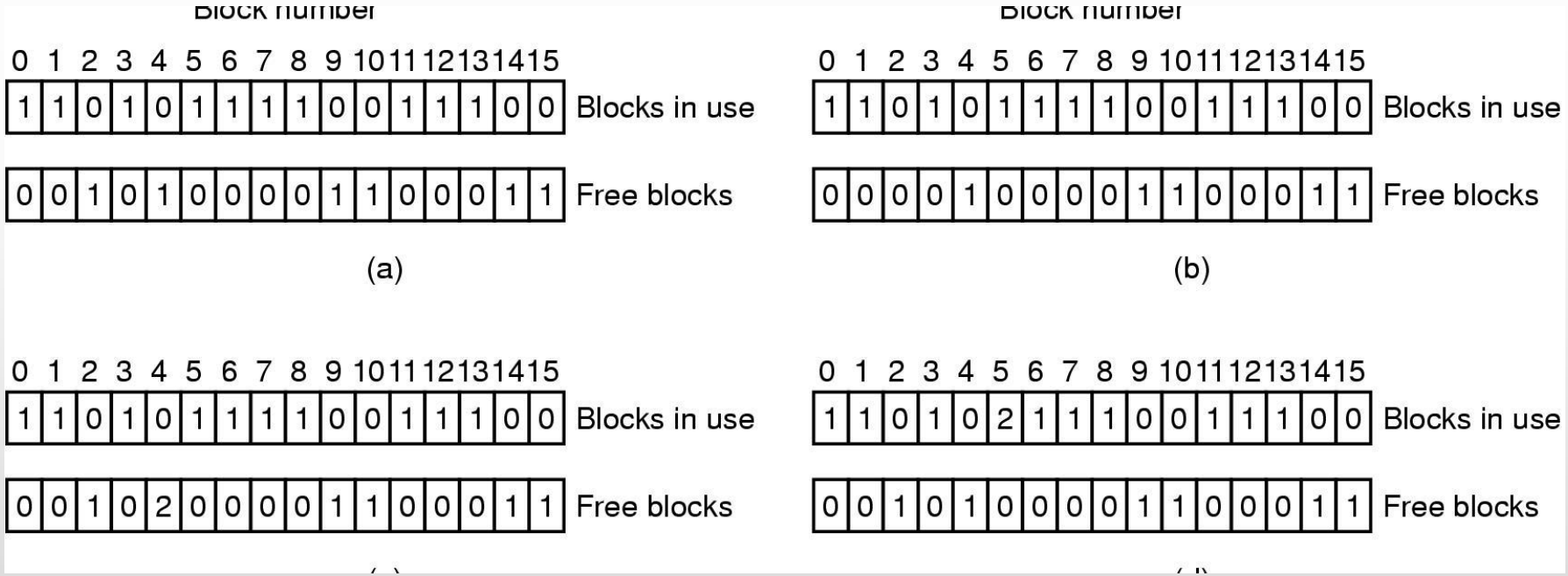
- Diskte boş fs ile başlanır
- Son tam döküm geri yüklenir.
- Önce dizinler, sonra dosyalar.
- Ardından kademeli olarak dökümler geri yüklenir (kaset, teyp üzerinde sıralıdırlar)

Dosya Sistemi Tutarlılığı

- Blokların tümü yazılmadan önce kilitlenme/kaza (crash), dosya sistemini tutarsız bir durumda bırakır
- Bloklarda ve dosyalarda tutarlılığı kontrol etmek için yardımcı programlara ihtiyaç var. Unix'te fsck, Windows'ta scandisk
- İki tablo kullanılır
- Bir dosyada bir blok kaç kez bulunur?
- Bir blok boş alanların tutulduğu listede kaç kez var?
- Cihaz tüm i-node'ları okur, sayaçları artırır

Dosya Sistemi Tutarlılığı

- Dosya sistemi durumları. (a) Tutarlı. (b) Eksik blok. (c) Serbest listede yinelenen blok. (d) Yinelenen veri bloğu.



Çözüm

- Eksik blok (b) - serbest listeye koy
- Serbest listede yinelenen blok (c) - serbest listeyi yeniden oluştur
- Yinelenen veri bloğu (d) - kullanıcıyı bilgilendir. Bir dosya kaldırılırsa blok her iki listede de görünür.

Dosya Sistemi Tutarlılığı

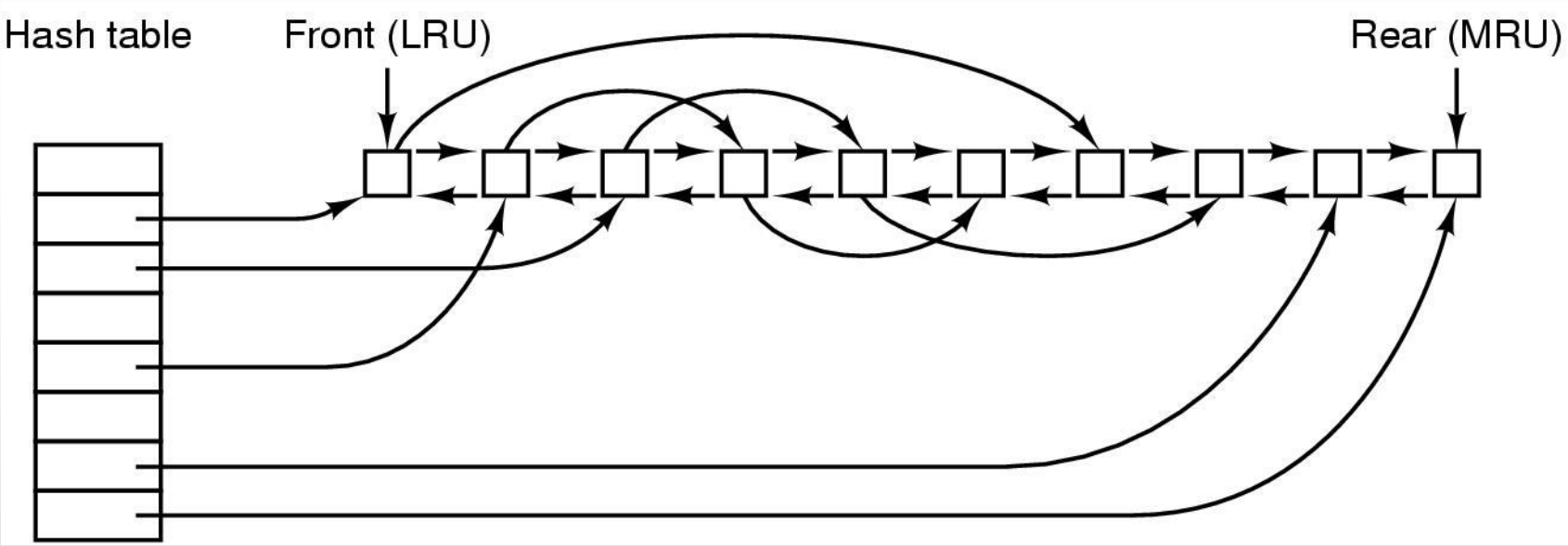
- Bloklar yerine dosyalara bak
- Dosya başına bir sayaç tablosu kullan
- Kök dizinde başla, aşağı in, dosya bir dizinde her göründüğünde sayacı artır
- Sayaçları i-node'lardan gelen bağlantı (link) sayılarıyla karşılaştırır.
  - Tutarlı olmak için aynı olmak zorunda

Dosya Sistemi Performansı

- Bellekten sözcük okuma: 32 ns
- Disk: 5-10 ms arama + 100 MB/sn aktarım
- Bellekte önbellek blokları
- Yönetmek için hash tablosu (cihaz, disk adresi)
- Önbellek bloklarını değiştirmek için algoritmaya ihtiyaç var - sayfalama algoritmaları kullanılır, örneğin LRU

Tampon Önbellek Veri Yapıları

- Buffer cache



Yer Değiştirme

- LRU ile ilgili bir problem - bazı bloklar nadiren kullanılıyor, ancak bellekte olmaları gerekiyor
- i-node değişiklik olduğunda diske yeniden yazılması gerekir. Çökme durumunda, sistem tutarsız bir durumda kalabilir
- LRU'yu değiştir
  - Blok tekrar kullanılabilir mi?
  - Blok, dosya sisteminin tutarlılığı için önemli mi?

Yer Değiştirme

- Kategorileri kullan: i-nodes, dolaylı (indirect) bloklar, dizin blokları, tam veri blokları, kısmi veri blokları
- İhtiyaç duyulacak olanları arkaya koy
- Blok ihtiyaç duyulup ve sonra değiştirilmişse, en kısa zamanda diske yazılır

Yer Değiştirme

- Değiştirilmiş blokları bir an önce diske koymak için
- UNIX senkronizasyon: değiştirilmiş tüm blokları diske yazılmaya zorlar.
- Güncelleme programı her 30 saniyede bir kontrol eder
- Windows: blok değiştiğinde hemen diske yaz (Write through cache)

İleriyi Okuma (read ahead)

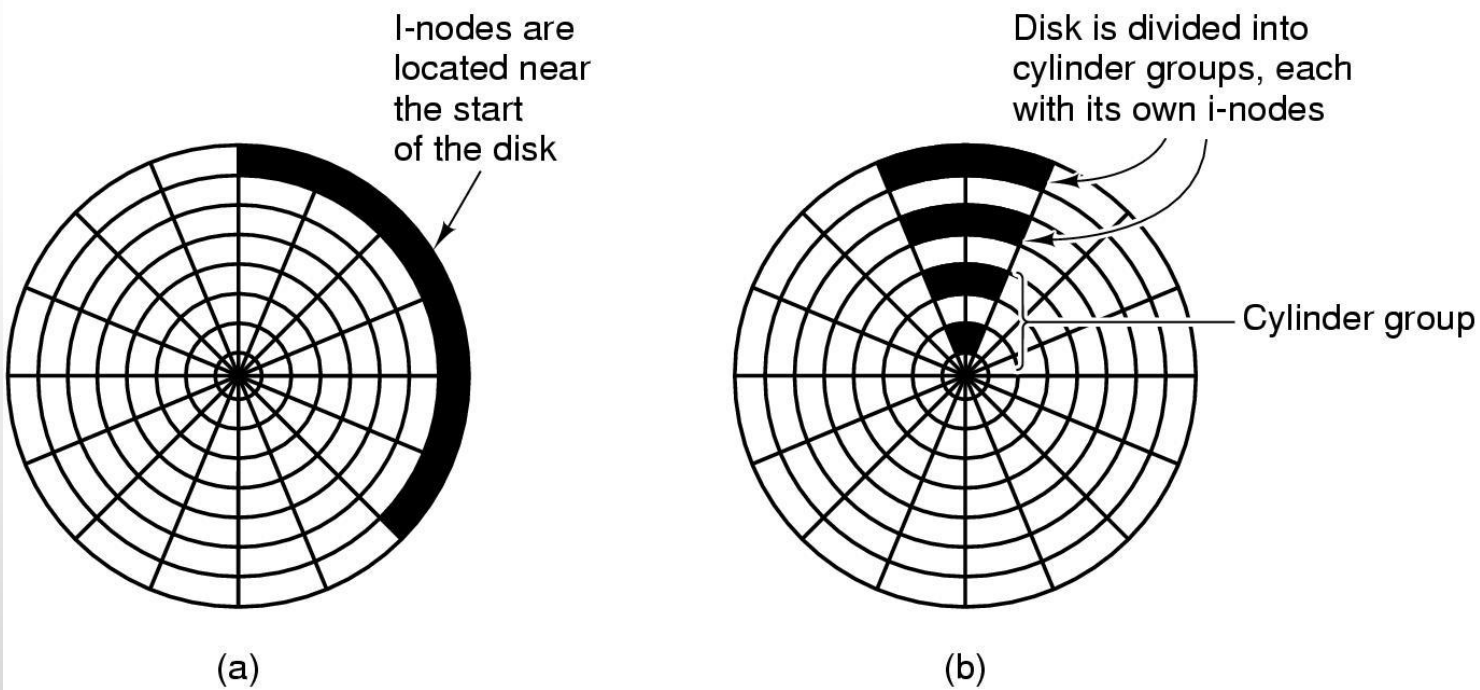
- Önbelleğe almak için k bloğu okunduğunda, k+1 bloğu önbellekte değilse onu da oku
- Yalnızca sıralı dosyalar için geçerlidir
- Dosyanın sıralı mı yoksa rastgele mi olduğunu belirlemek için bir bit kullanılır. Bir arama yap, bitin değerini çevir (flip).

Kol Hareketini Azaltmak

- Sıralı olarak erişilecek blokları birbirine yakın yerleştirmeye çalış.
- Bellekte bir biteşlem tutarak yapmak kolaydır, blokları boş liste ile arka arkaya yerleştirmek gerekir
- Önbellek blokları 1 KB ise, yer tahsisini boş (free) listeden 2 KB parçalar halinde yap
- Ardışık blokları aynı silindire koymaya çalış
- i-node'ları arama süresini azaltmak amacıyla yerleştir

Kol Hareketini Azaltmak

(a) Diskin başına yerleştirilen I-düğümleri. (b) Disk, her biri kendi blokları ve i-düğümleri olan silindir gruplarına bölünmüştür.

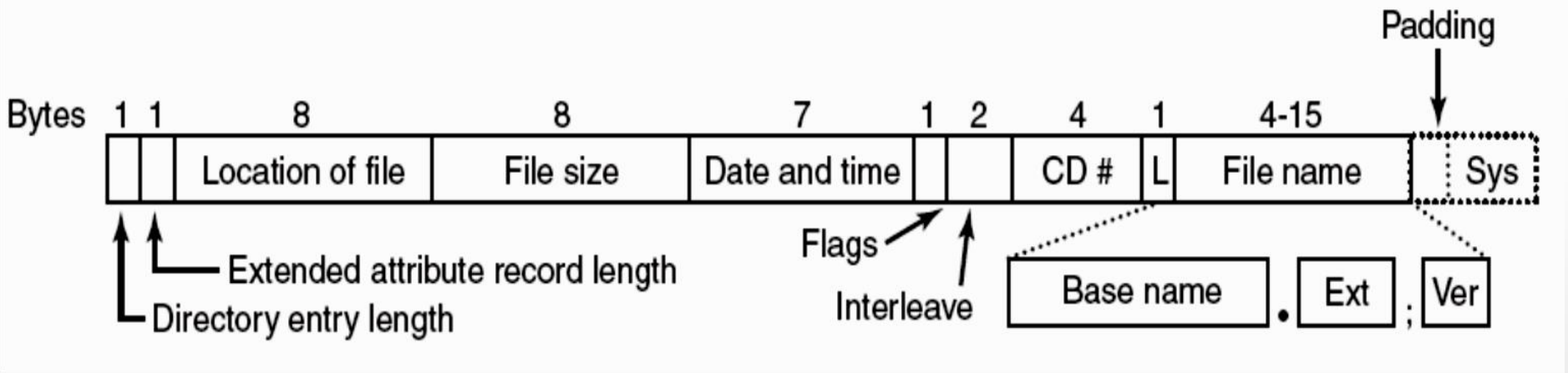


Diskleri Birleştirme (defrag)

- Başlangıçta, dosyalar diske bitişik olarak yerleştirilir.
- Zamanla delikler (hole) oluşur
- Windows defrag programı, bir dosyanın farklı bloklarını bir araya getirir
- Linux defrag işlemini desteklemez. Farklı dosyalar birbirine uzak yerleştirilir.

The ISO 9660 Dosya Sistemi

- Dizin girişi.



Rock Ridge Interchange Protokolü (RRIP)

- CD-ROM'larda kullanılan, dosya sisteminin yeteneklerini artıran ve diskte depolanan dosyalar hakkında ek bilgi sağlayan ISO 9660 dosya sistemi biçiminin bir uzantısı.
- Dosya sahipliği, izinler ve sembolik bağlantılar hakkında ek bilgiler sağlayarak CD-ROM'ların Unix tabanlı sistemlerle uyumluluğunu geliştirir.
- Unix tabanlı sistemlerle geliştirilmiş uyumluluk, uzun dosya adları için destek ve Unix tarzı sembolik bağlantılar için destek.
- Unix olmayan sistemlerde sınırlı destek ve daha eski CD-ROM sürücüleriyle olası uyumluluk sorunları.

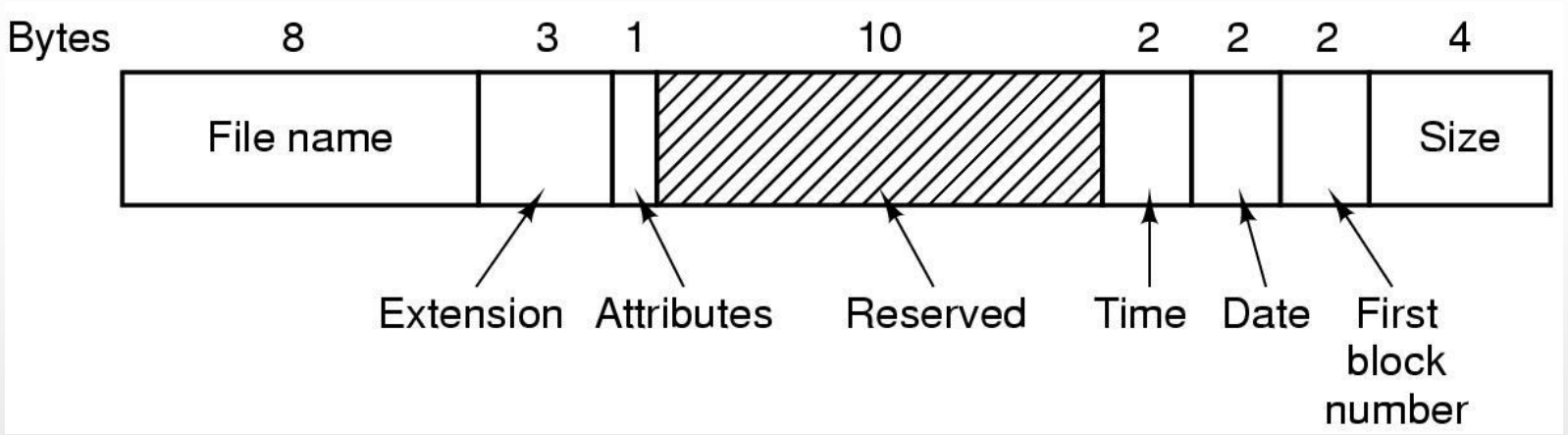
Rock Ridge Interchange Protokolü (RRIP)

- PX - POSIX attributes. POSIX öznitelikleri
- PN - Major ve minor cihaz numaraları
- SL - Symbolic link. Sembolik bağ
- NM - Alternative name. Seçenek adı
- CL - Child location. Çocuk konumu
- PL - Parent location. Ebeveyn konumu
- RE - Relocation. Yer değiştirme
- TF - Time stamps. Zaman damgaları

Joliet Uzantısı (extension)

- CD-ROM'larda kullanılan, dosya sisteminin yeteneklerini artıran ve Unicode karakterlerini destekleyen ISO 9660 dosya sistemi biçiminin bir uzantısı.
- Unicode karakterleri ve uzun dosya adları için destek sağlayarak CD-ROM'ların Windows tabanlı sistemlerle uyumluluğunu geliştirir.
- Windows tabanlı sistemlerle geliştirilmiş uyumluluk, uzun dosya adları için destek ve Unicode karakterler için destek.
- Windows olmayan sistemlerde sınırlı destek ve daha eski CD-ROM sürücüleriyle olası uyumluluk sorunları.
- Dizin, sekiz seviyeden daha derine inebilir. Uzantıları olan dizin adları

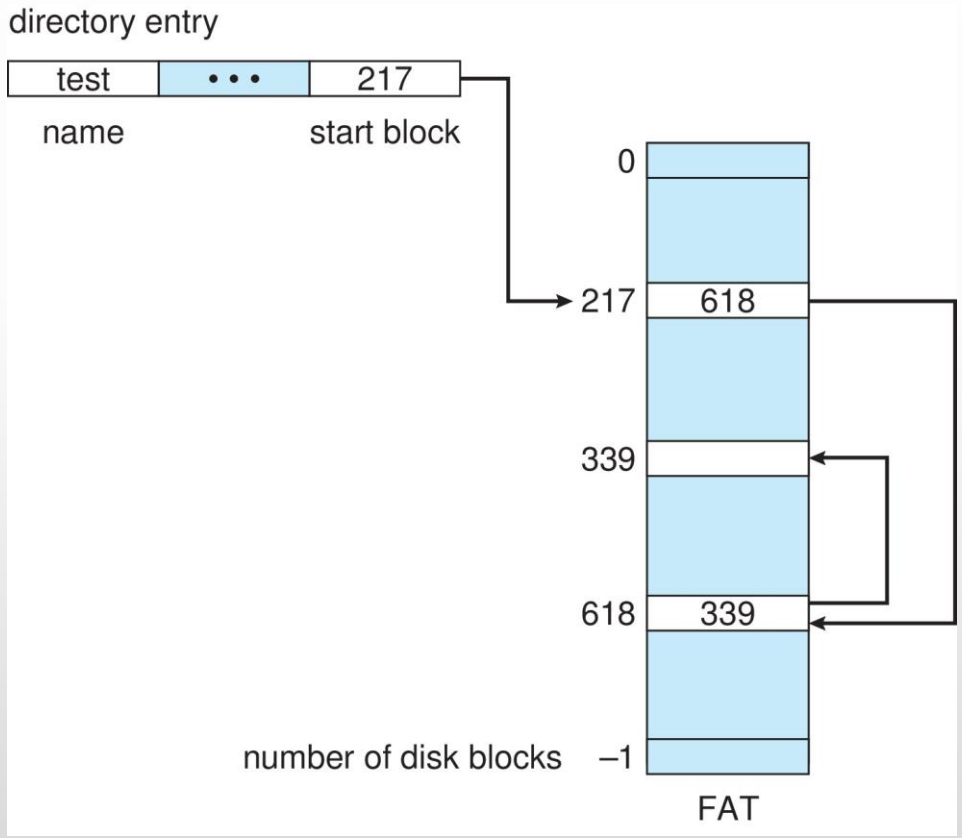
MS-DOS Dosya Sistemi – Dizin Girdisi



Blok Boyutları için Maksimum Bölüm Boyutu

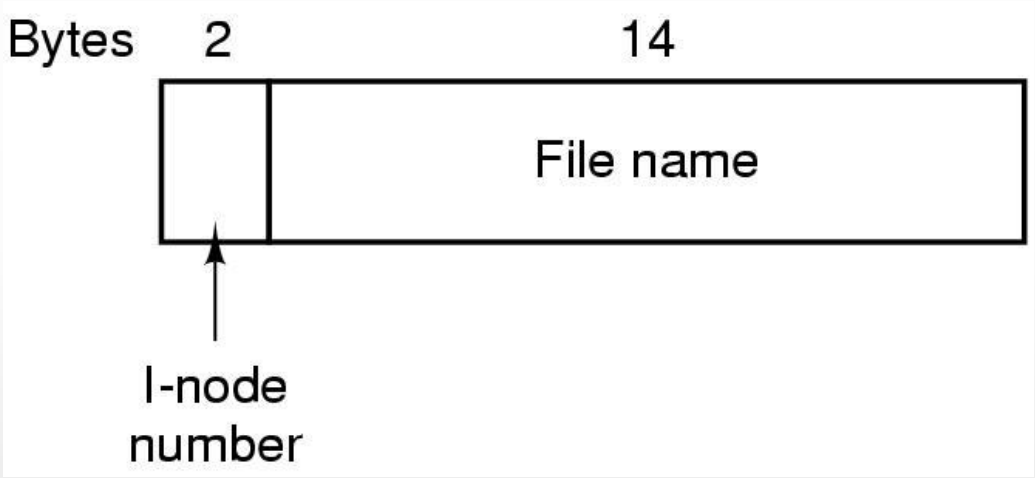
Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

File Allocation Table

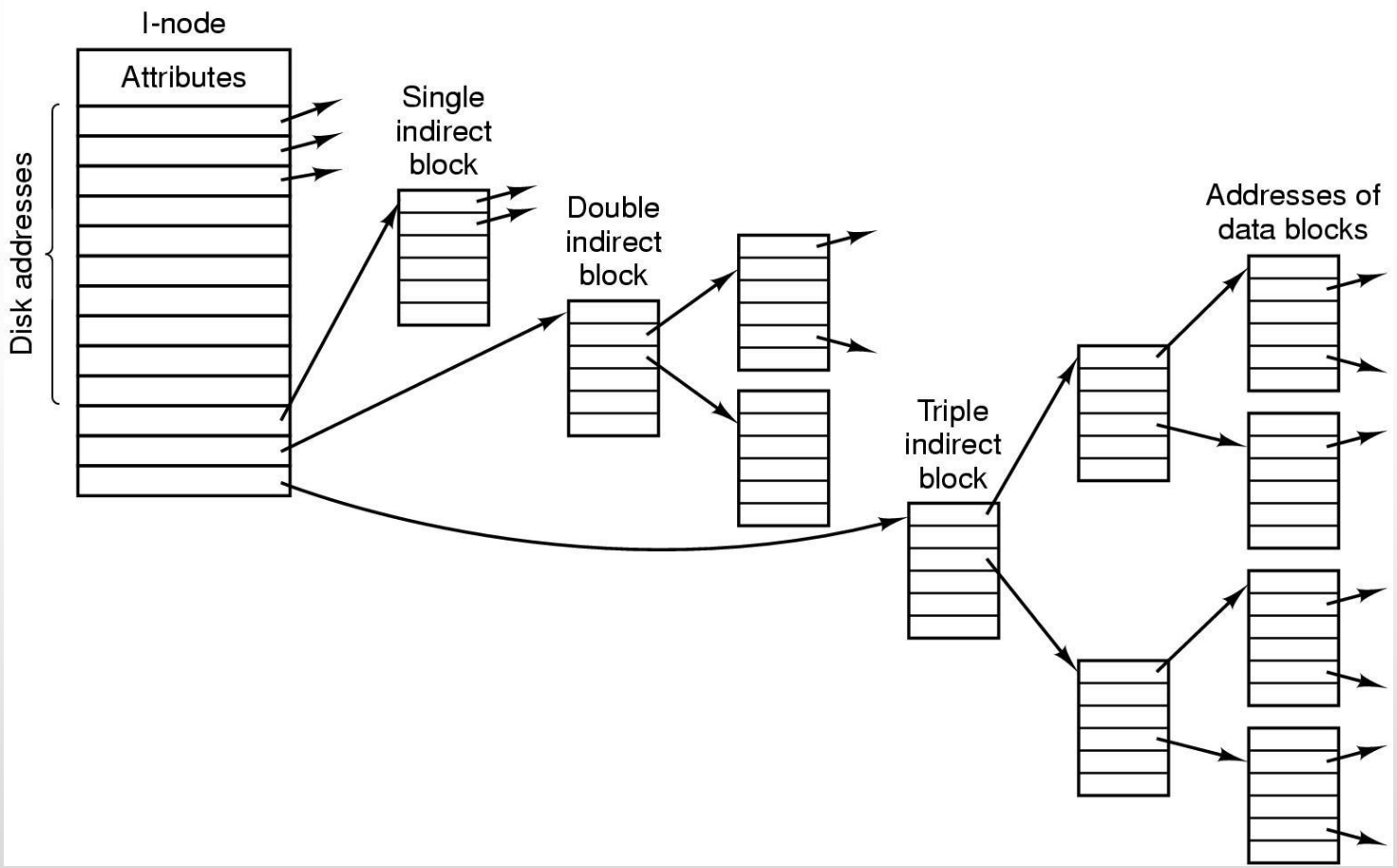


UNIX Version 7 – Dizin Girdisi

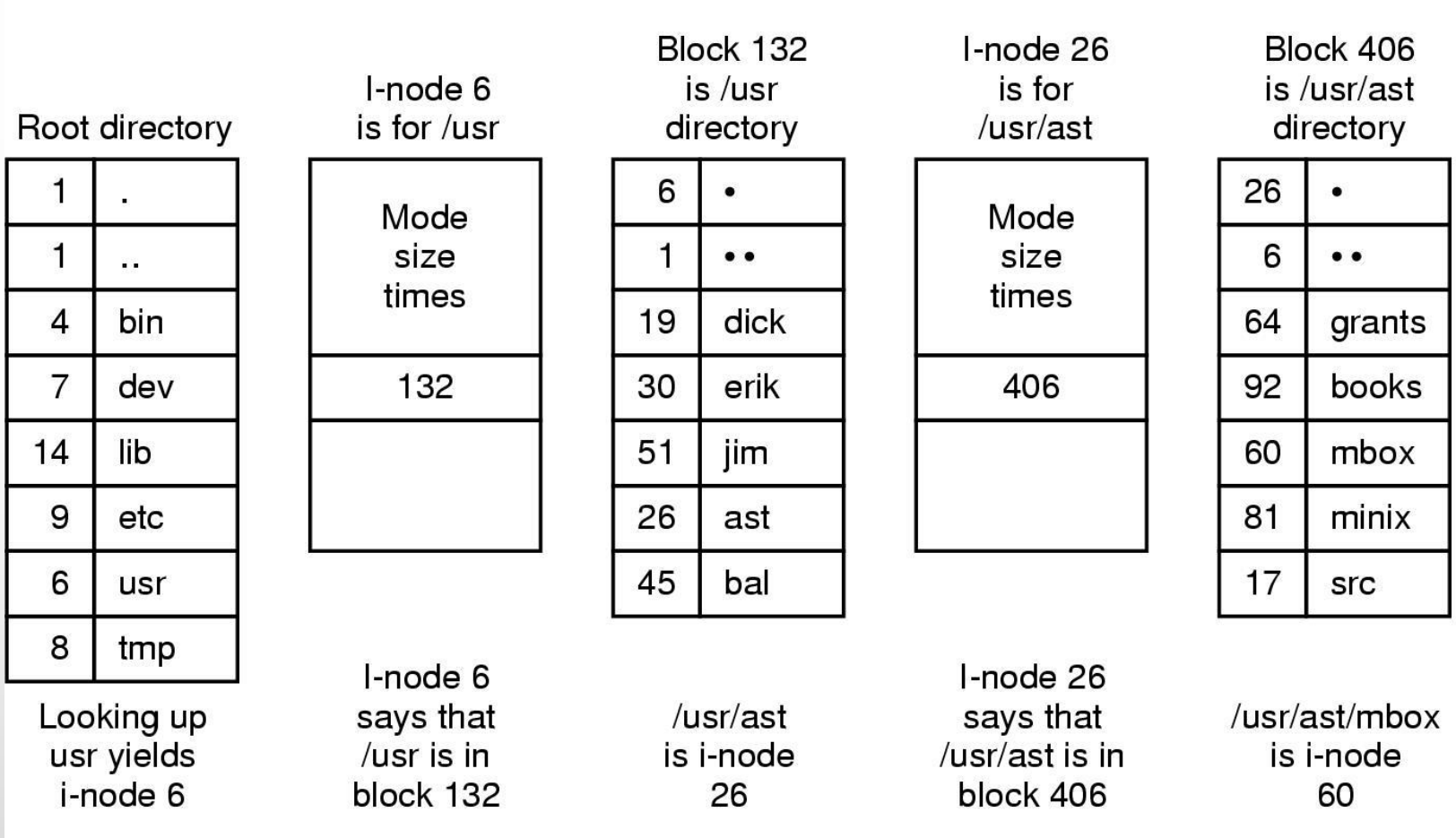
- .



UNIX i-node



/usr/ast/mbox.conf arama adımları



SON