

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

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

- .

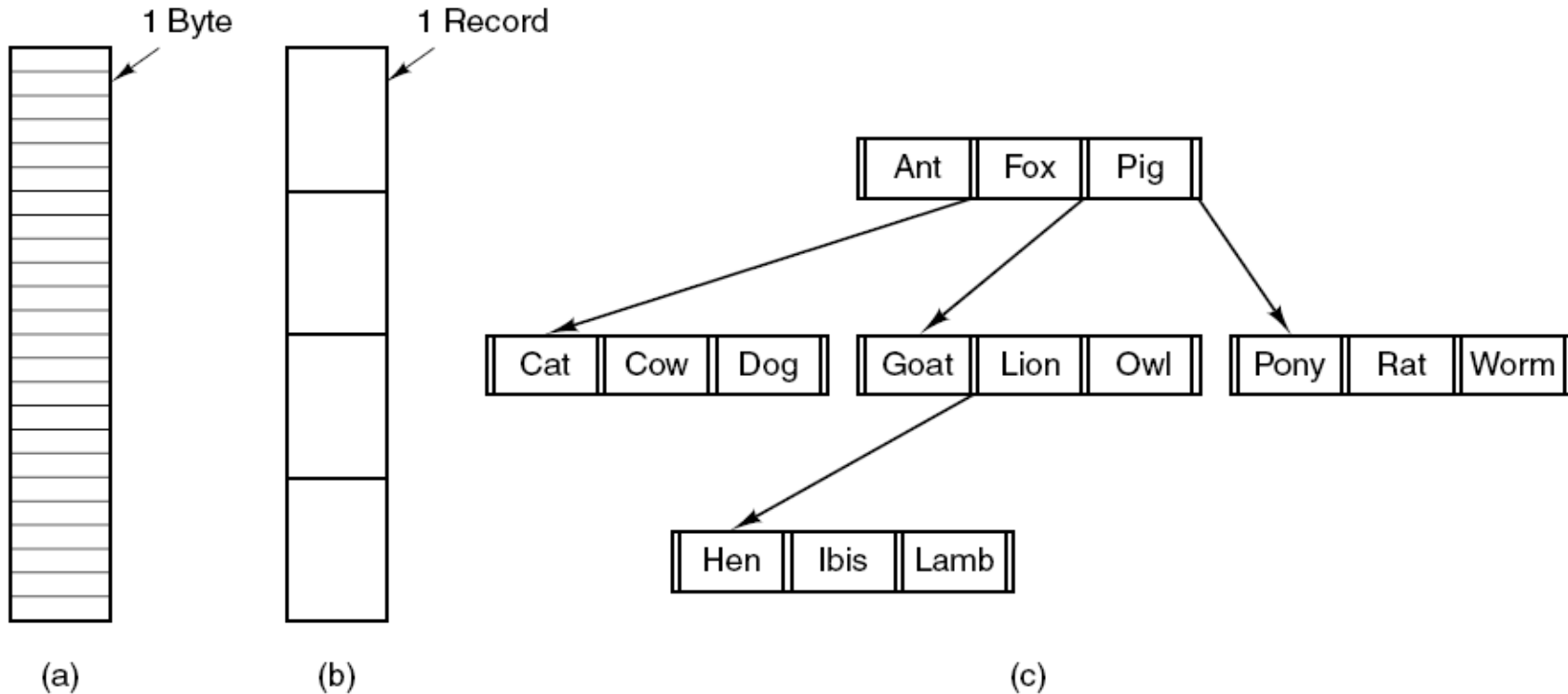
Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
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
- Karakter özel dosyaları – seri (serial) model G/Ç cihazları (yazıcı)
- Blok özel dosyaları – blok tabanlı modeller (disk)

Normal (regular) Dosyalar

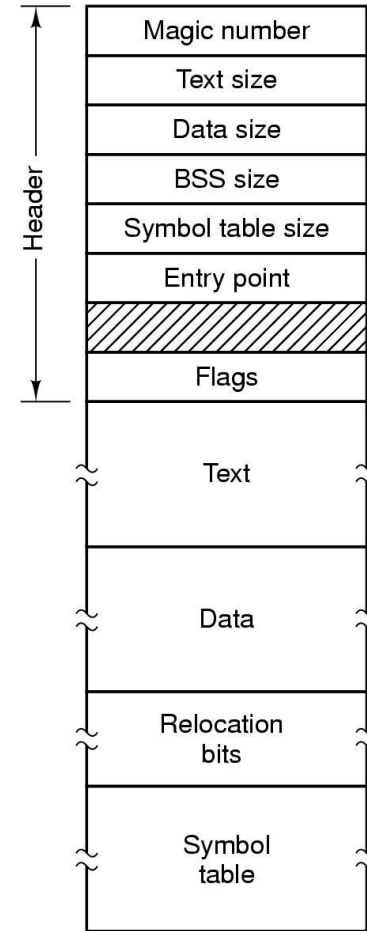
- ASCII veya ikili (binary)
- ASCII
 - Yazdırılabilir
 - Programları bağlamak için boru hattı (pipe) kullanılabilir (ASCII üretiliyor/tüketiyorsa)

İkili Dosya Tipleri

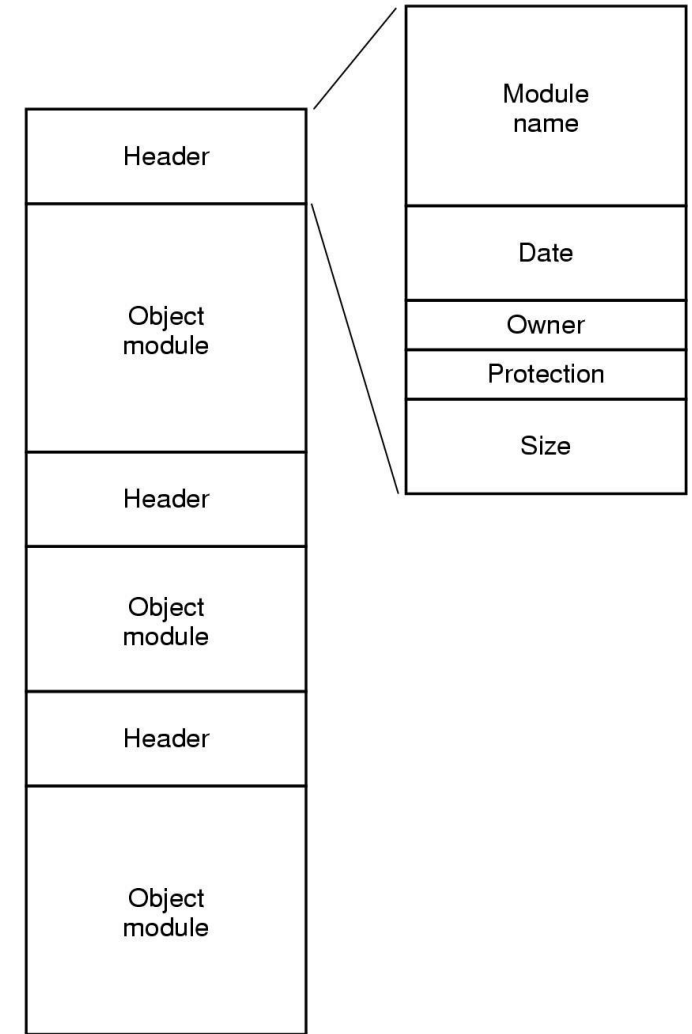
- İki Unix örneği
 - Yürütülebilir (magical field, dosyayı yürütülebilir olarak tanımlar)
 - Arşiv olarak derlenmiş, bağlı (linked) kütüphane prosedürleri hariç
- Her işletim sistemi kendi yürütülebilir dosyasını tanımalıdır

İkili Dosya Tipleri

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



(a)



(b)

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 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

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

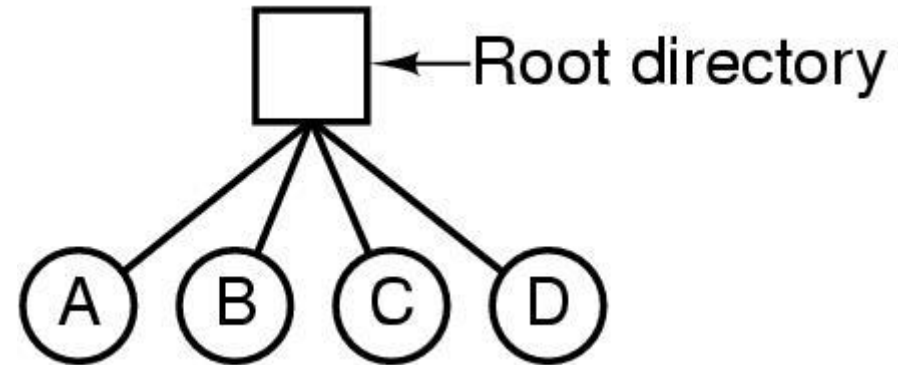
- abc dosyasını xyz'ye kopyalar
- Eğer xyz varsa üzerine yazılır
- Yok ise yaratılır
- Sistem çağrıları kullanılır (okuma, yazma)
- 4K boyutunda parçalar halinde okur ve yazar
- abc dosyasından bir tampon belleğe oku (read sistem çağrısı)
- Tampondan xyz dosyasına yaz (write sistem çağrısı)

Dizinler

- Bir dosya koleksiyonunu düzenlemek için kullanılan dosyalar
- Bazı işletim sistemlerinde klasörler (folder) olarak da adlandırılır

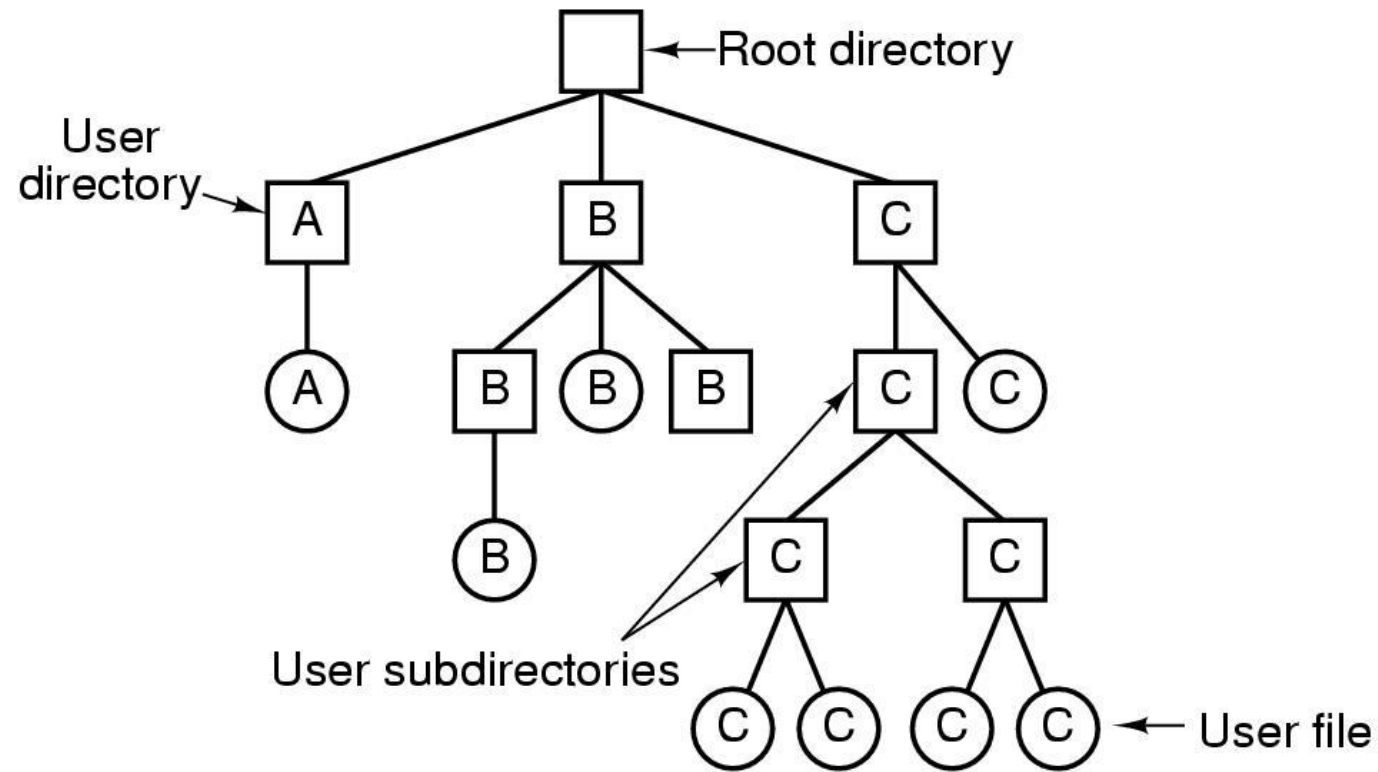
Dört Dosya İçeren Tek Düzeyli Dizin

- .



Hiyerarşik Dizin Sistemleri

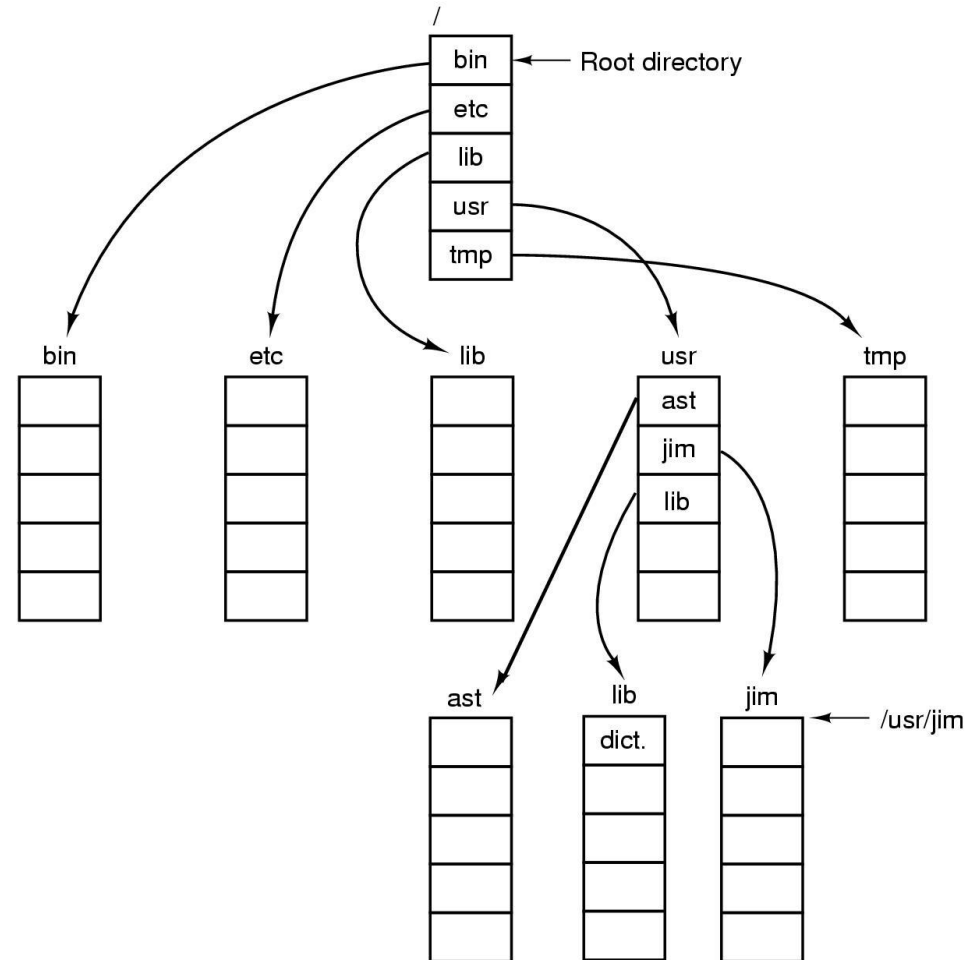
• .



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

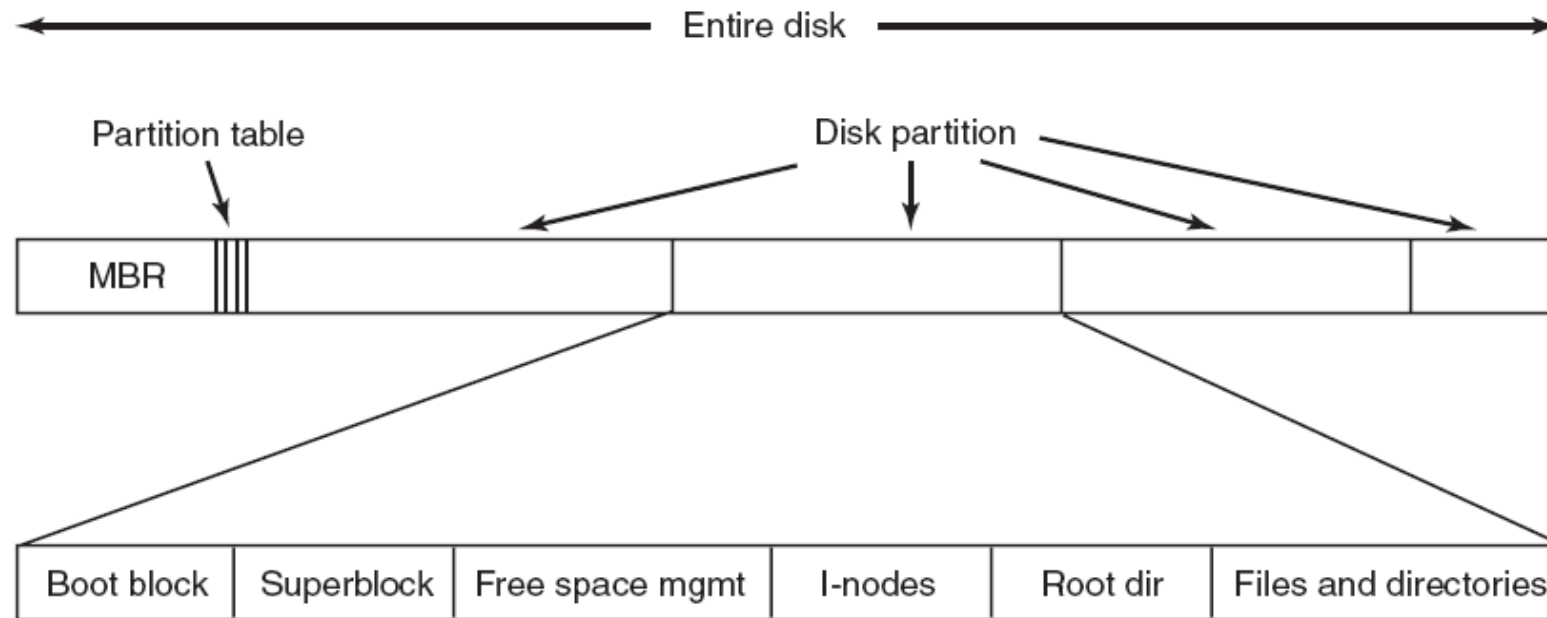
Dosya Gerçekleme (implementation)

- 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)

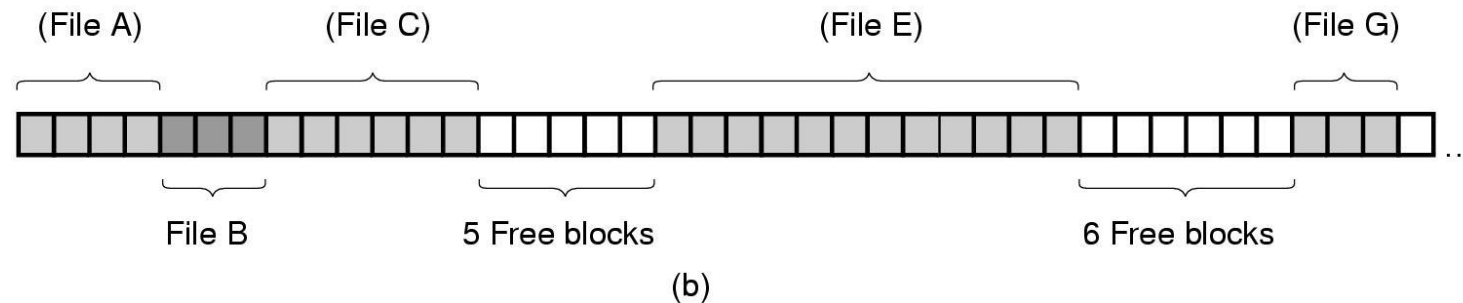
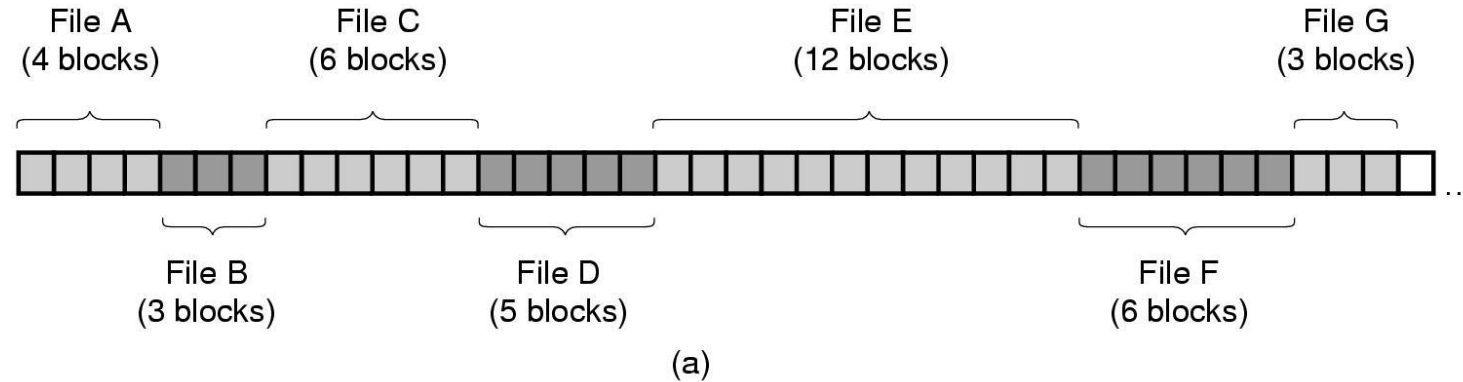
- Süperblock, dosya sistemi hakkında bilgi içerir (fs tipi, blok sayısı..)
- i-nodes dosyalar hakkında bilgi içerir

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.

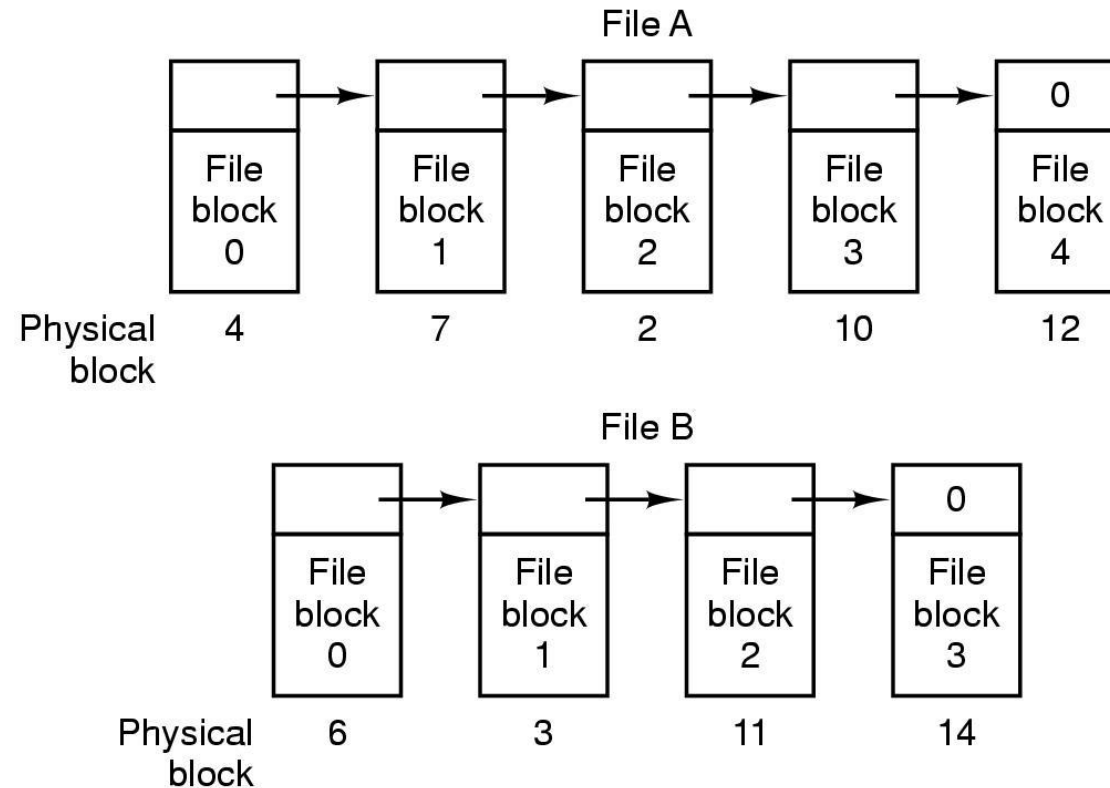


Bitişik Yer Tahsisi

- 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

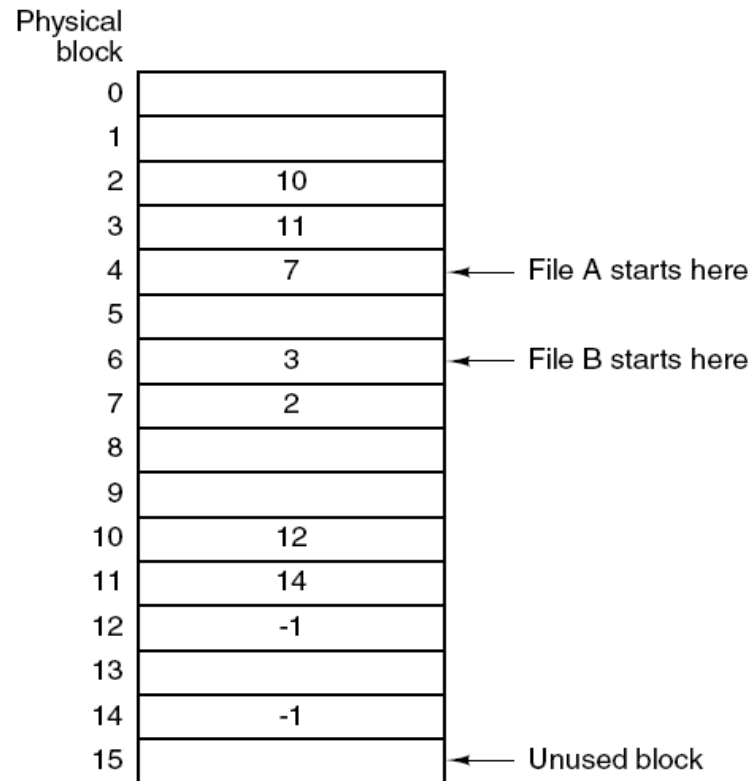
- İyi
- Parçalanma sorunu olmaz
- Kötü
- Rastgele erişim yavaş.
- Bir bloğa ulaşmak için işaretçileri takip etmek gerekir

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

- İşaretçiler (pointer) bellekte bir tabloda tutulur
- File Allocation Table (FAT)
- Windows

Tablo Kullanılarak Bağlı Liste Yer Tahsisi

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



Tablo Kullanılarak Bağlı Liste Yer Tahsisi

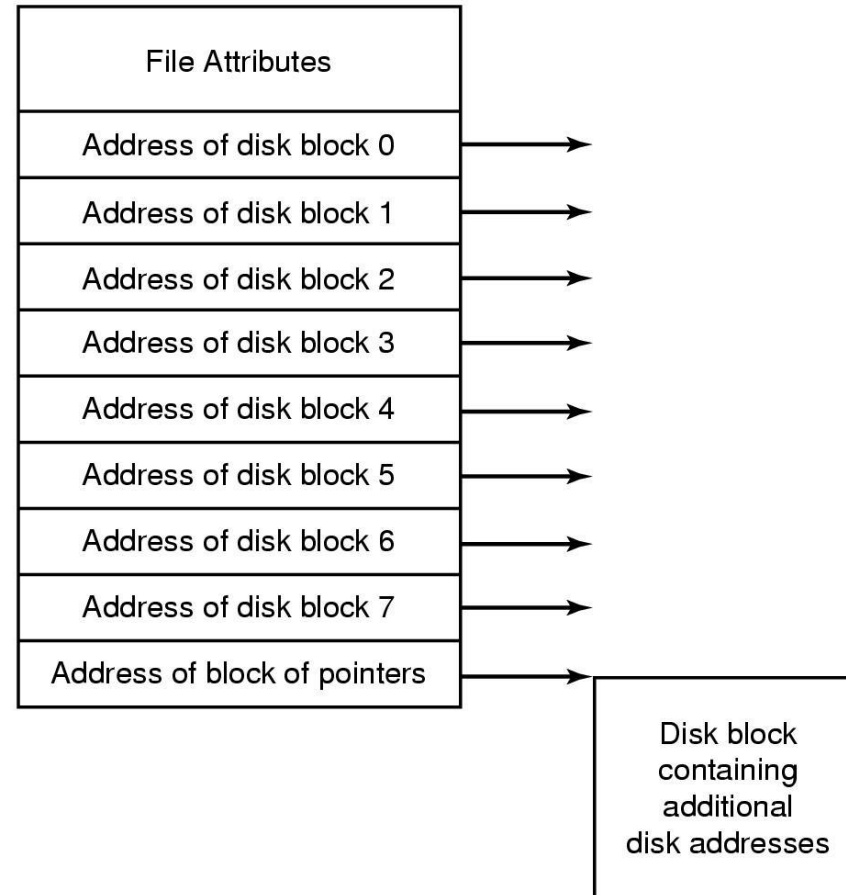
- Tablonun boyutu gerçekten büyük oluyor
- Örneğin, 1 KB bloklu 200 GB disk, 600 MB'lık bir tabloya ihtiyaç duyar
- Tablo boyutunun büyümesi, disk boyutunun büyümesiyle doğru orantılıdır

I-nodes

- Veri yapısını yalnızca etkin dosyalar için bellekte tutar
- Veri yapısı, blokların disk adreslerini ve dosyaların özniteliklerini listeler
- K aktif dosya, dosya başına N blok => en fazla $k \cdot n$ blok
- Büyüme sorununu çözer
- N ne kadar büyük olabilir?
- Tablodaki son giriş, diğer disk bloklarına işaretçiler içeren disk bloğuna işaret eder.

Örnek I-node

• .



Dizinler

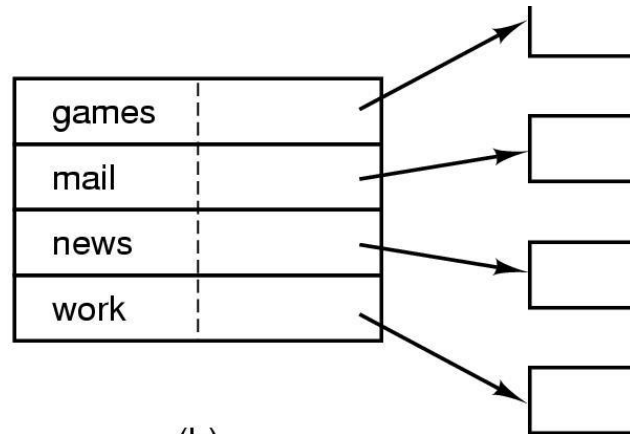
- Open file, dizini bulmak için kullanılan yol adı (path)
- Dizin, aşağıdakileri bilgileri kullanarak blok adreslerini belirtir
 - İlk bloğun adresi (bitişik yer)
 - İlk bloğun sayısı (bağlı liste)
 - i-node sayısı

Dizinler

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

games	attributes
mail	attributes
news	attributes
work	attributes

(a)



(b)

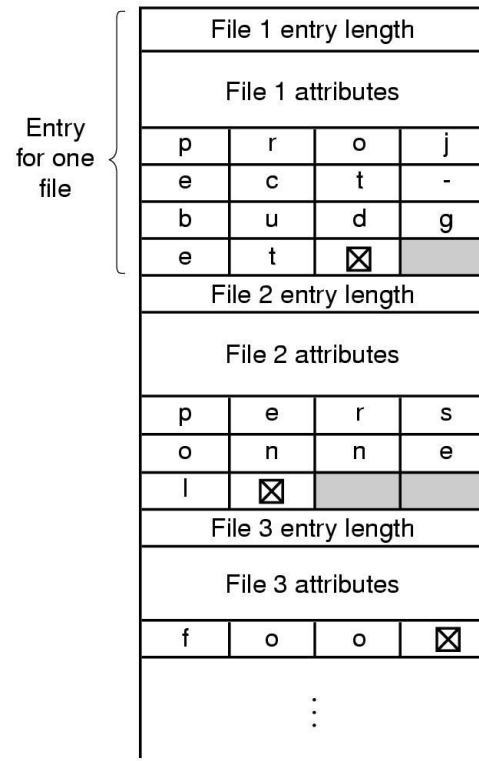
Data structure
containing the
attributes

Dizinler

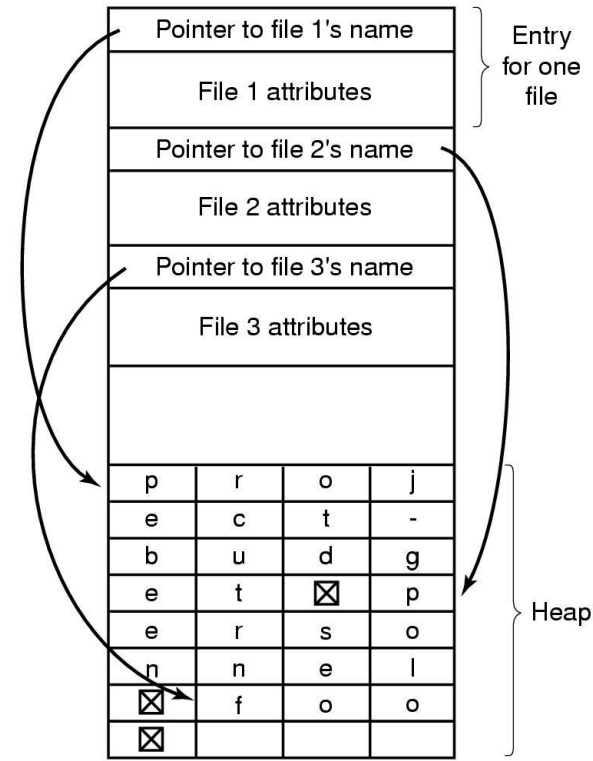
- Değişken uzunluklu adlarla nasıl başa çıkarız?
- Çok uzun adlar problem
- İki yaklaşım
 - Sabit başlık ve ardından değişken uzunluklu adlar
 - Yığın işaretçisi adları işaret eder

Dizinler

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



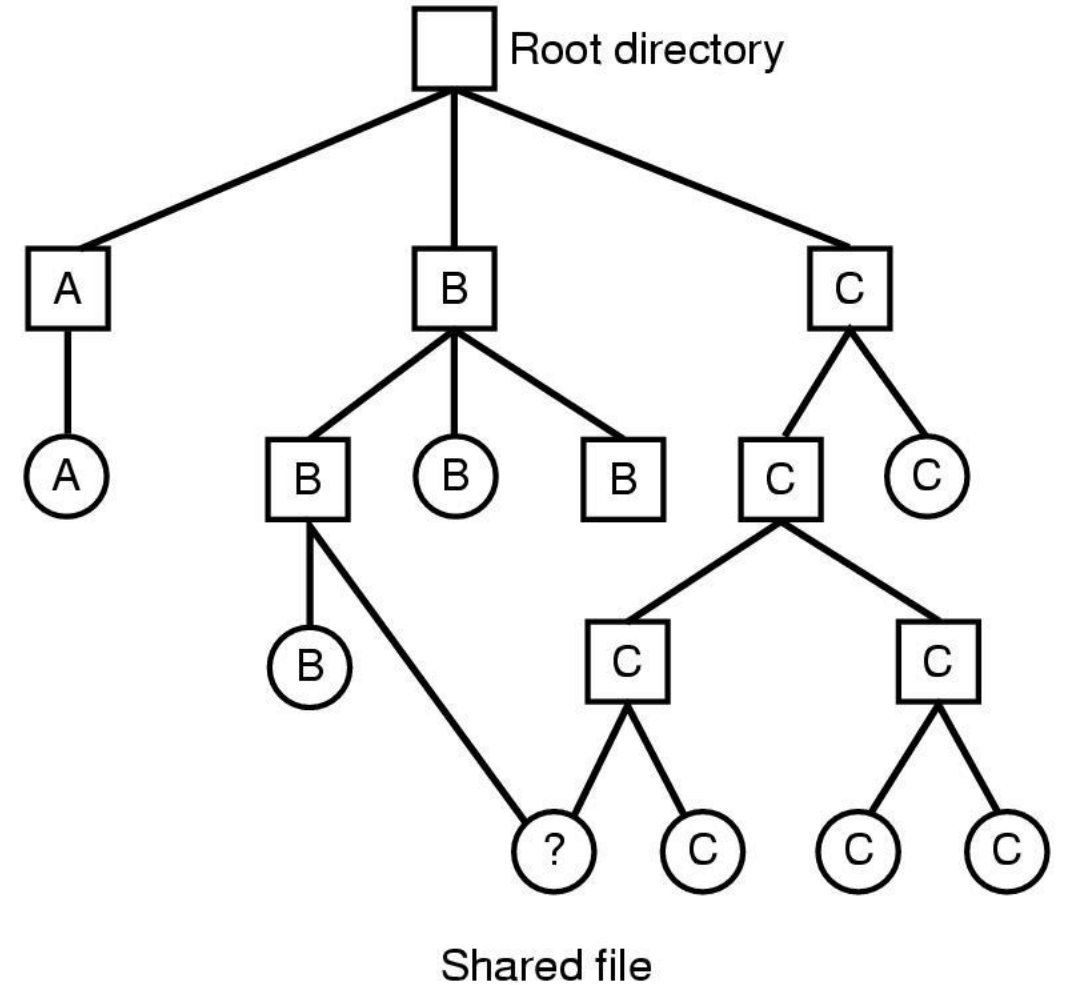
(a)



(b)

Paylaşımlı Dosyalar

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



Paylaşımlı Dosyalar

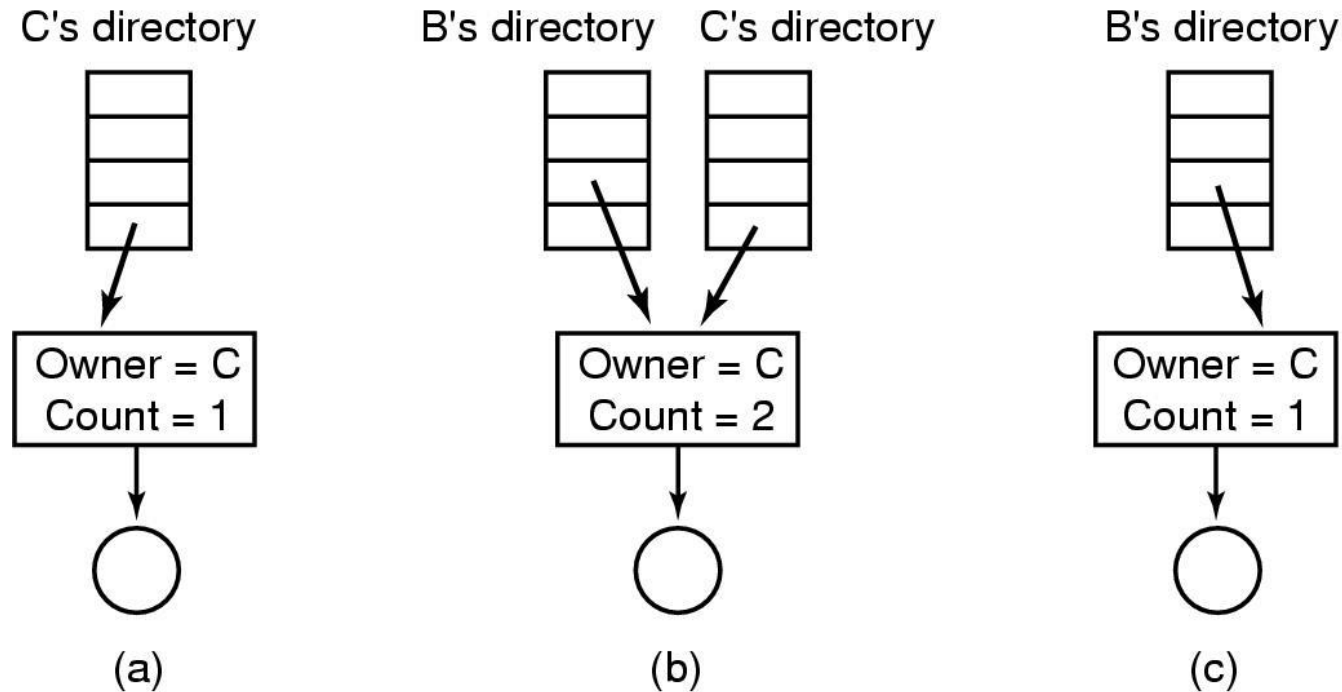
- B veya C yeni bloklar eklerse, diğer sahip nasıl öğrenir?
- Paylaşılan dosyalar için özel i-node kullan - dosyanın paylaşıldığını gösterir
- Sembolik bağlantı (symbolic link) kullanın - sahibi C ise, B'nin dizinine konulan özel bir dosya. Bağlı (linked) olduğu dosyanın yol adını içerir

I-node Problem

- C dosyayı kaldırır, 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 yapılabilir
- 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-düğüm 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

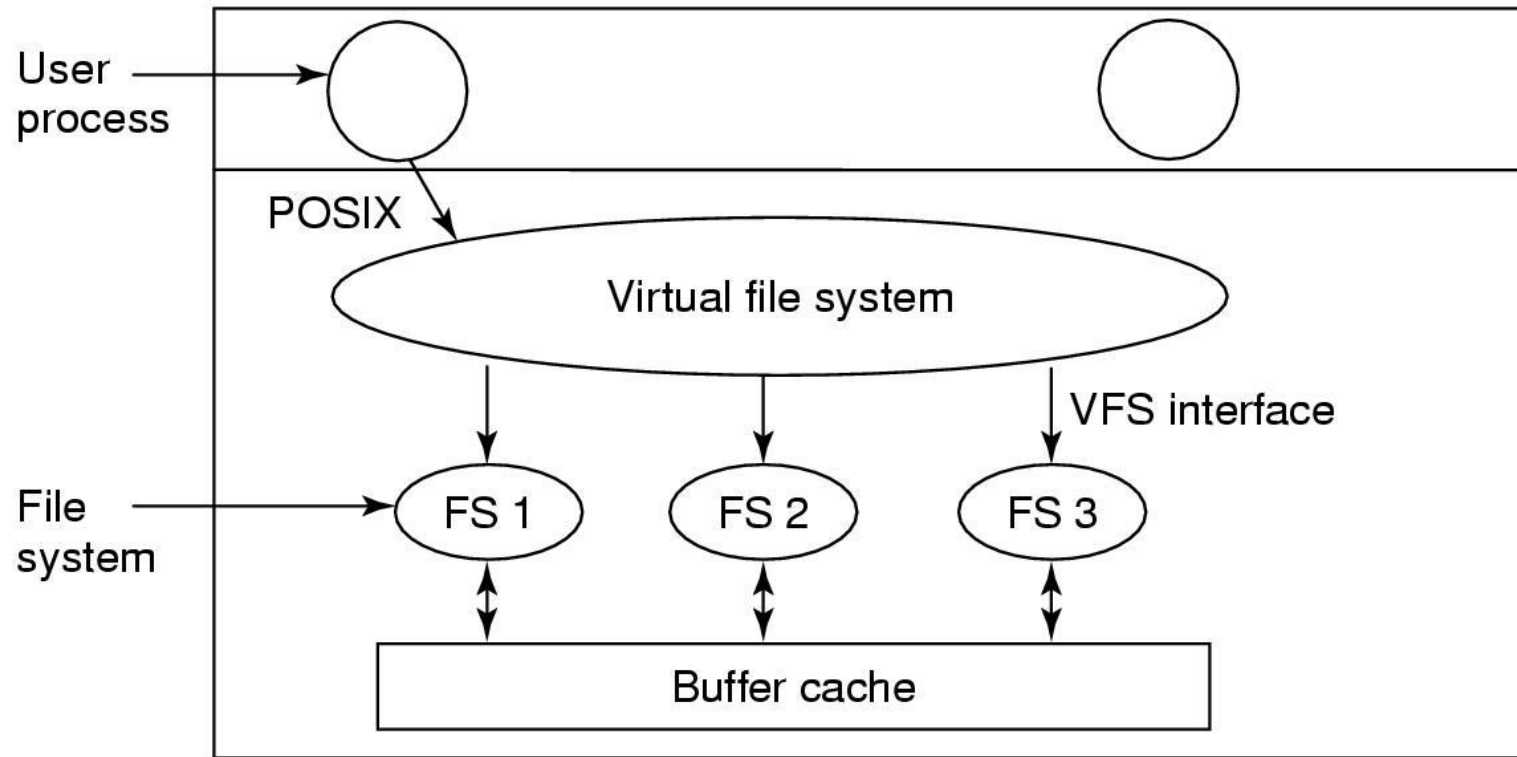
Günlük (journaling) Dosya Sistemleri

- 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

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

Modern Bilgisayarın Bileşenleri:

SON