Bölüm 4: 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 Örnekleri

- .

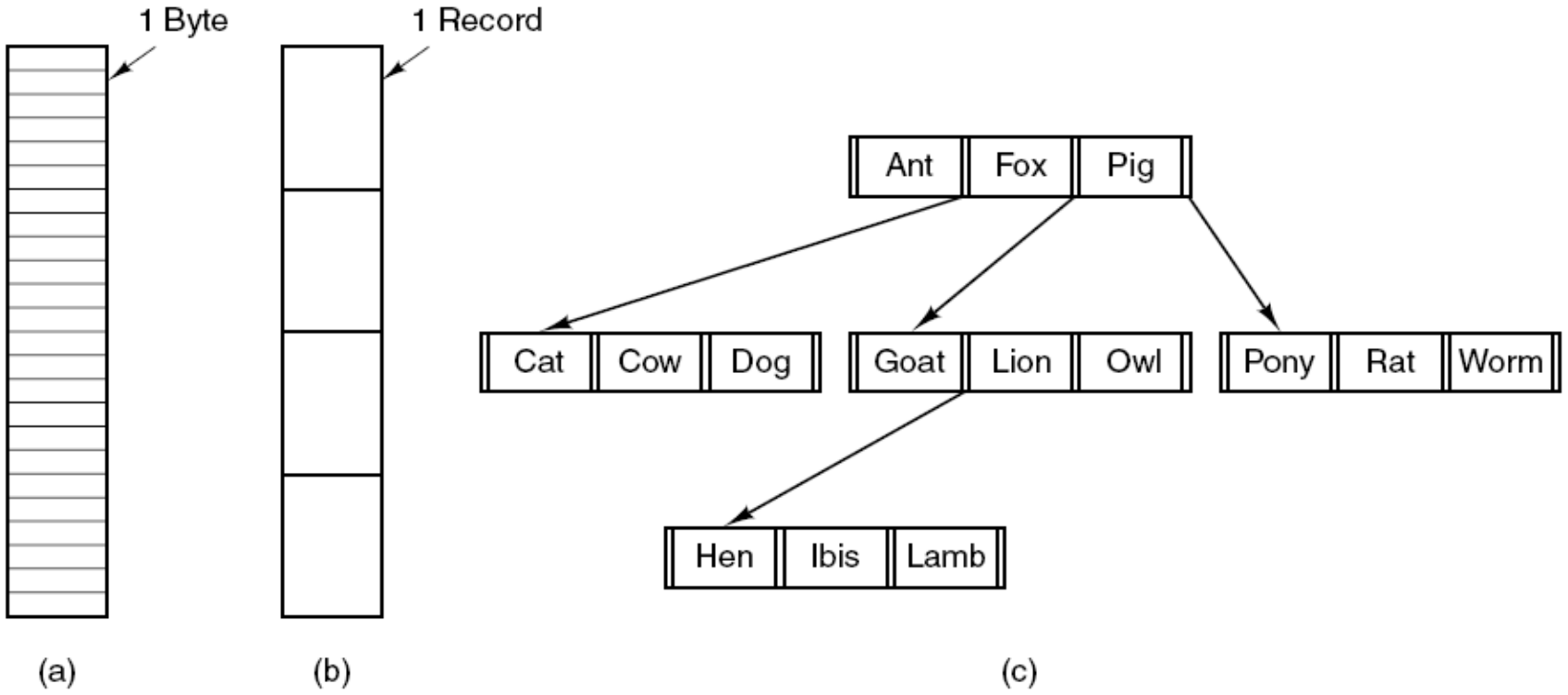
Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
file.mpg	Movie encoded with the MPEG standard
file.o	Object file (compiler output, not yet linked)
file.pdf	Portable Document Format file
file.ps	PostScript file
file.tex	Input for the TEX formatting program
file.txt	General text file
file.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

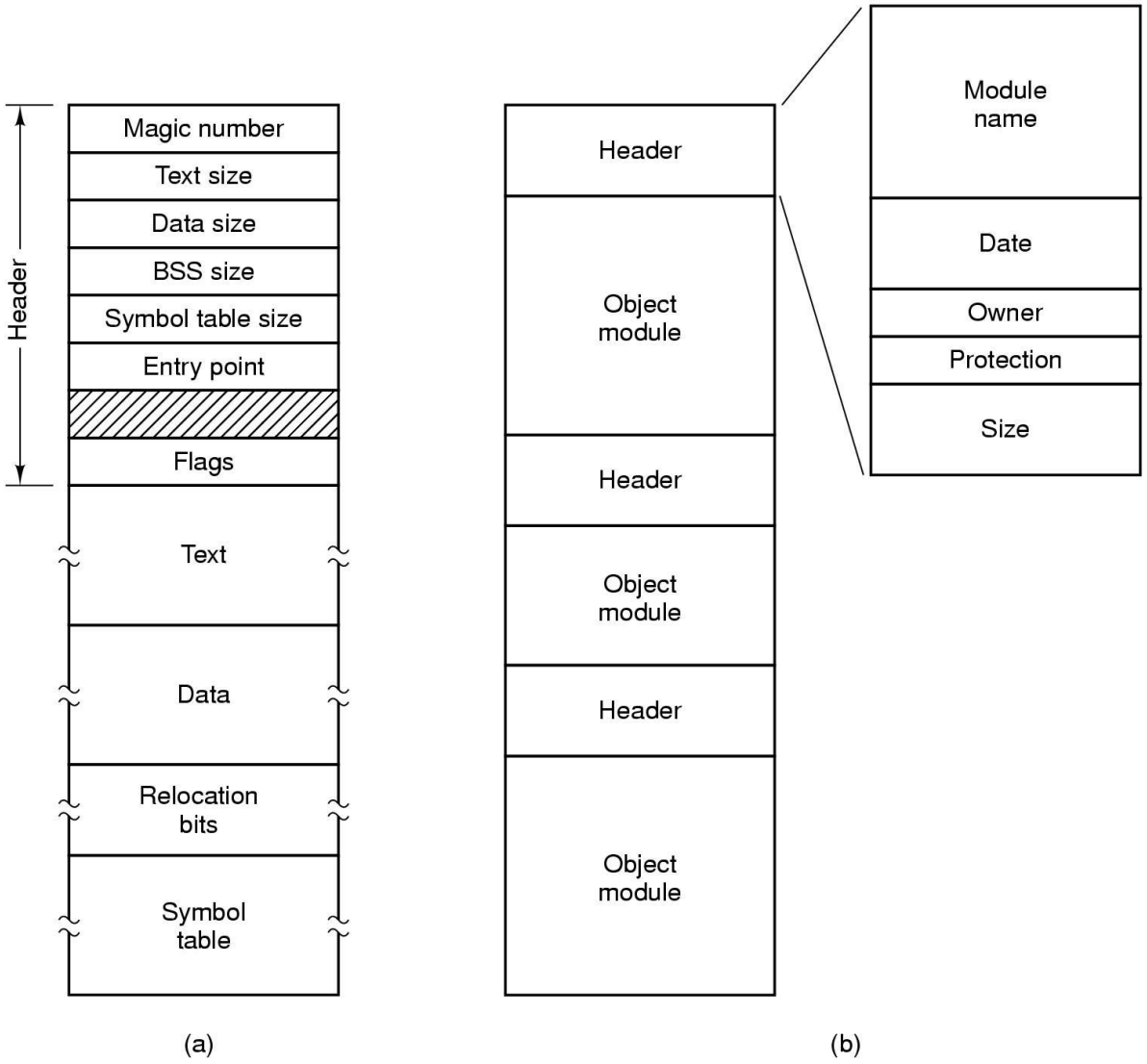
- ASCII veya ikili (binary)
- ASCII
  - Yazdırılabilir
  - Programları bağlamak için boru hattı (pipe) kullanılabilir (ASCII üretiyor/tüketiyorsa)

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
- (b) Derlenmiş ancak bağlanmamış arşiv kütüphanesi



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

- .

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file was last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Stat Veri Yapısı

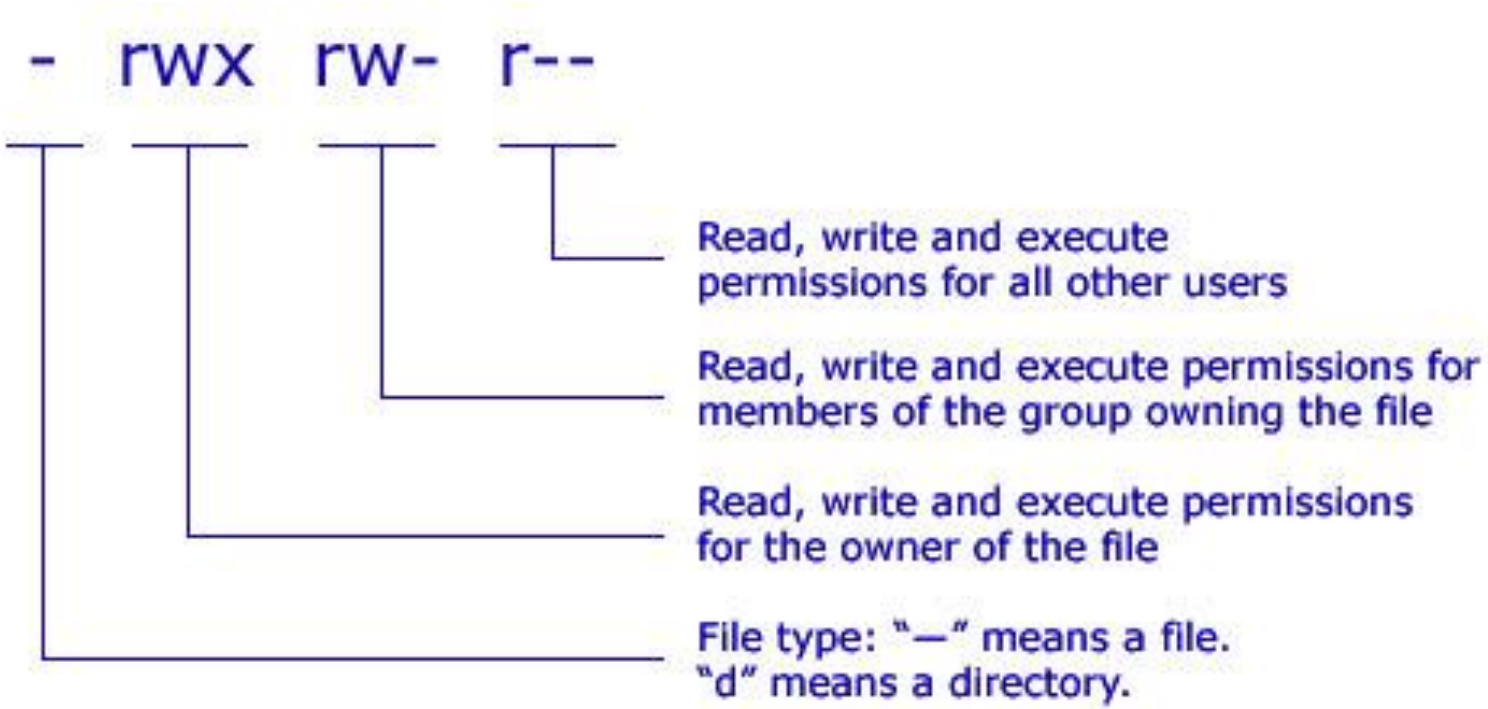
```
Struct stat{
        mode_t      st_mode;     // file type and mode (permission)
        ino_t       st_ino;      //inode number
        dev_t       st_dev;      //device number
        dev_t       st_rdev;     //device number (special)
        nlink_t     st_nlink;    //number of links
        uid_t       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

- .



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

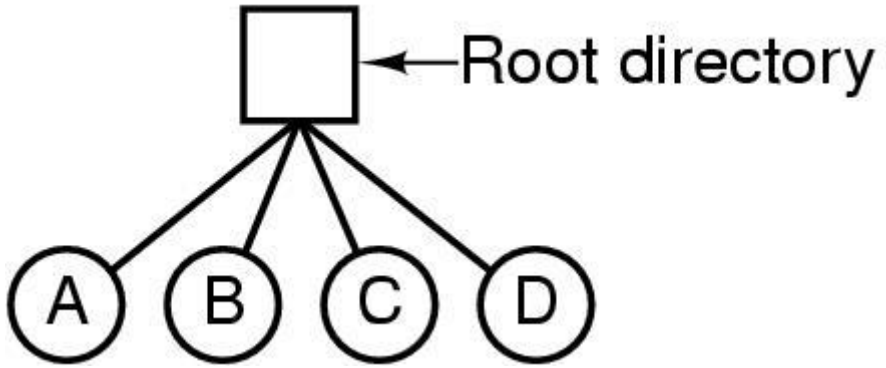
- abc dosyasını xyz'ye kopyalar
- Eğer xyz varsa üzerine yazılır
- Yok ise yaratılır
- Sistem çağrıları kullanılır (okuma, yazma)
- 4K boyutunda parçalar halinde okur ve yazar
- abc dosyasından bir tampon belleğe oku (read sistem çağrısı)
- Tampondan xyz dosyasına yaz (write sistem çağrısı)

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 (ssymbolic link)
  - Başka bir dosyayı adlandıran küçük bir dosyaya işaret (point) eden bir ad oluşturulur.

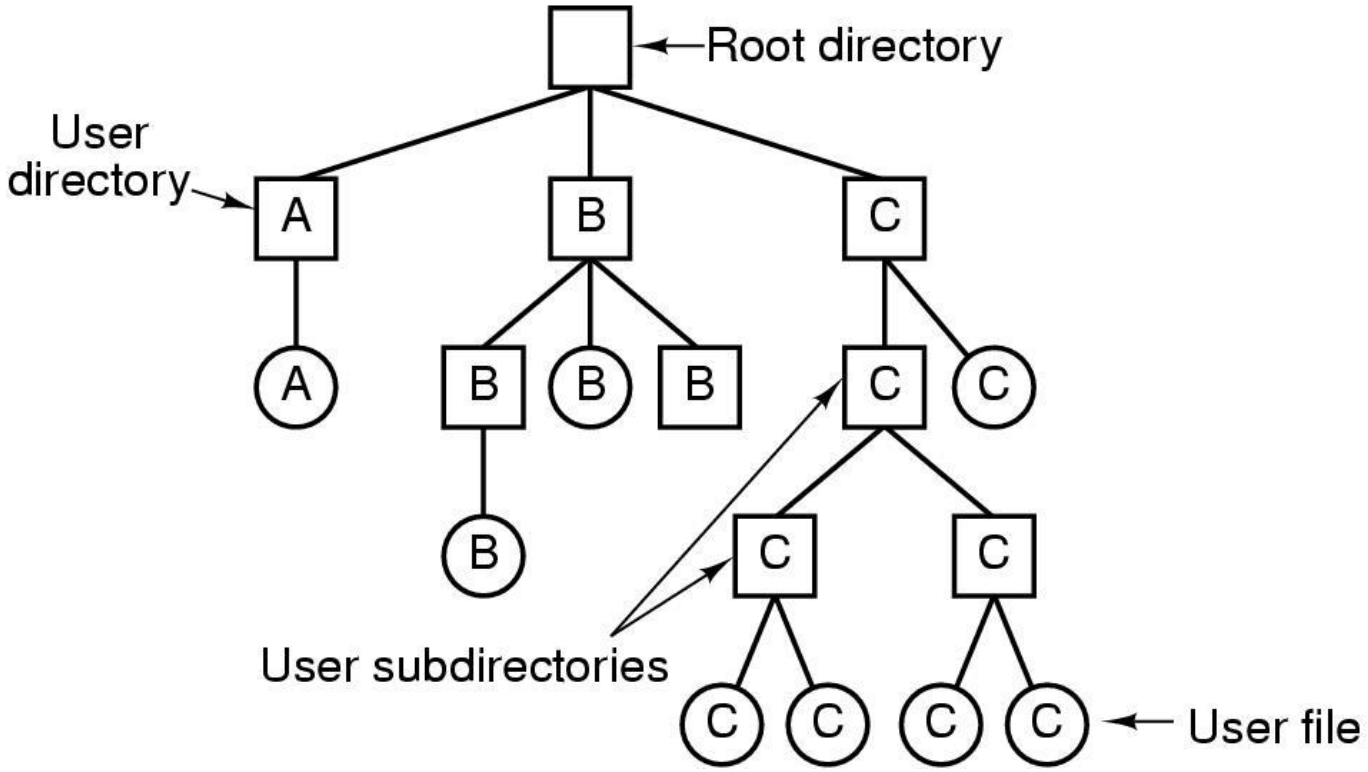
Dört Dosya İçeren Tek Düzeyli Dizin

- .



Hiyerarşik Dizin Sistemleri

- .

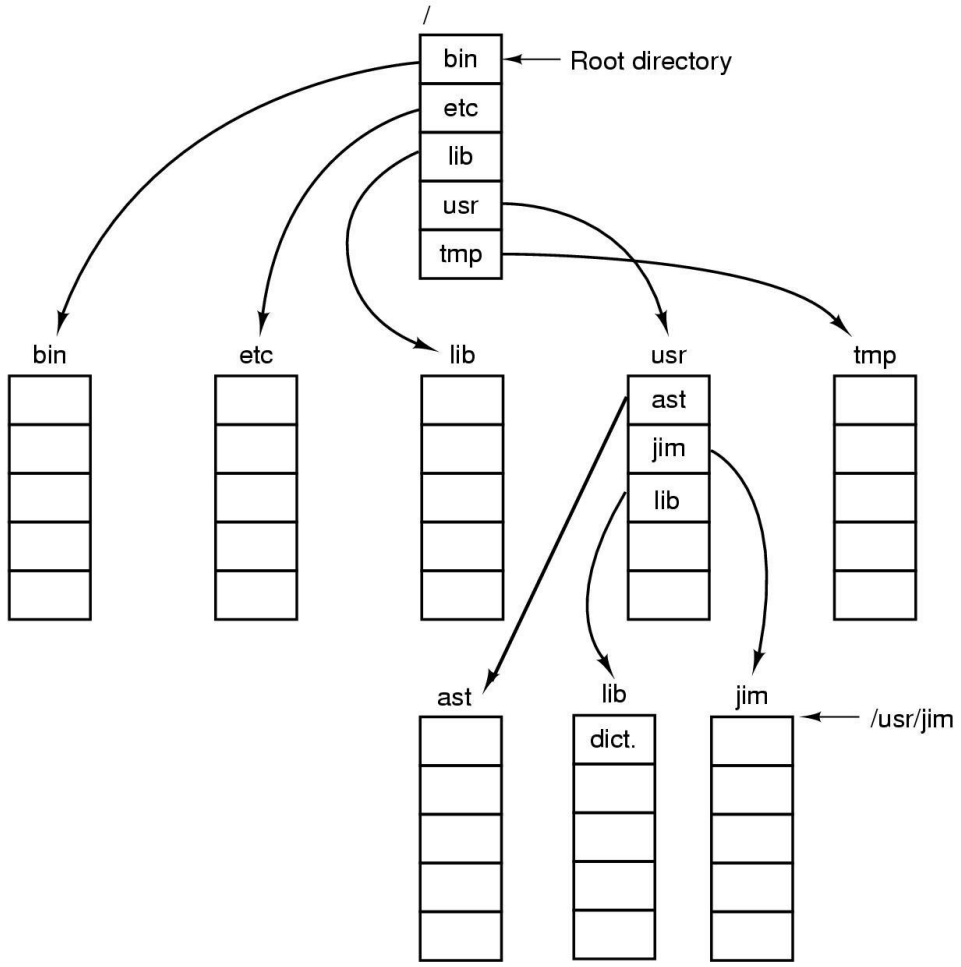


Yol (path) Adları

- 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 Gerçekleme (implementation)

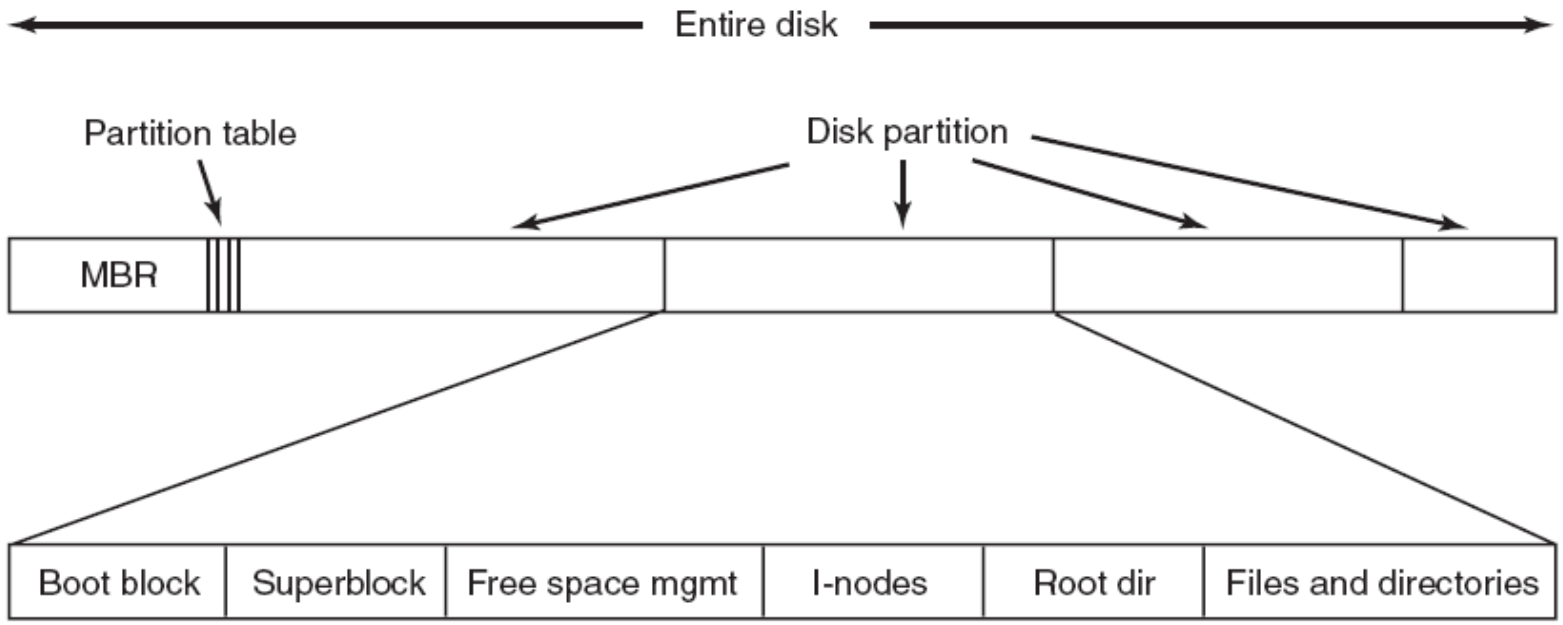
- Dosyalar disklerde saklanır.
- Diskler bir veya daha fazla bölümden (partition) oluşabilir.
- Her bölümde ayrı «dosya sistemi» olabilir
- Diskin 0. sektörü, Ana Önyükleme Kaydıdır (master boot record)
- Bilgisayarın açılışı (boot) için kullanılır
- MBR'nin sonu bölüm tablosuna sahiptir.
- Tabloda her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

Dosya 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)

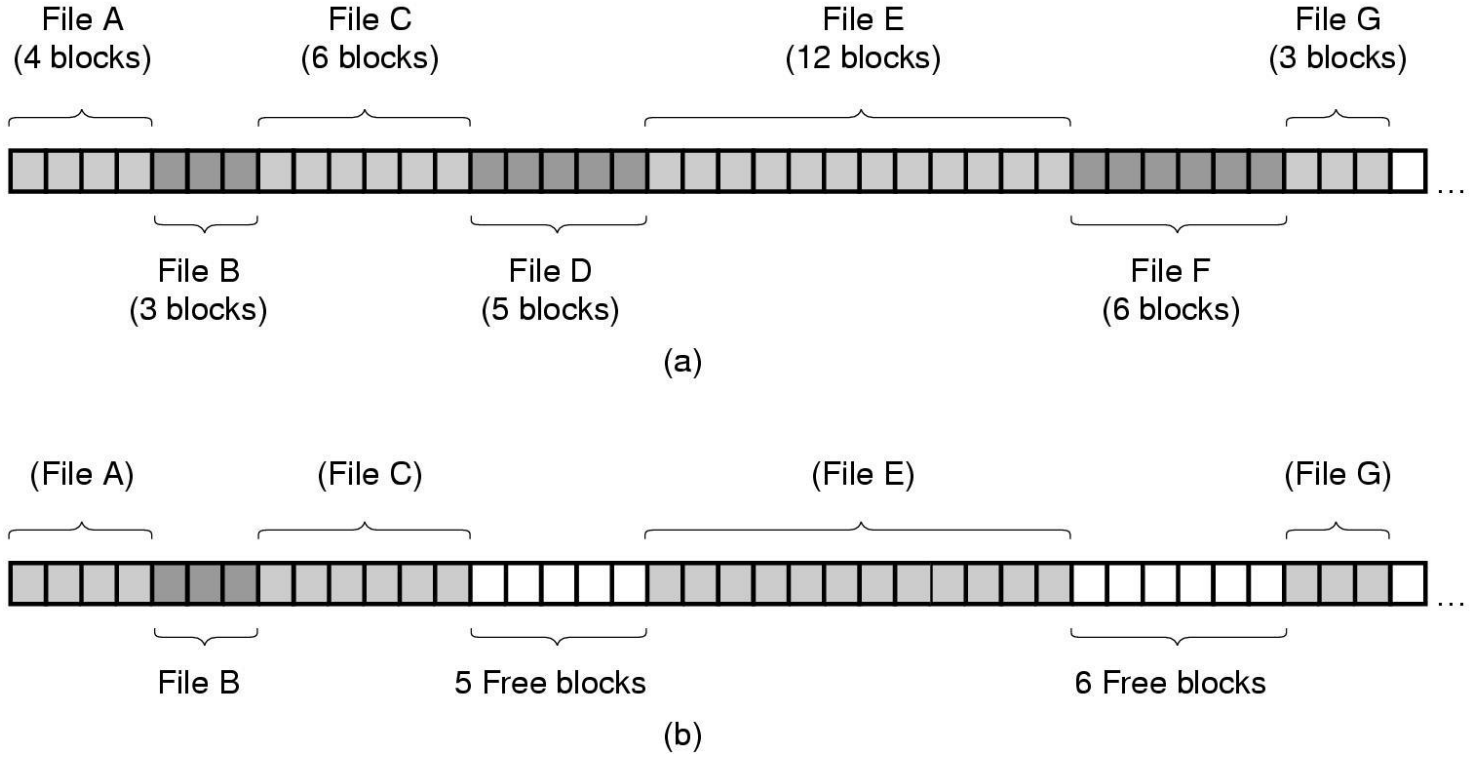
- Süperblock, dosya sistemi hakkında bilgi içerir (fs tipi, blok sayısı..)
- i-nodes dosyalar hakkında bilgi içerir

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

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

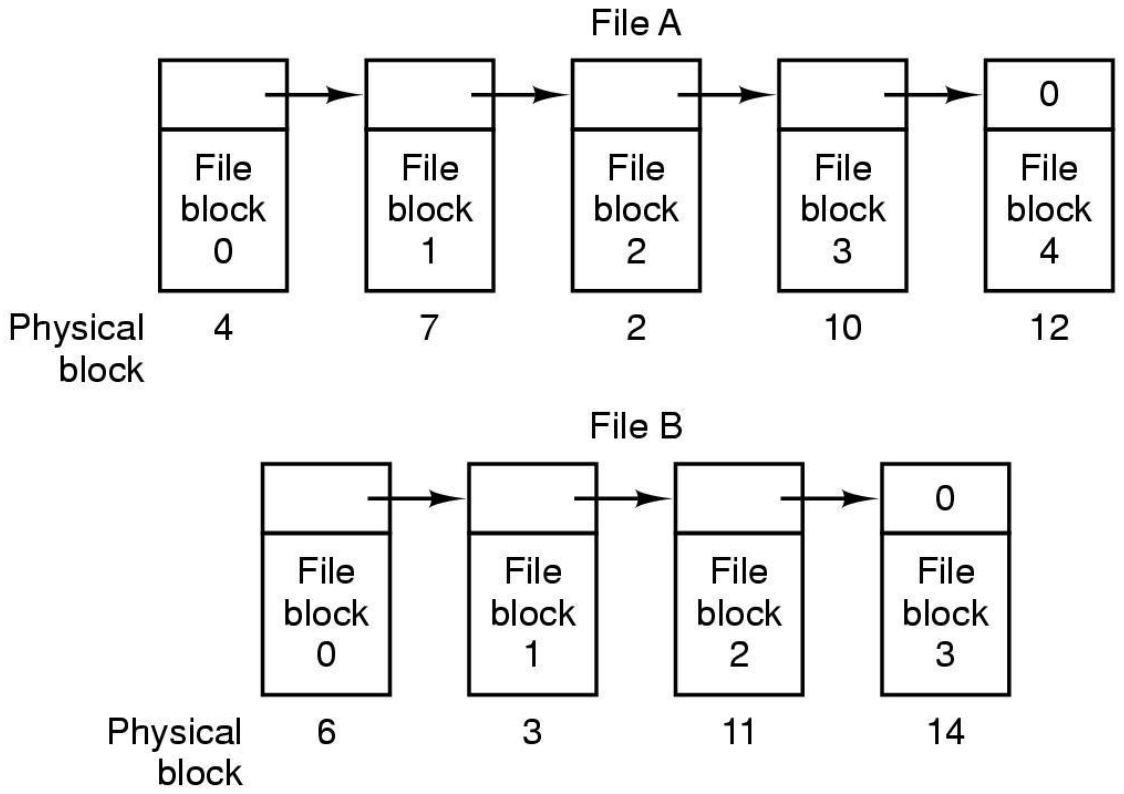


Bitişik Yer Tahsisi

- Uygulaması kolay
- Okuma performansı harika.
- Dosyadaki ilk bloğu bulmak için yalnızca bir arama (seek) yeterli.
- Disk zamanla parçalanır (fragmented)
- CD-ROM'lar, dosya sistemi boyutu sabit olduğu için bitişik yer tahsisi kullanır
- DVD'ler birkaç ardışık 1 GB dosyada saklanır çünkü DVD standardı maksimum 1 GB dosya boyutuna izin verir

Bağlı Liste Yer Tahsisi

- Bir dosyayı, disk bloklarından bağlı liste olarak saklamak.



Bağlı Liste Yer Tahsisi

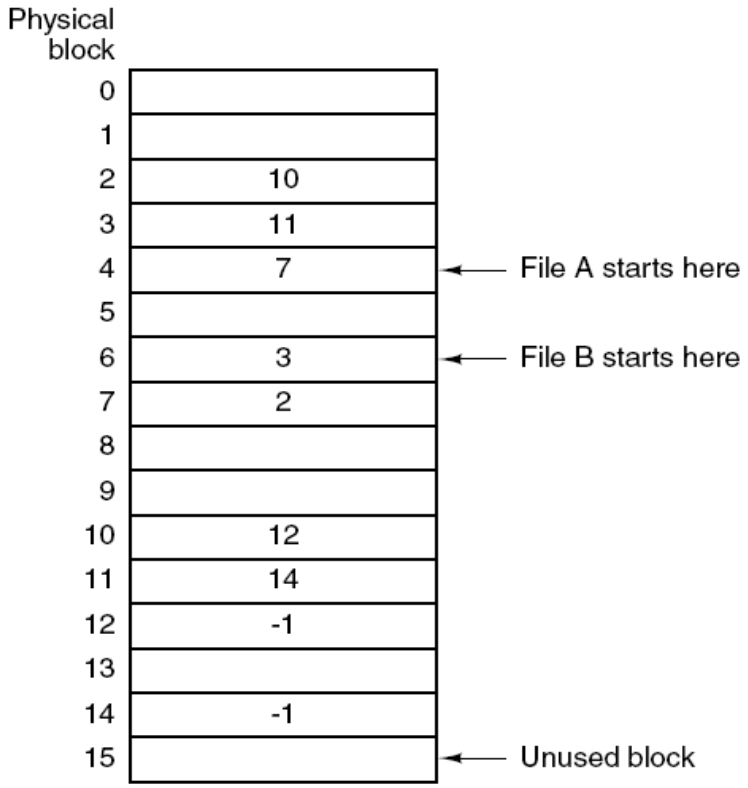
- Her dosyayı disk bloklarının bağlı bir listesi olarak tutar
- Sadece dahili parçalanma olur
- Rastgele erişim yavaş.
- Bir bloğa ulaşmak için işaretçileri takip etmek gerekir
- Kopyalama işleminde işaretçiler ek maliyete neden olur.

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

- İşaretçiler (pointer) bellekte bir tabloda tutulur
- File Allocation Table (FAT)
- Tüm bloğun kullanımını alabilir
- Rastgele erişim kolaydır
- sadece başlangıç bloğunun numarasını saklar
- Tüm tabloyu bellekte tutar!
- Ölçeklenebilir değil!

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

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



Tablo Kullanılarak Bağlı Liste Yer Tahsisi

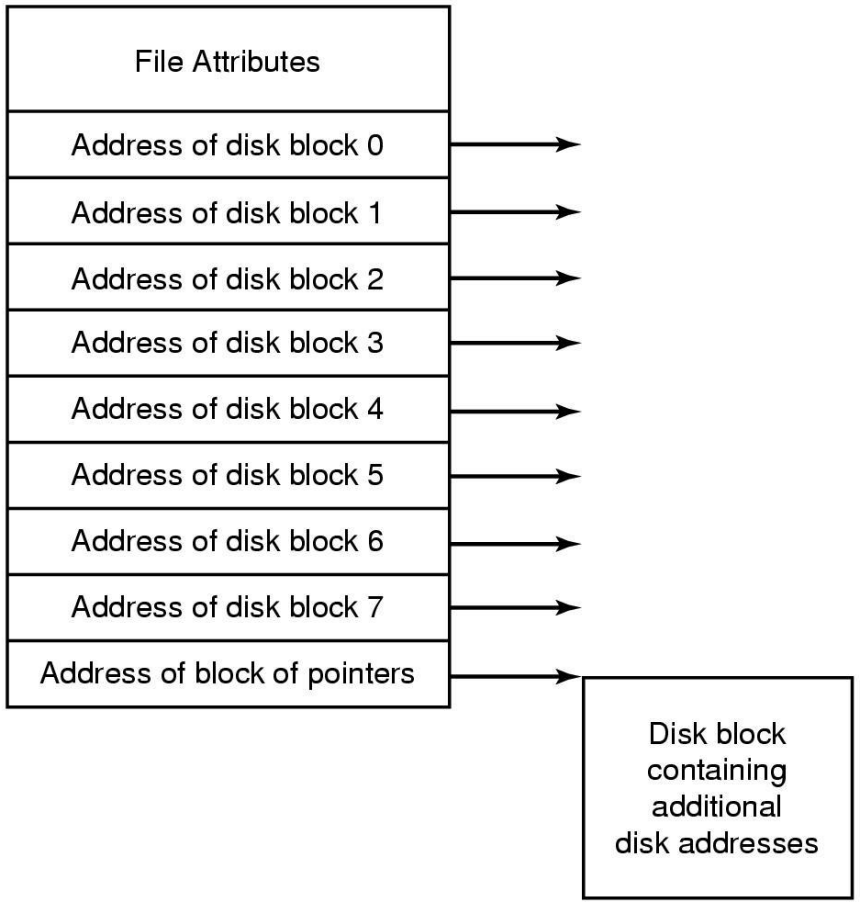
- Tablonun boyutu gerçekten büyük oluyor
- Örneğin, 1 KB bloklu 200 GB disk, 600 MB'lık bir tabloya ihtiyaç duyar
- Tablo boyutunun büyümesi, disk boyutunun büyümesiyle doğru orantılıdır

I-nodes

- Veri yapısını yalnızca açık dosyalar için bellekte tutar
- Veri yapısı, blokların disk 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

- .

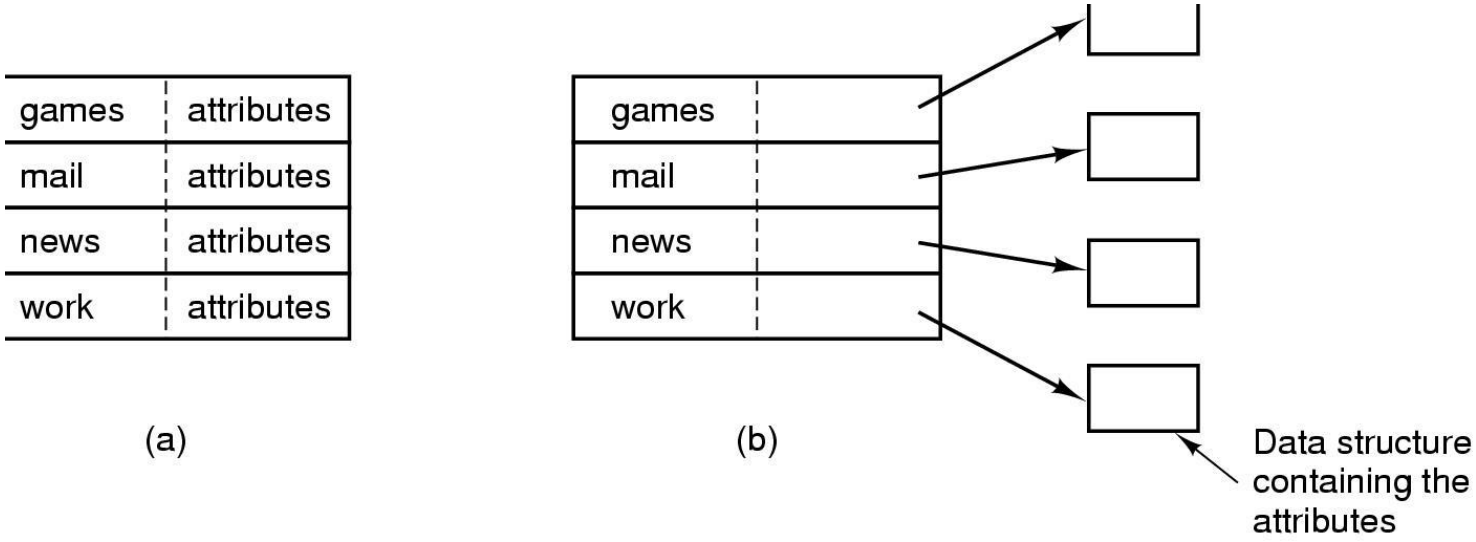

File Attributes
Address of disk block 0
Address of disk block 1
Address of disk block 2
Address of disk block 3
Address of disk block 4
Address of disk block 5
Address of disk block 6
Address of disk block 7
Address of block of pointers
Disk block containing additional disk addresses

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)

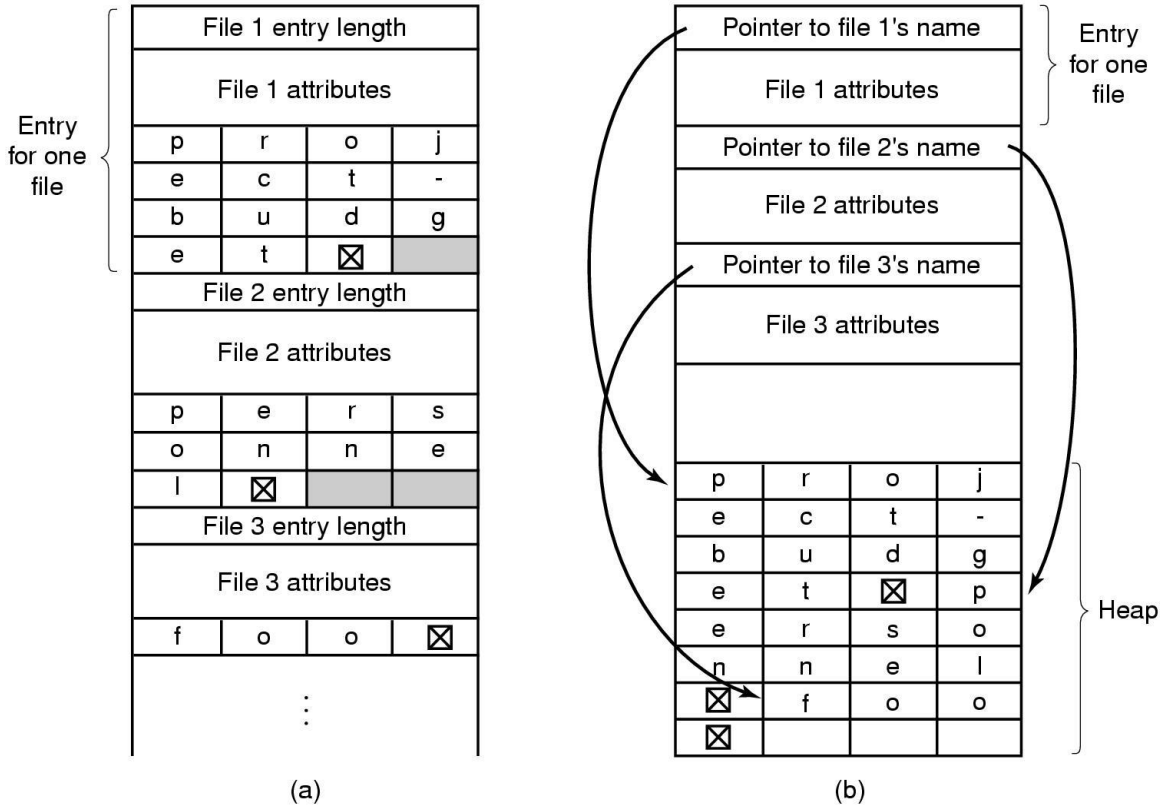


Dizinler

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

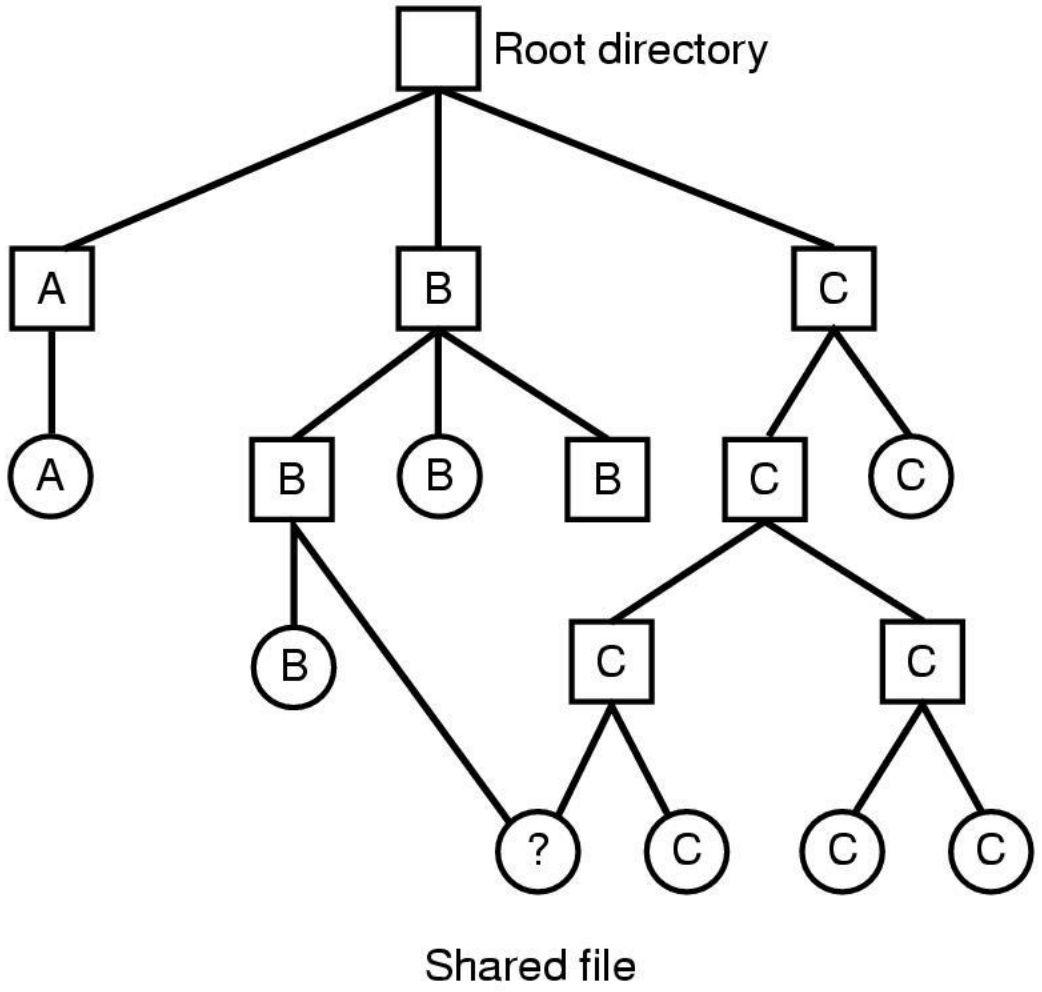
Dizinler

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



Paylaşımlı Dosyalar

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



Shared file

Paylaşımlı Dosyalar

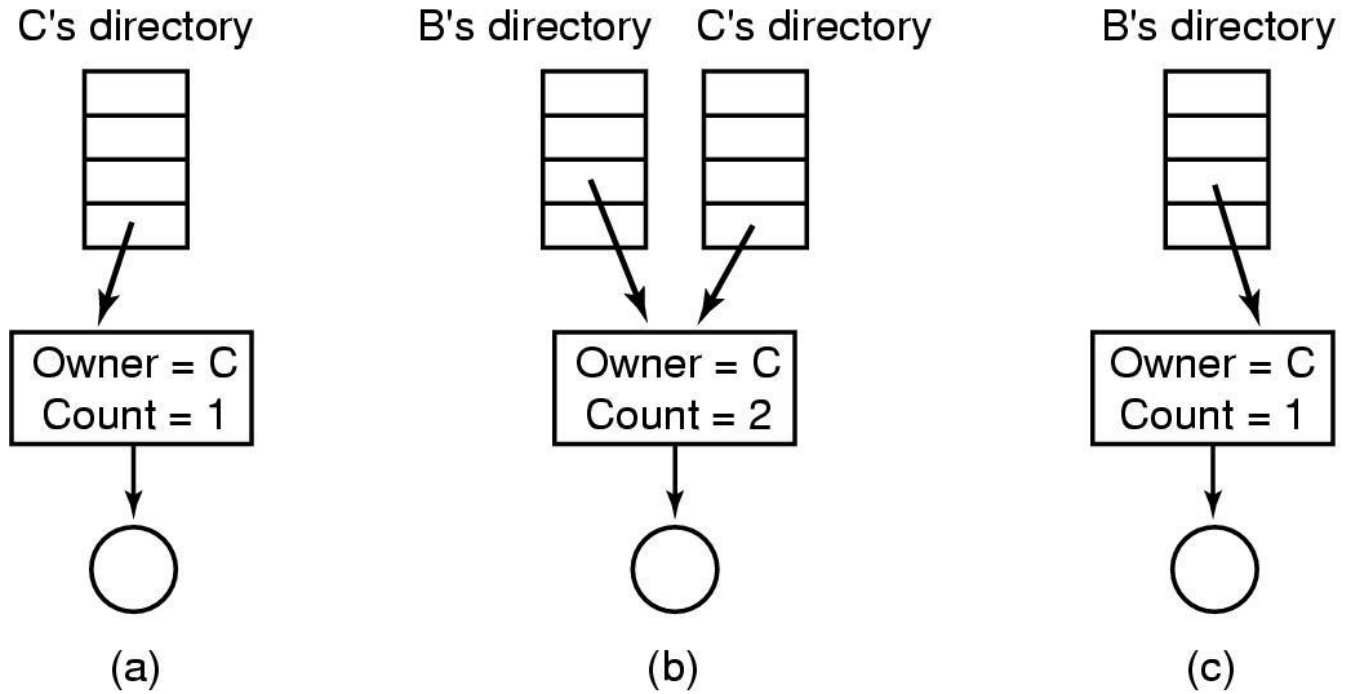
- B veya C yeni bloklar eklerse, diğer sahip nasıl öğrenir?
- Paylaşılan dosyalar için özel i-node kullan - dosyanın paylaşıldığını gösterir
- Sembolik bağlantı (symbolic link) kullanın - sahibi C ise, B'nin dizinine konulan özel bir dosya. Bağlı (linked) olduğu dosyanın yol adını içerir

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

- CPU daha hızlı, diskler ve bellekler daha büyük ancak disk arama süresi azalmadı
- Daha büyük önbellekler - önbellekten okuma yapabilir
- Diskteki verilerin güncellenmesi gerektiğinden yazma işlemleri optimize edilmeli
- Disk log-collect olarak yapılandırılır ve logları periyodik olarak diskteki bir segmente gönderir. Yazma işlemleri çok küçük olma eğilimindedir
- Segment, içerik özetine sahiptir (i-nodes, dizinler….).
- i-node haritası diskte tutulur ve i-node'ları bulmak için bellekte önbelleğe alınır

Günlük (log) Yapılandırılmış Dosya Sistemi

- Temizleyici iş parçacığı günlüğü sıkıştırır.
- Segmenti mevcut i-düğümler için tarar, kullanılmayanları atar ve mevcut olanları belleğe gönderir.
- Yazıcı iş parçacığı, mevcut olanları yeni segmente yazar.
- Unix'te iyi çalışır.
- Çoğu dosya sistemiyle uyumlu değil
- Kullanılmıyor

Günlük (journaling) Dosya Sistemleri

- Çökmeler olduğunda kaybolan dosyalara karşı korunmak gerek
- 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

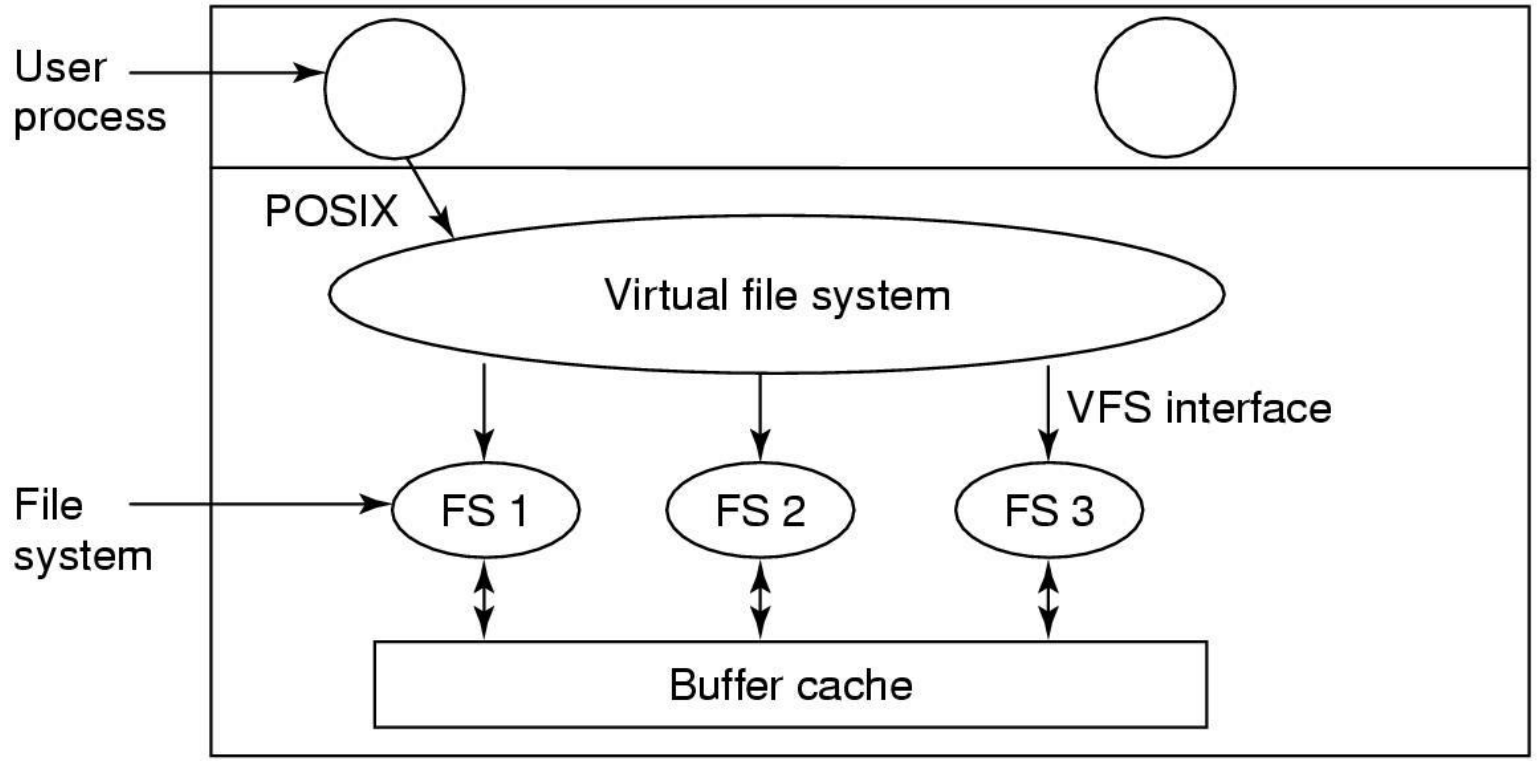
- Eylemleri gerçekleştirmeden önce bir günlük tut, günlüğü diske yaz ve ardından eylemleri gerçekleştir.
- Bir kazadan kurtulabilir mi!
- Eylemler eşgüçlü (idempotent) olmalı. Bunu yapmak için veri yapıları düzenlenmeli
- Blok n'yi serbest olarak işaretle, idempotent bir işlemdir.
- Bir listenin sonuna serbest bırakılmış bloklar eklemek idempotent değildir
- NTFS (Windows) ve Linux günlük kaydı kullanır

Sanal Dosya Sistemleri

- Aynı makinede birden fazla dosya sistemi var
- 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

- .

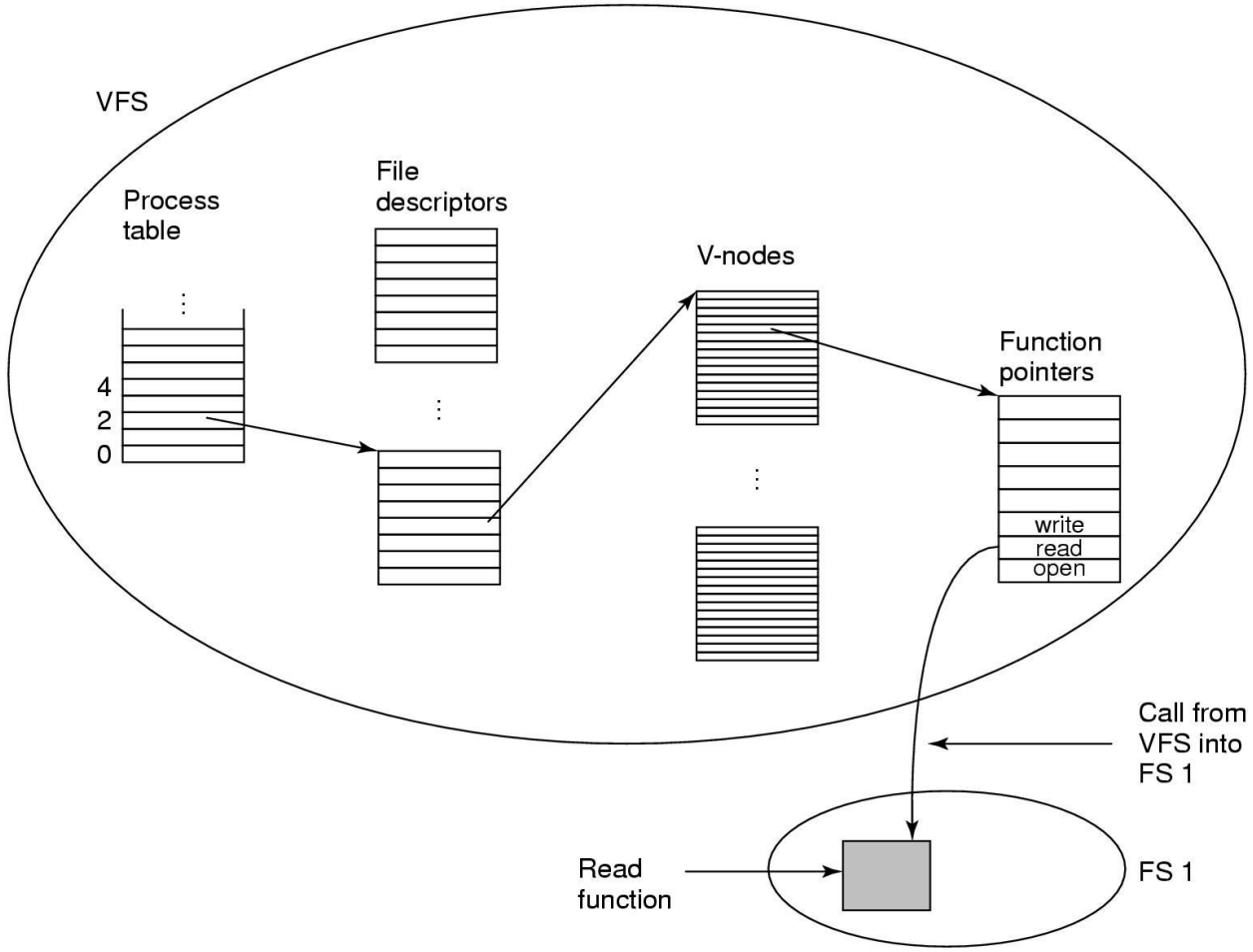


Sanal Dosya Sistemi Nasıl Çalışır

- 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

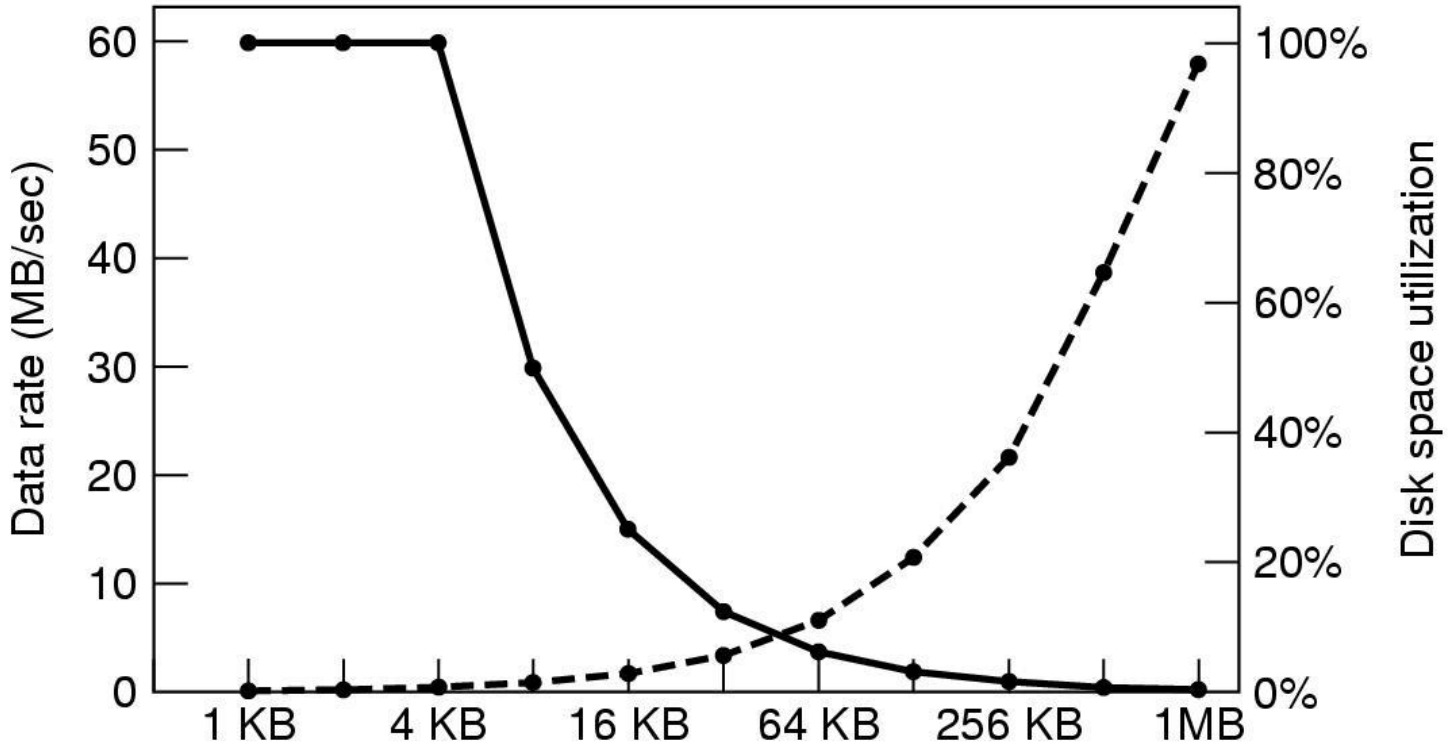
- Verilen boyuttan daha küçük olan dosyaların yüzdesi

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.

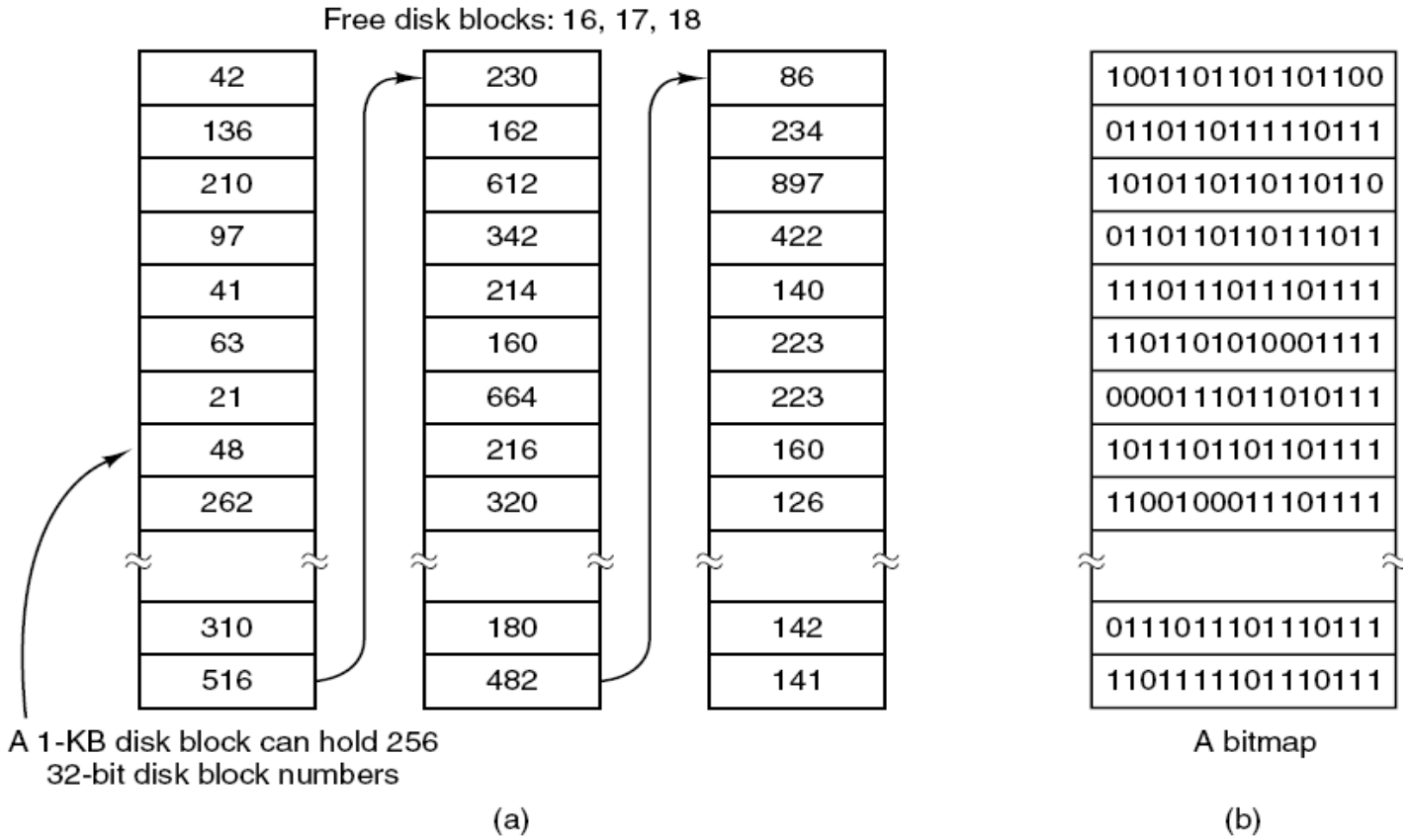


Disk Alanı Yönetimi

- 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

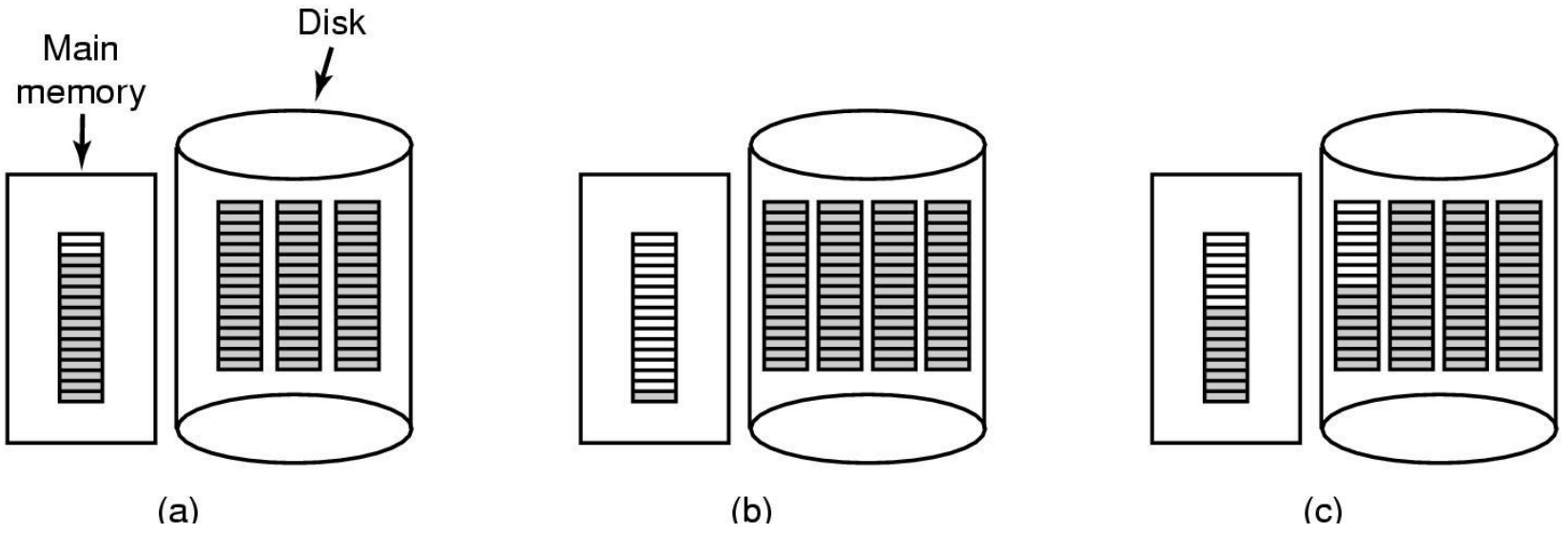


Boş Blokların İzini Tutma

- 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.

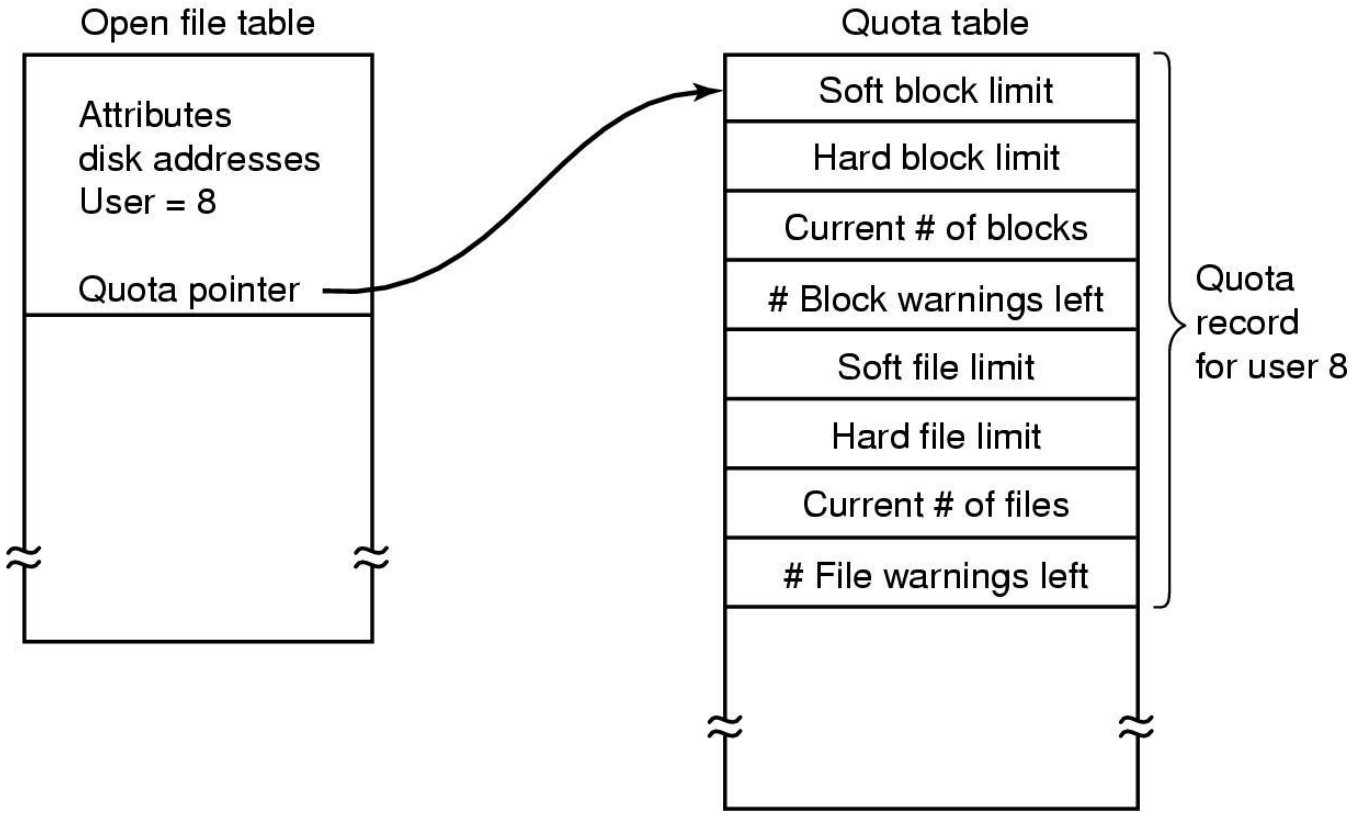


Disk Kotaları

- Açık dosyalar tablosundaki giriş (entry), kota tablosuna işaret eder
- Her açık dosya için bir giriş
- 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

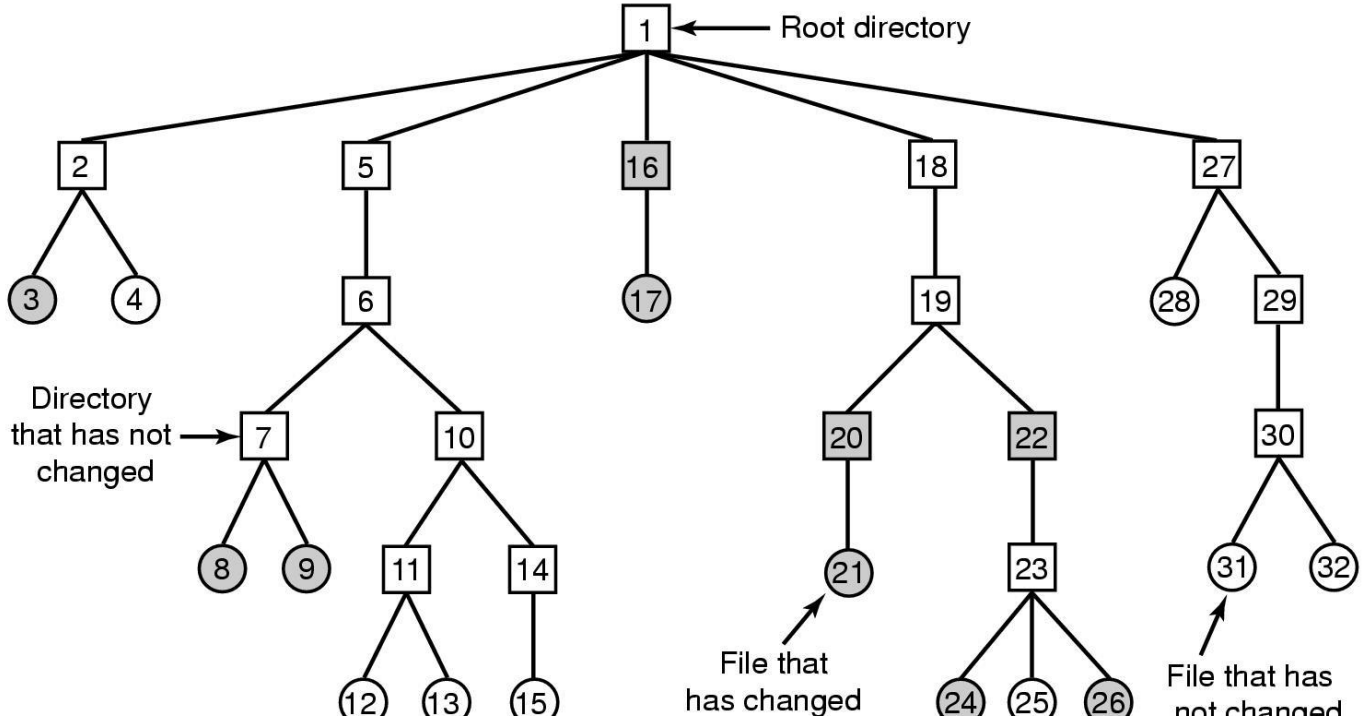
- uygulaması kolay
- Belirli bir dizini atlayamaz (skip)
- Kademeli dökümler yapamaz (incremental)
- dosyalar tek tek geri yüklenemez
- Mantıksal döküm stratejisi daha yaygın kullanılır

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.

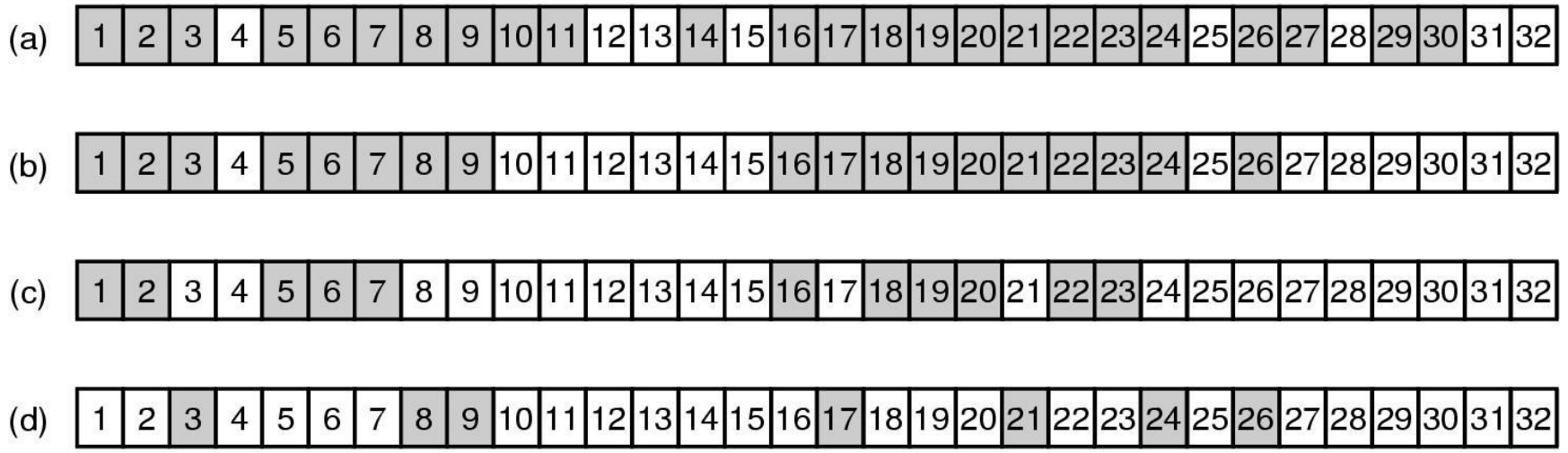


Mantıksal Döküm Algoritması

- 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

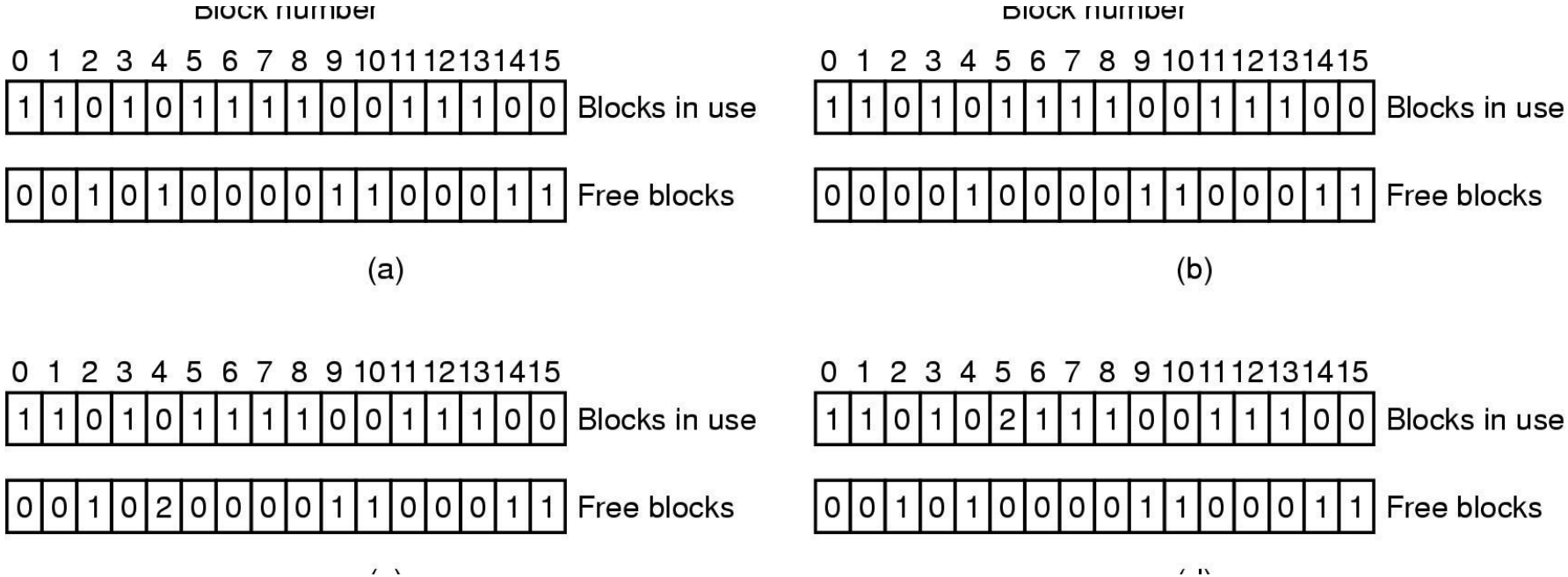
- Diskte boş fs ile başlanır
- Son tam döküm geri yüklenir.
- Önce dizinler, sonra dosyalar.
- Ardından kademeli olarak dökümler geri yüklenir (kaset, teyp üzerinde sıralıdırlar)

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

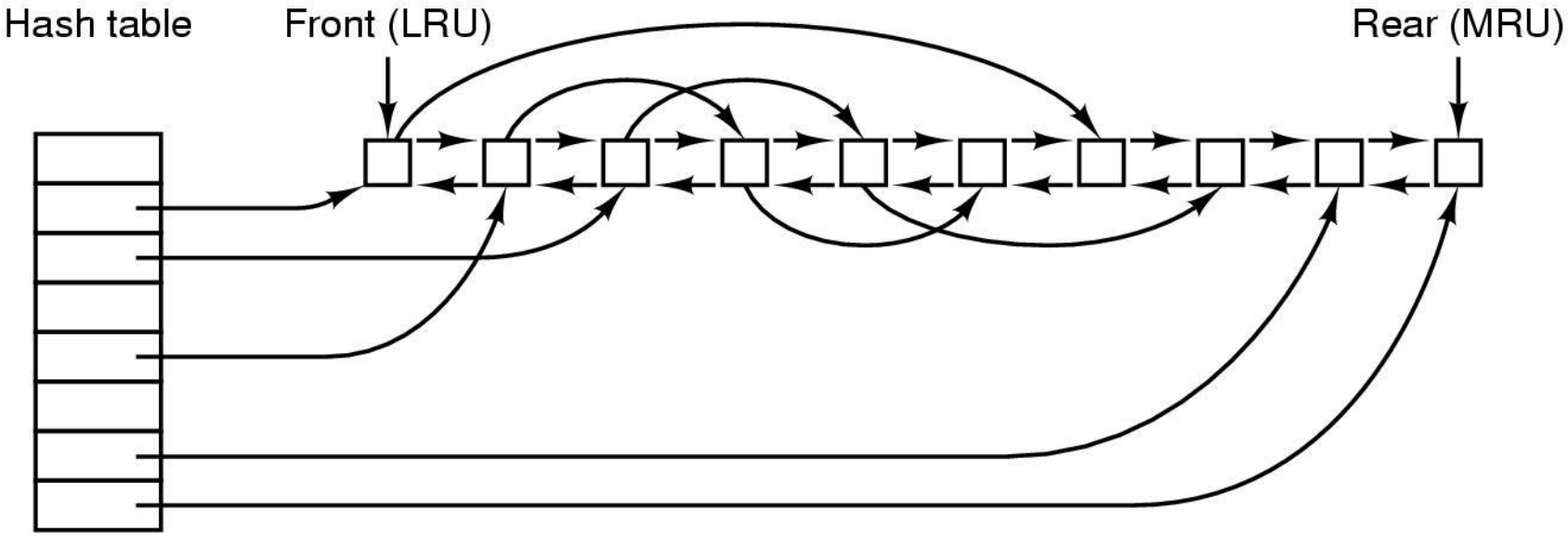
- Bloklar yerine dosyalara bak
- Dosya başına bir sayaç tablosu kullan
- Kök dizinde başla, aşağı in, dosya bir dizinde her göründüğünde sayacı artır
- Sayaçları i-node'lardan gelen bağlantı (link) sayılarıyla karşılaştırır.
  - Tutarlı olmak için aynı olmak zorunda

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)

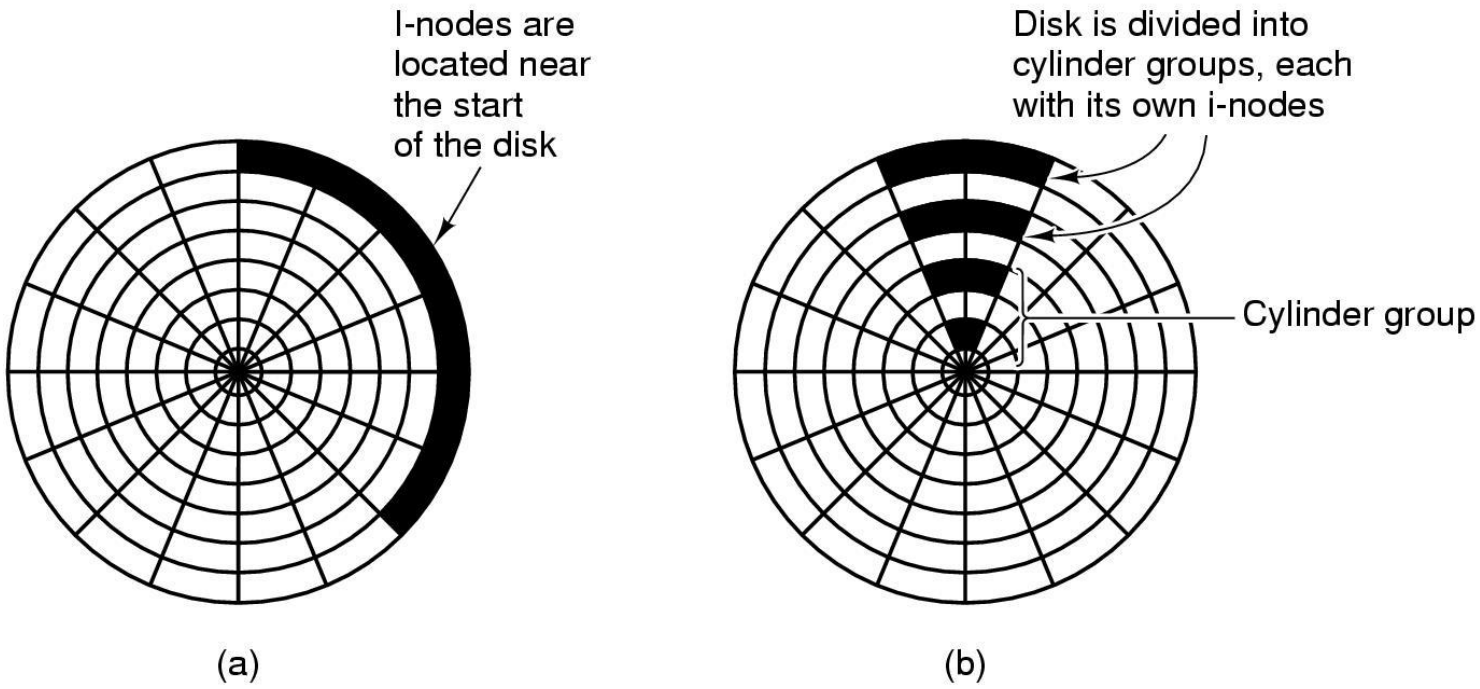
- Önbelleğe almak için k bloğu okunduğunda, k+1 bloğu önbellekte değilse onu da oku
- Yalnızca sıralı dosyalar için geçerlidir
- Dosyanın sıralı mı yoksa rastgele mi olduğunu belirlemek için bir bit kullanılır. Bir arama yap, bitin değerini çevir (flip).

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.

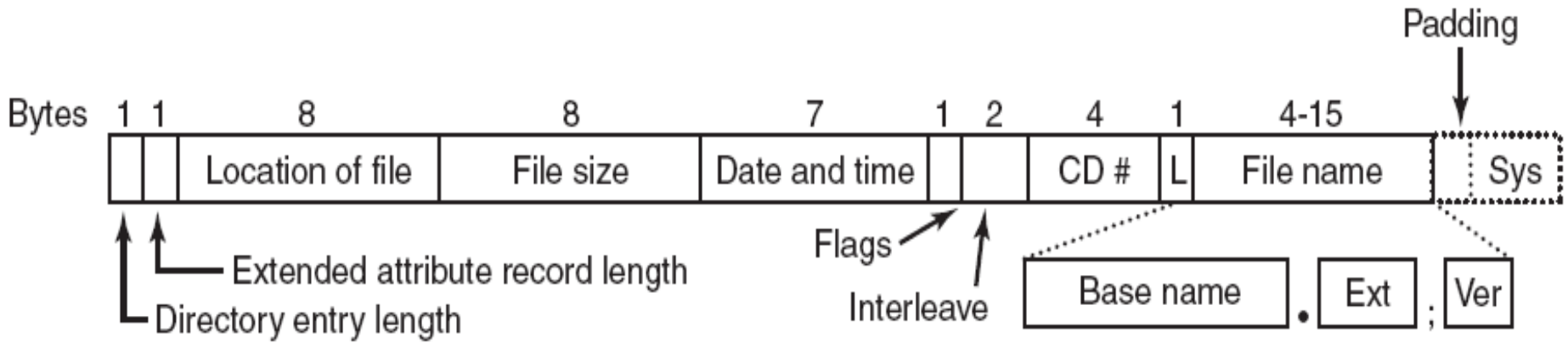


Diskleri Birleştirme (defrag)

- 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 Protocol (RRIP)

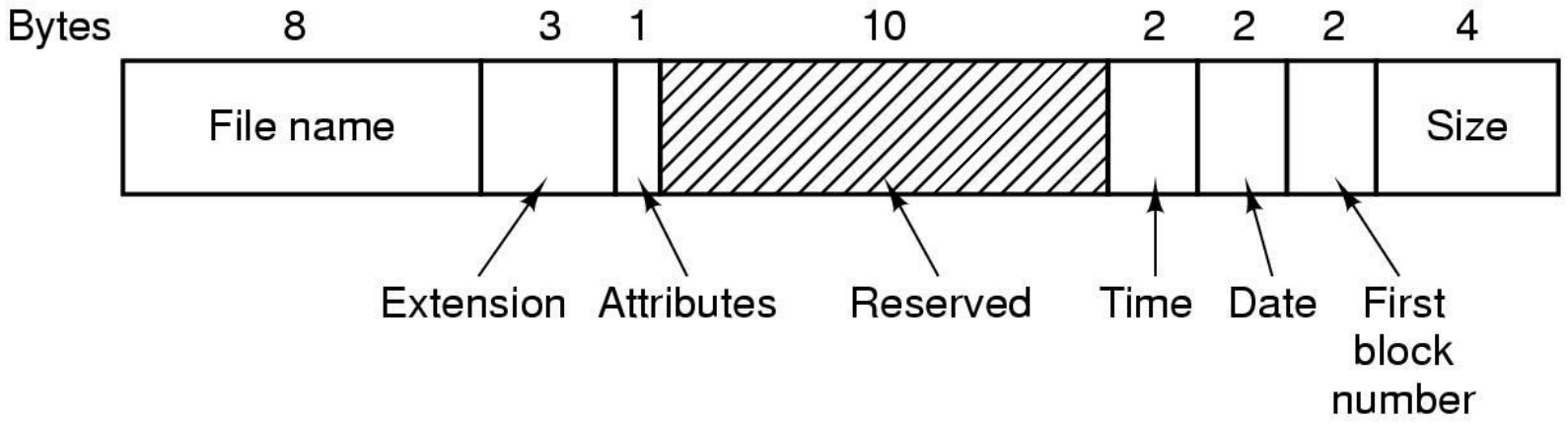
- PX - POSIX attributes. POSIX öznitelikleri
- PN - Major ve minor cihaz numaraları
- SL - Symbolic link. Sembolik bağ
- NM - Alternative name. Seçenek adı
- CL - Child location. Çocuk konumu
- PL - Parent location. Ebeveyn konumu
- RE - Relocation. Yer değiştirme
- TF - Time stamps. Zaman damgaları

Joliet Uzantı Alanları

- Uzun dosya adları.
- Unicode karakter seti.
- Dizin, sekiz seviyeden daha derine inebilir.
- Uzantıları olan dizin adları

MS-DOS Dosya Sistemi – Dizin Girdisi

- .



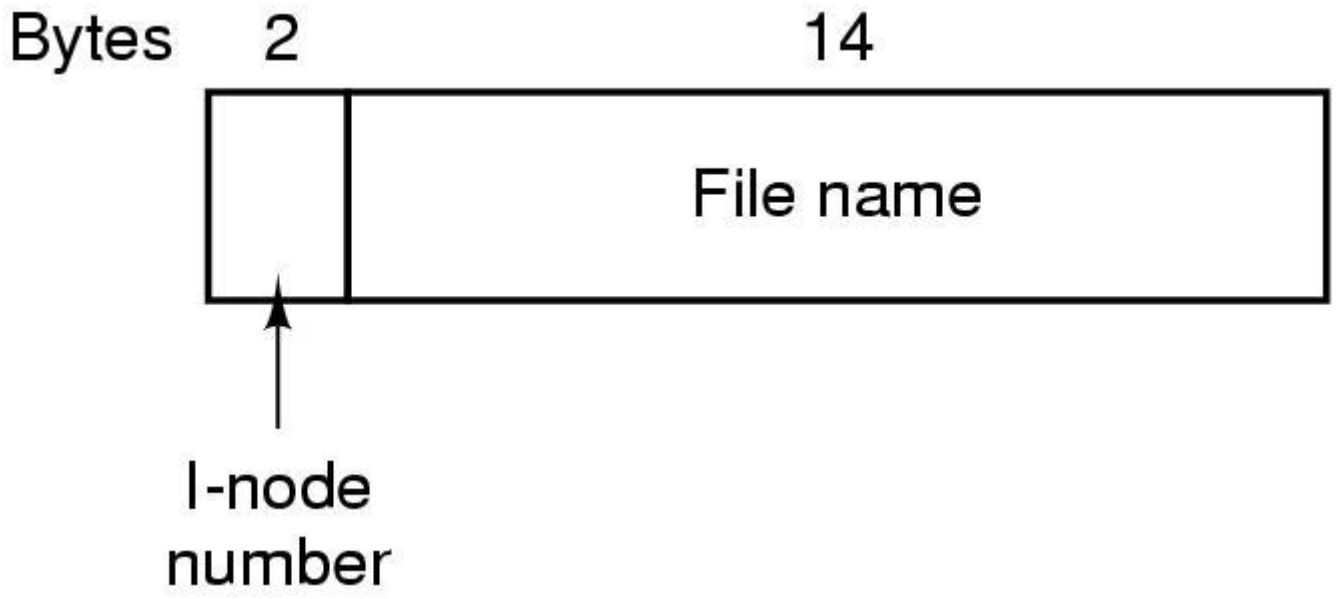
Blok Boyutları için Maks Bölüm Boyutu

- .

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

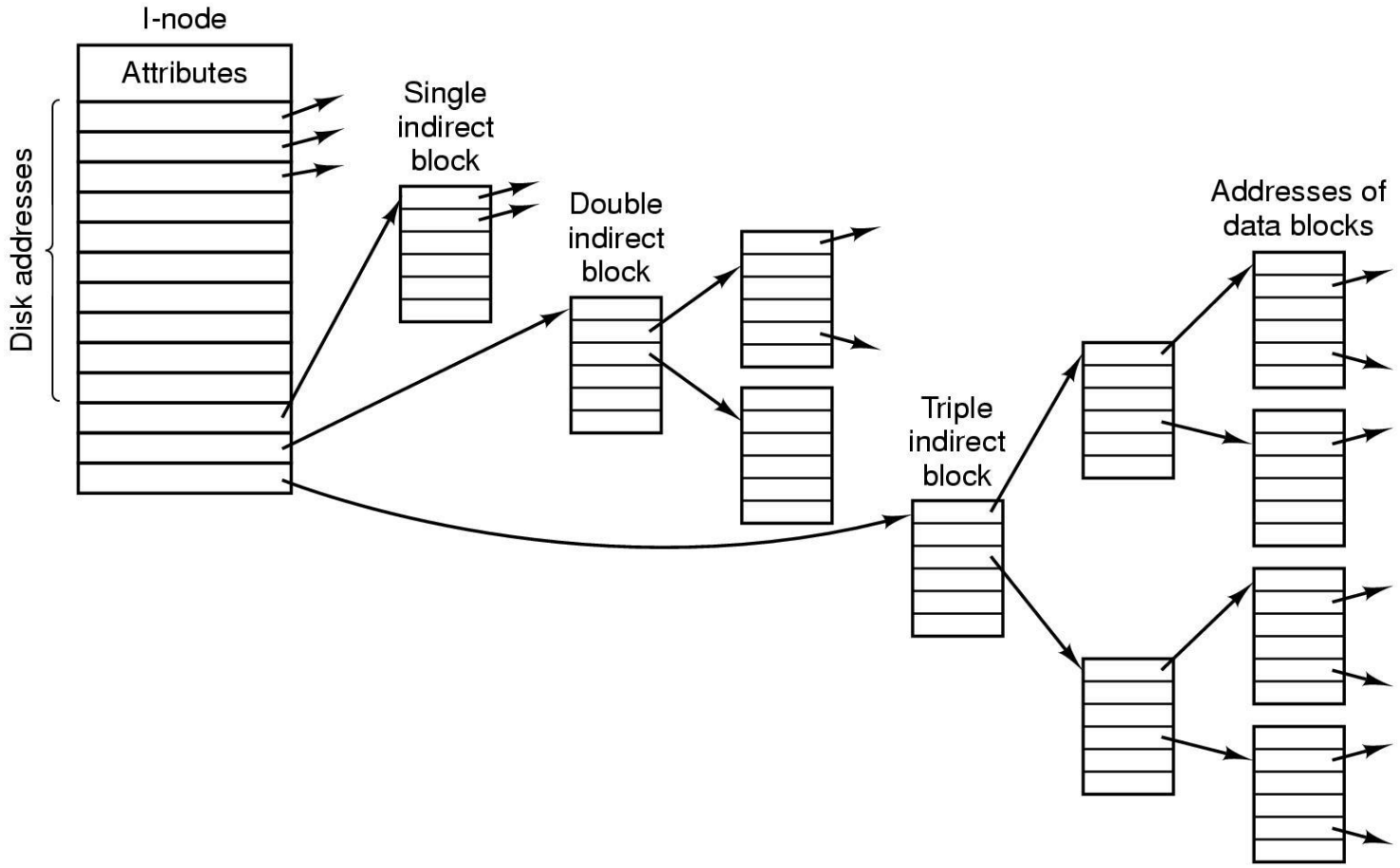
UNIX V7 – Dizin Girdisi

- .



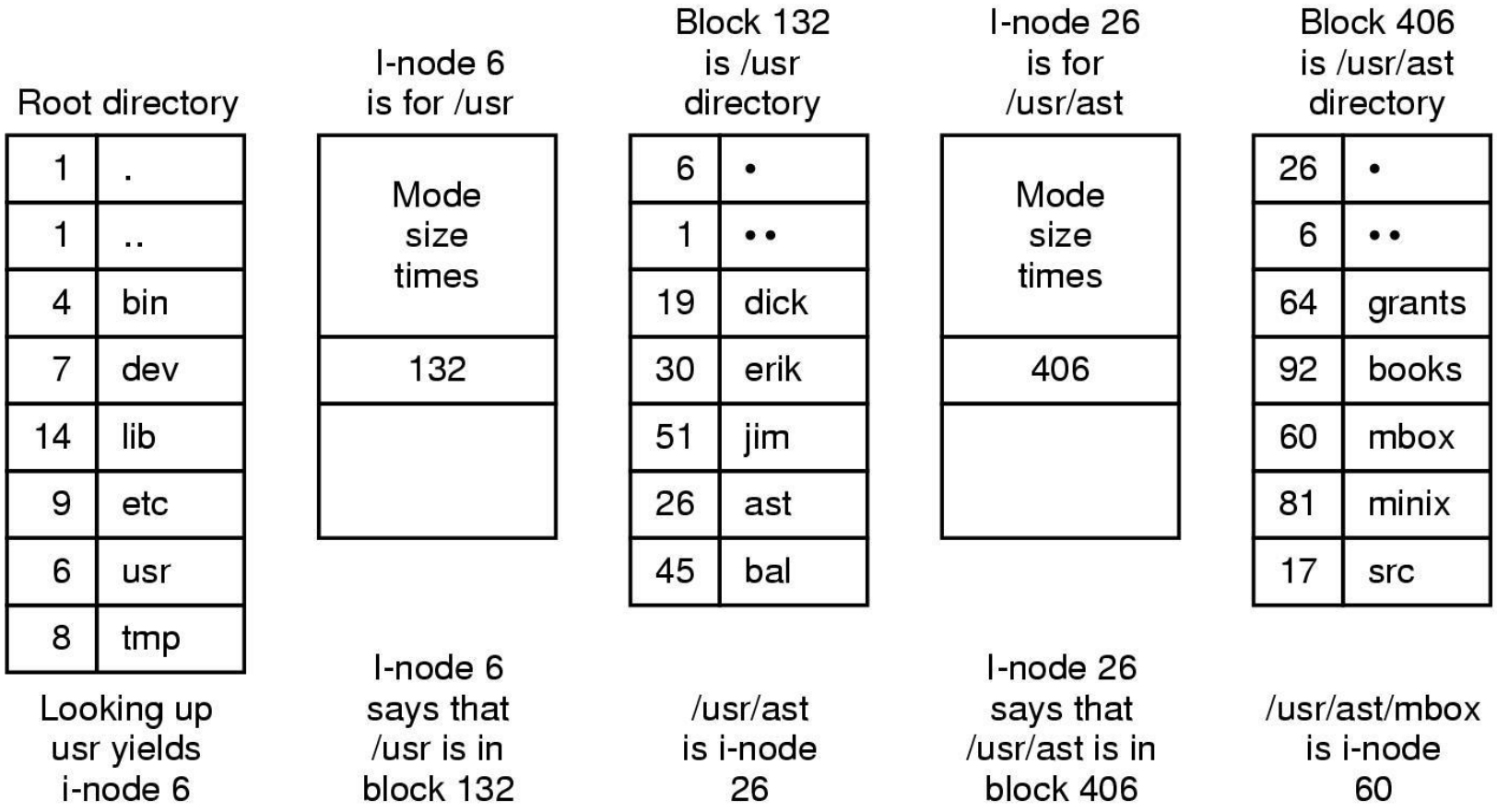
UNIX i-node

- .



/usr/ast/mbox.conf arama adımları

- .



SON