

Sistem Operasi

Sistem File

2016

Latar Belakang

- Proses menyimpan dan mengambil informasi
- Kapasitas memori proses dibatasi ukuran vmem
- Ketika proses selesai (terminate) isi memori (untuk proses itu) hilang
- Beberapa proses berbagi informasi

- Kebutuhan memori:
 - besar
 - ***persistent***
 - akses jamak (concurrent)

Solusi?
Sistem File!

Sistem File

- Pengguna (kenyamanan)
 - Tidak peduli detail
 - Peduli pada interface
 - Abstraksi ke disk
 - “Satu-satunya sifat bersahabat dari disk adalah di menyimpan data secara handal (persistent).”
 - Divais bisa berbeda: tape, IDE/SCSI, NFS
 - File dan direktori
- OS (efisiensi)
 - Peduli pada implementasinya

Outline

- Sudut Pandang User
 - File ←
 - direktori
- Sudut Pandang OS
- Contoh:

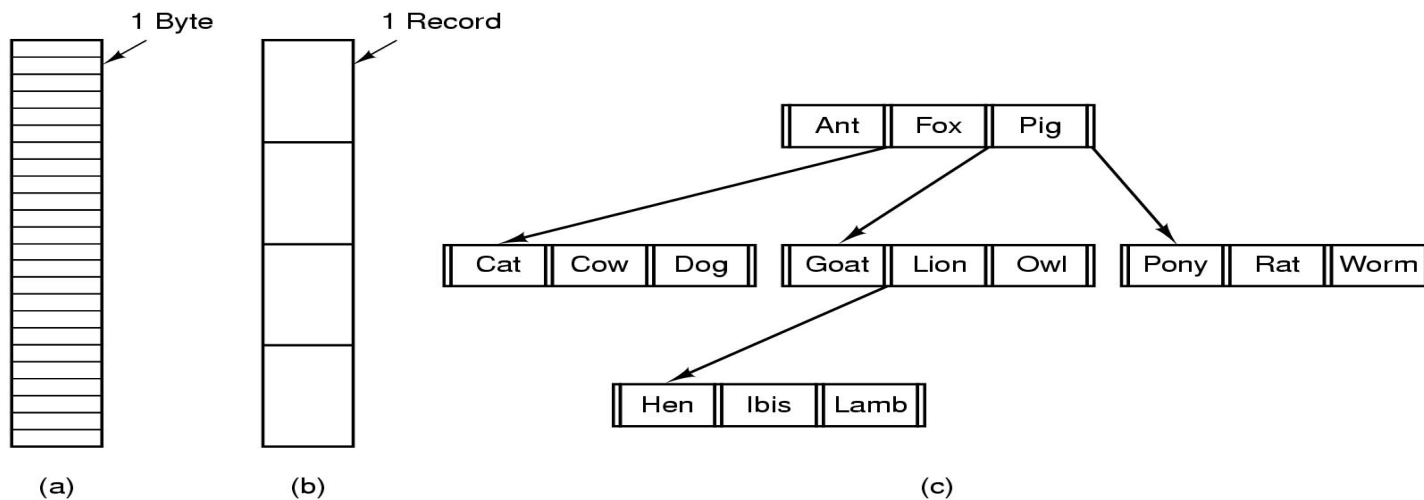
Penamaan File

- Cara penamaan file
 - Diberikan proses yang membuatnya
 - Diakses oleh proses lain menggunakan nama itu
- Bervariasi untuk berbagai OS
 - Semua mendukung penamaan 1 sampai 8 huruf
 - Beberapa mendukung penamaan panjang
 - Beberapa membedakan huruf besar dan huruf kecil.
- Ektensi File
 - Berarti beberapa property.

Standar Penamaan File (ekstensi)

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

Struktur File



- Tiga jenis file

- Urutan byte (Unix, Windows)
- Urutan record (Beberapa dekade lalu)
- tree (komputer mainframe)

Jenis File

- File umum
 - File ASCII
 - File Binary
- direktori
 - File sistem untuk memelihara sistem file
- File berkarakter khusus
 - Chatdev!
- File blok khusus

Akses File

- Akses sekuensial
 - Membaca semua byte/record dari awal
 - Tidak bisa loncat, bisa rewind atau back up
 - Cocok untuk media tape magnetik (tape)
- Akses random
 - Membaca byte/record secara acak
 - Keharusan untuk sistem basis data
 - Membaca dapat berupa ...
 - Pindahkan penanda file (seek), kemudian baca atau ...
 - Baca dan pindahkan penanda file
- Os modern menggunakan akses random

Atribut File

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 has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Operasi File

1. Create
2. Delete
3. Open
4. Close
5. Read
6. Write
7. Append
8. Seek
9. Get attributes
10. Set Attributes
11. Rename

Contoh: Unix `open()`

```
int open(char *path, int flags [, int mode])
```

- `path` adalah nama file
- `flags` adalah bitmap untuk menset switch
 - `O_RDONLY`, `O_WRONLY`...
 - `O_CREATE` then use `mode` for perms
- berhasil, kembali ke index

Contoh : WinNT CreateFile ()

- Returns handle objek file:

```
HANDLE CreateFile (  
    lpFileName, // name of file  
    dwDesiredAccess, // read-write  
    dwShareMode, // shared or not  
    lpSecurity, // permissions  
    ...  
)
```

- Objek file digunakan untuk semua: file, direktori, disk drive, port, pipe, socket dan console

Outline

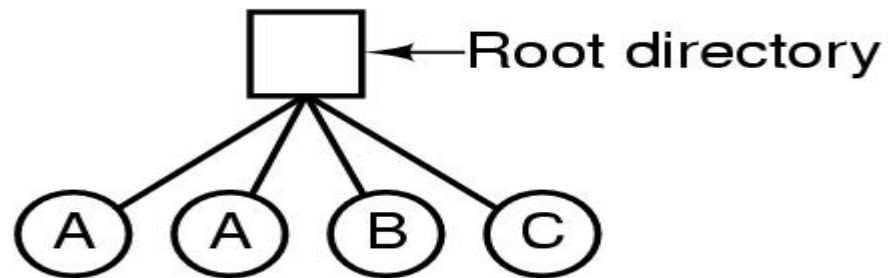
- Sudut Pandang User
 - File
 - Direktori
- Sudut Pandang OS
- Contoh:



← selesai

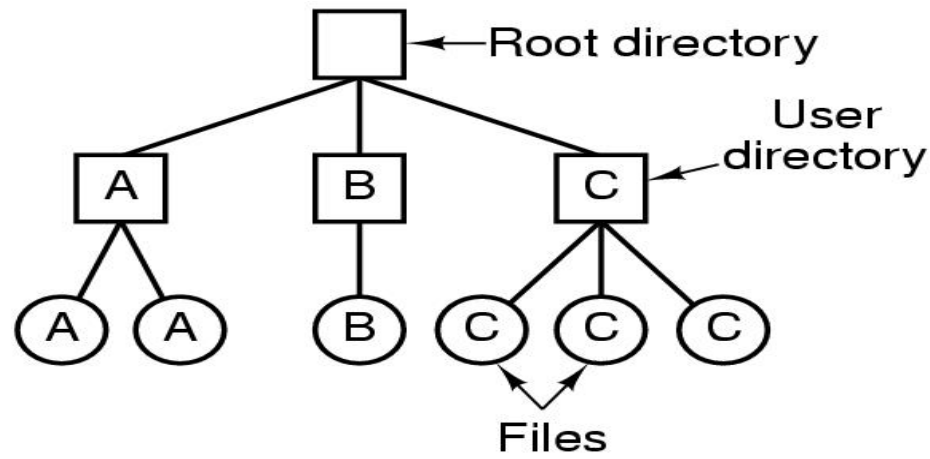
Direktori

Sistem Direktori Satu Tingkat



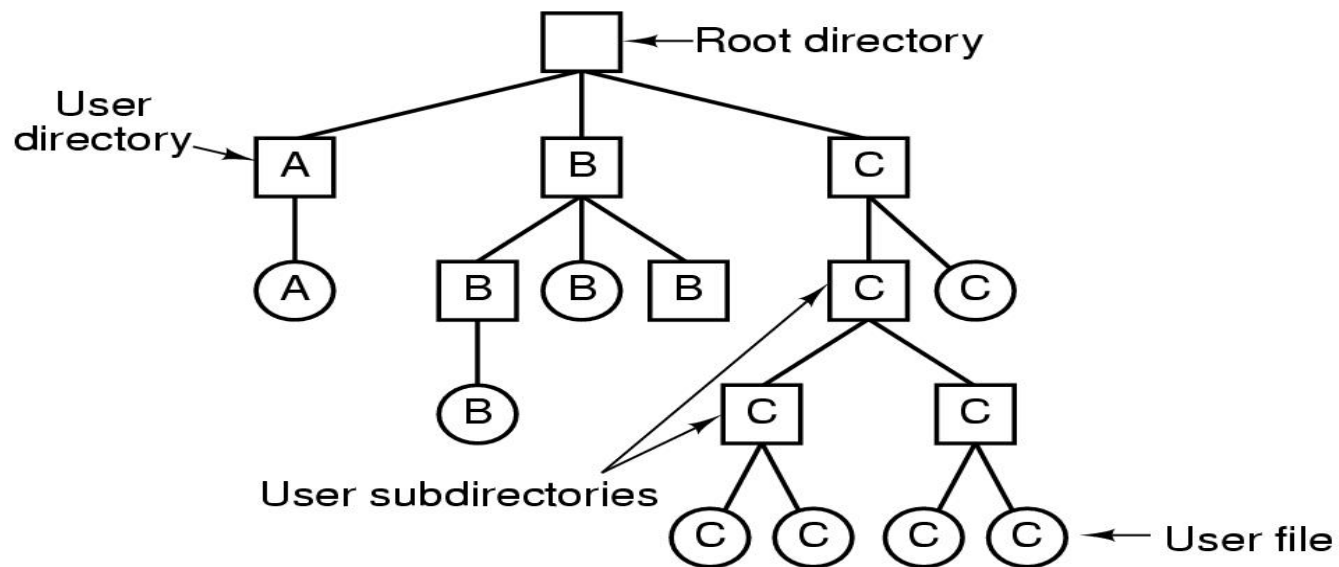
- Sebuah sistem direktori satu tingkat
 - Terdiri dari 4 file
 - Dipunyai oleh 3 orang berbeda, A, B, dan C

Sistem Direktori Dua Tingkat



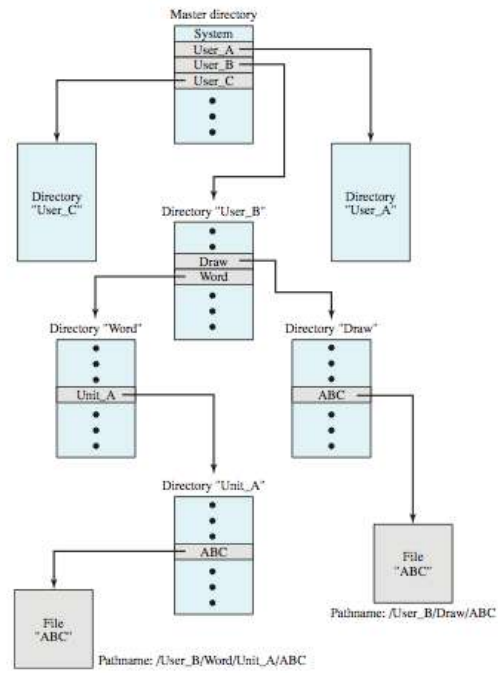
Huruf menyatakan pemilik dari direktori dan file

Sistem Direktori Hirarkis



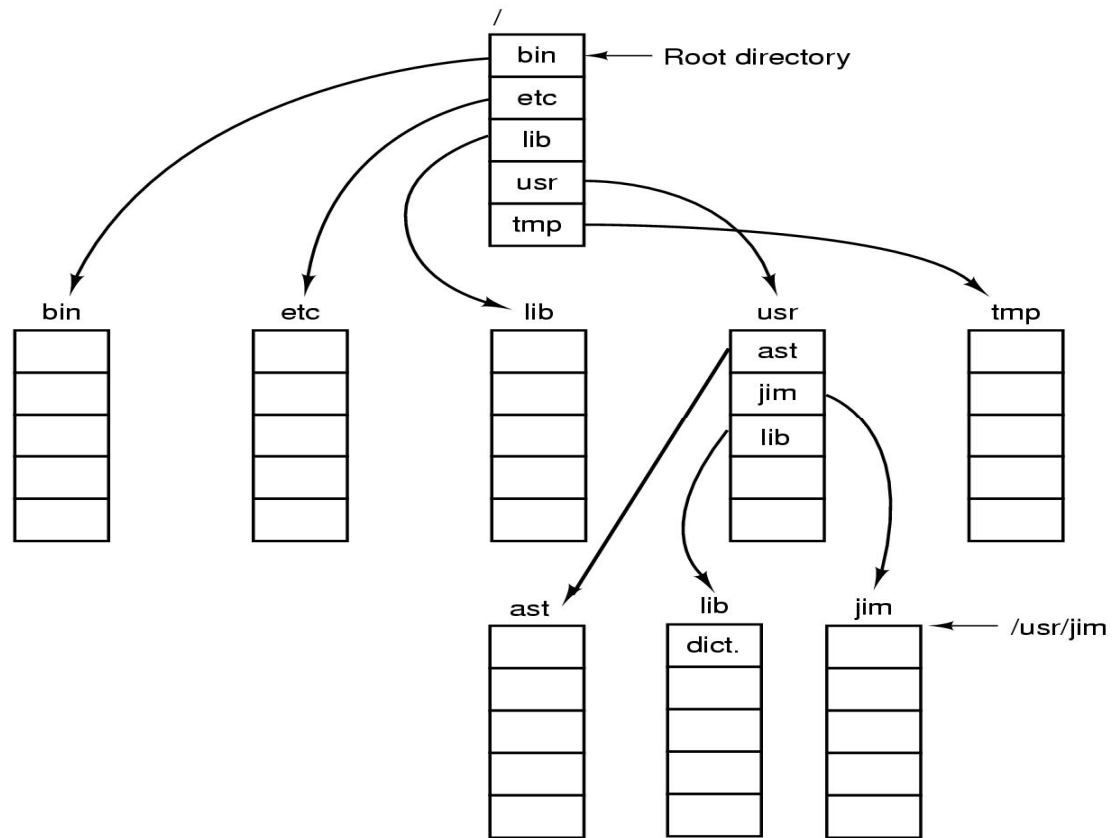
Sebuah sistem direktori hirarkis

Direktori



Path

- Path absolut
 - Contoh: `/usr/include/stdio.h`
 - Pemisah path : `\`, `/`, `>`
- Path relatif
 - Direktori kerja
 - Contoh:
 - Direktori kerja: `/usr/include`
 - Path relatif: `stdio.h`
- `“.”` dan `“..”`



Tree direktori UNIX

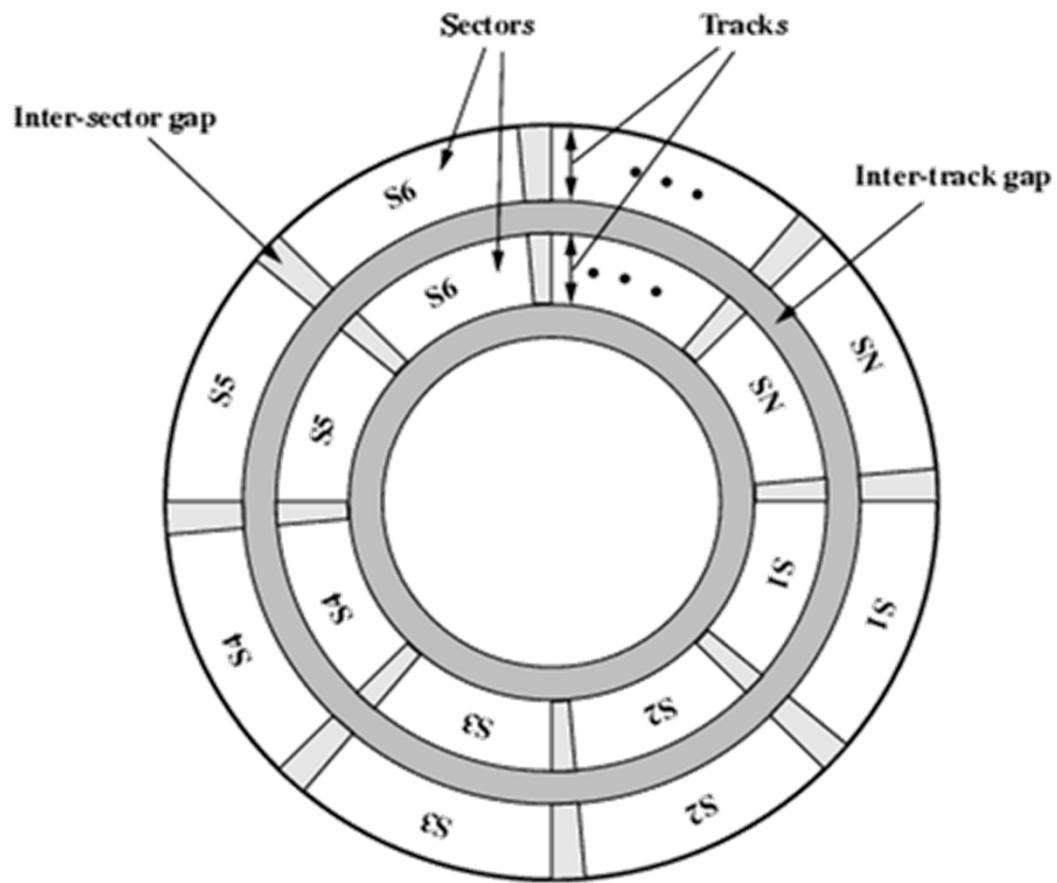
Operasi direktori

1. Create
2. Delete
3. Opendir
4. Closedir
5. Readdir
6. Rename
7. Link
8. Unlink

Outline

- Sudut Pandang User
 - File
 - Direktori
- Sudut Pandang OS
- Contoh:

← selesai
← selesai
←



Kapasistas Disk

- Satu 'cell' berukuran 512B (1/2 kB)
- Sel dibentuk oleh : track/cylinder dan sector
- Track/cylinder adalah lingkaran (lintasan) data yang dibentuk oleh head jika head tersebut diam, jumlah track tetap (dibuat pabrik)
- Sector adalah membagi track menjadi cell-cell, sector **ditentukan** oleh OS

Kapasitas Disk

Jenis media	track	sector	Side	Kapasitas
Disk 160 kB	40	8	1 (single side)	160 kB
Disk 320 kB	40	8	2 (double side)	320 kB
Disk 360 kB	40	9	2	360 kB
Disk 720 kB	80	9	2	720 kB
Disk 1.44 MB	80	18	2	1.44 MB
HD 420 MB	826	63	16	416.304 MB

Disk logik dan fisik

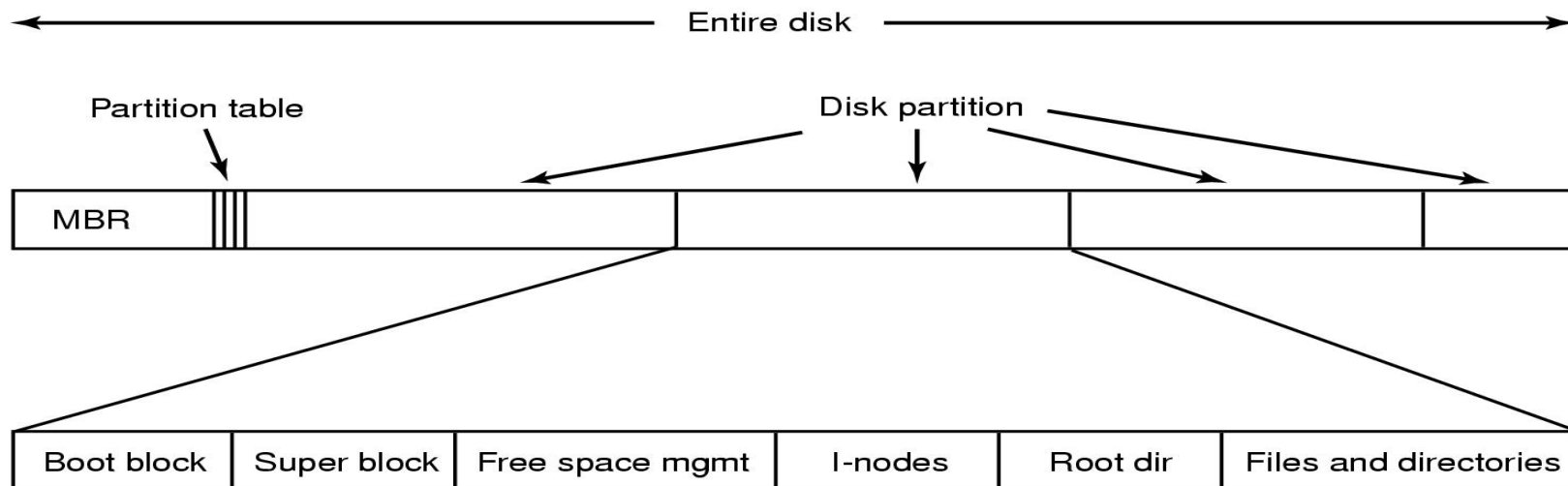
Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free	Fault Tolerance	
94 MB	Simple	Basic		Healthy (...)	94 MB	94 MB	100 %	No	(
Dell Vista Recover...	Simple	Basic	NTFS	Healthy (P...	15.00 GB	8.50 GB	57 %	No	(
Windows-7 (C:)	Simple	Basic	NTFS	Healthy (S...	283.00 GB	22.62 GB	8 %	No	(

Disk	Layout	Type	File System	Status	Capacity	Free Spa...	% Free	Fault Tolerance
Disk 0 Basic 298.09 GB Online	94 MB Healthy (OEM Partit	Dell Vista Recovery Disk (D:) 15.00 GB NTFS Healthy (Primary Partit	Windows-7 (C:) 283.00 GB NTFS Healthy (System, Boot, Page File, Active, Crash Dump, Primar					
CD-ROM 0 DVD (E:)								

Layout Sistem File

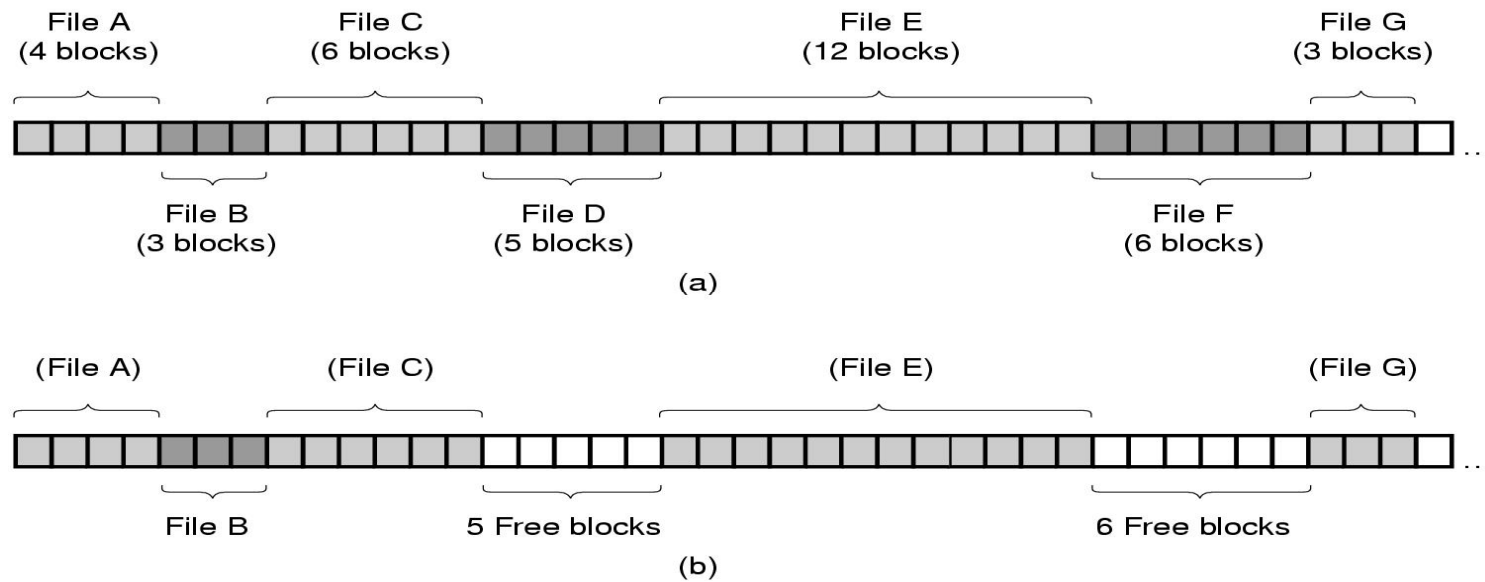
- MBR (Master Boot Record)
 - Sector 0 dari disk
 - Digunakan untuk proses booting
 - Mengandung tabel partisi
 - Satu partisi bertanda aktif
 - Setiap partisi mempunyai sebuah blok boot, untuk memuat OS
- Super blok
 - Jenis sistem file
 - Jumlah blok
 - Informasi lain

Layout Sistem File



Layout sistem file yang mungkin

Implementasi File



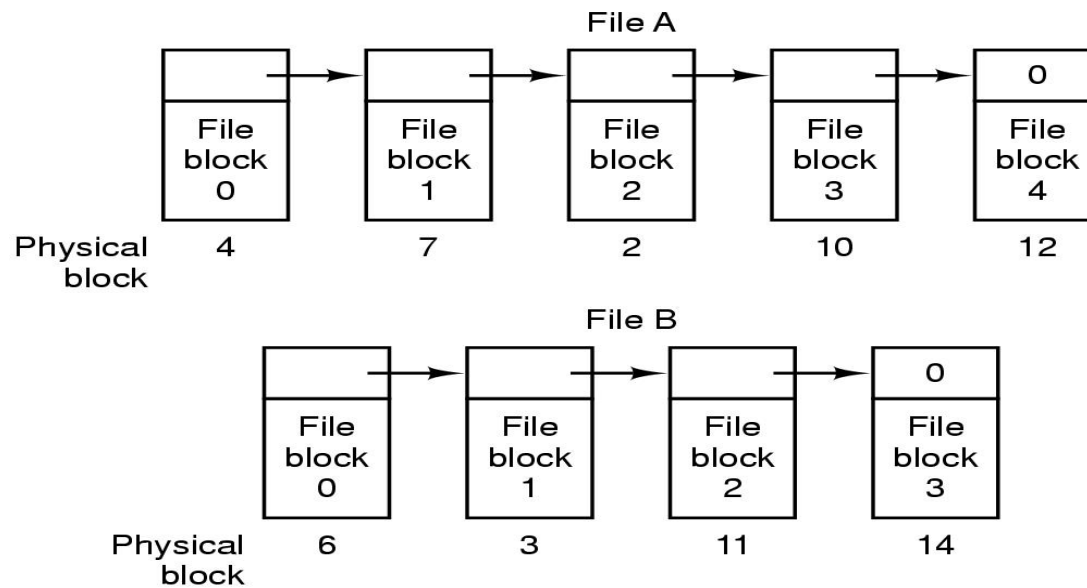
(a) Alokasi ruang di disk secara kontinu untuk 7 file

(b) Kondisi dari disk setelah file *D* dan *F* dihapus

Alokasi kontinu

- Simpan setiap file sebagai urutan blok kontinu di disk (tidak boleh dipecah menjadi beberapa bagian)
- Kelebihan
 - Mudah diimplementasikan
 - Performa baca bagus
- Kekurangan
 - Fragmentasi (ada lokasi di disk yang kosong dilewati karena tidak mencukupi untuk menaruh data)
- Contoh : CD-R!

Linked Lisk dari Blok di Disk



Tanpa Fragmentasi (selalu ruang kosong diisi lebih dahulu),
Akses acak pelan, overhead

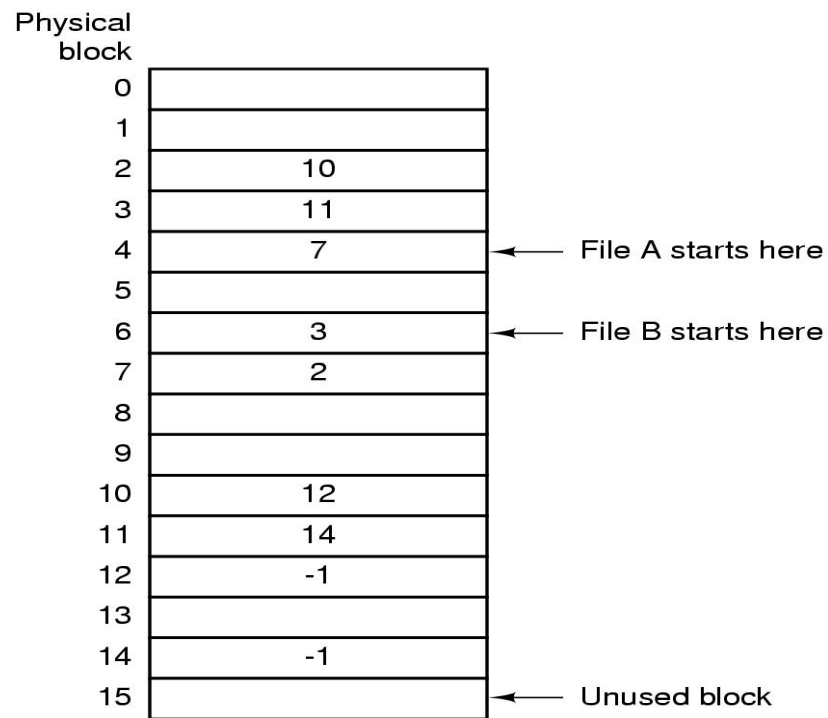
File Allocation Table

- Disk 160 kB (40, 8, 1)
 - → 320 cell (cukup 9 bit alamat cell, dipilih 12 bit = 1.5 x 8 bit), FAT = 480B
 - Data terletak di sel 15,25,71,80 maka akan ditulis di FAT 0000 0000
1001 1111 0000 0001 0000 0100 0000 0111 0000 0101
- Disk 360 kB (40, 9, 2)
 - → 720 cell (cukup 10 bit, dipilih 12 bit)
- Disk 1.44 MB (80, 18, 2)
 - → 2880 cell (pas 12 bit)

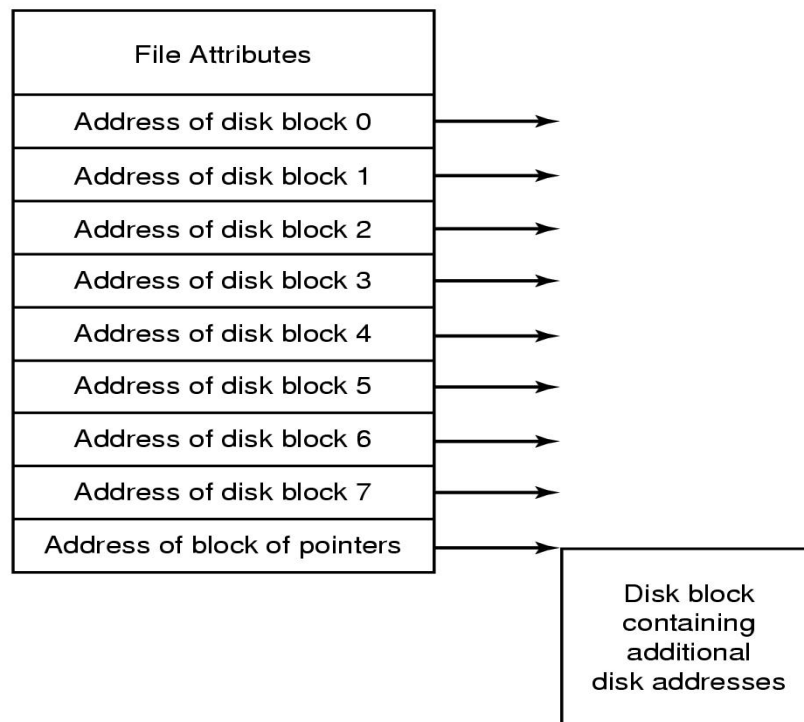
Cluster

- HD 13 MB (200, 63, 2)
 - → 25200 cell (butuh 15 bit, dipilih 16 bit)
 - → FAT = 25200 x 16 bit = 50400 B (99 cell)
- Mulai dikenalkan konsep **cluster** (gabungan cell)
- HD 72 MB (400, 63, 4)
 - → 100800 cell (butuh 17 bit, dipilih 16 bit), FAT = 394 cell
- HD 72 MB (800, 63, 8)
 - → 403200 cell (butuh 19 bit, dipilih 16 bit)

Linked list menggunakan sebuah tabel di RAM



I-Node (index-node)



- Asosiasikan setiap file sebuah i-node, yaitu daftar dari alamat blok
- Hanya alamat disk dengan jumlah tetap, bagaimana jika semua digunakan?

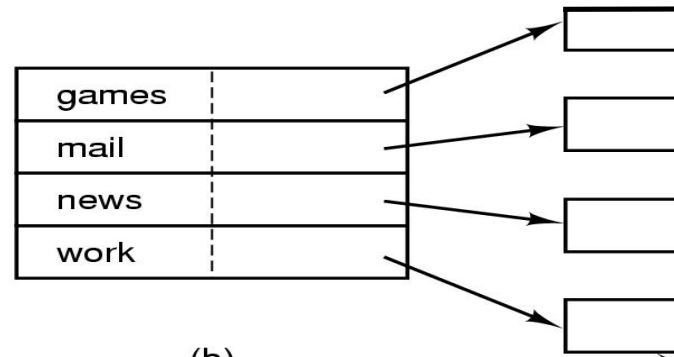
Implementasi Direktori

- Sebelum sebuah file bisa dibaca, harus dibuka dahulu
- Masukan direktori menyediakan info dari file
 - Alamat disk dari seluruh file (alokasi kontinu)
 - Jumlah dari blok pertama (linked list)
 - Jumlah dari i-node (i-node)
- Dimana atribut file harus disimpan
 - Di masukan direktori (Windows)
 - Di i-node (Unix)

Penyimpanan atribut file

games	attributes
mail	attributes
news	attributes
work	attributes

(a)



(b)

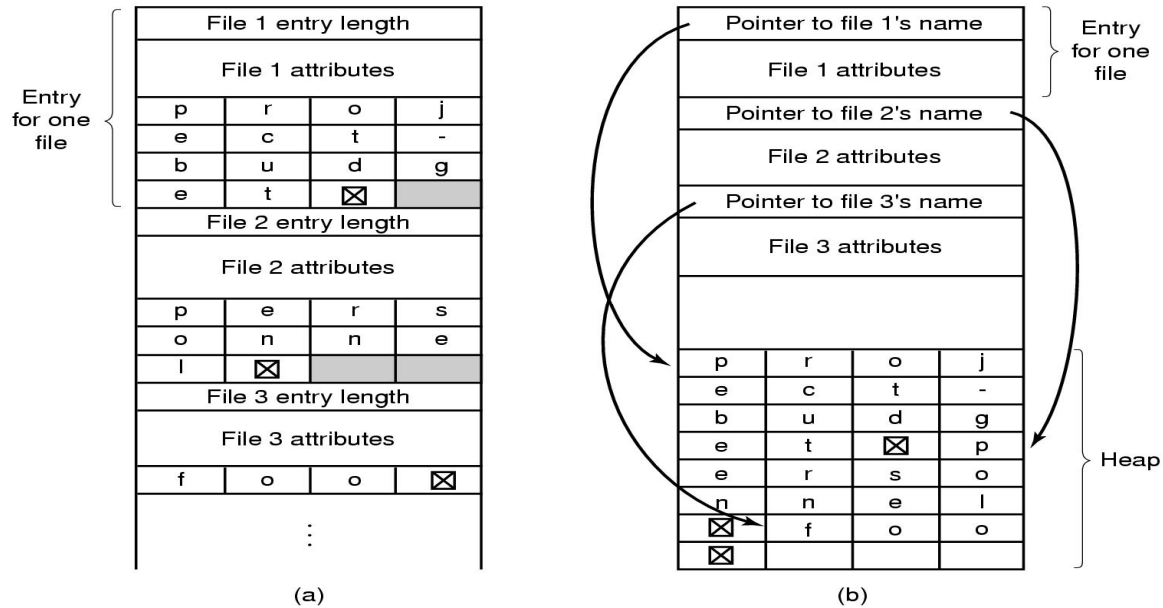
Data structure
containing the
attributes

- (a) Sebuah direktori sederhana dengan ukuran masukan tetap
alamat disk dan atribut di masukan direktori
- (b) direktori di setiap masukan merefer ke sebuah i-node

Nama file panjang di direktori

- File-name ukuran sama
 - Cadangkan 256 huruf untuk setiap nama file
 - Sederhana, tapi memboroskan banyak ruang direktori
- Header panjang tetap diikuti oleh nama-file panjang variabel
 - Gap ukuran variabel ketika sebuah file di hapus
 - Page fault bisa terjadi ketika membaca sebuah file-name
- Masukkan Fixed-length direktori plus nama-file di sebuah heap

Nama file panjang di direktori



- a. Header dg panjang tetap diikuti nama-file panjang variabel
- b. Masukan direktori panjang tetap plus nama-file di sebuah heap

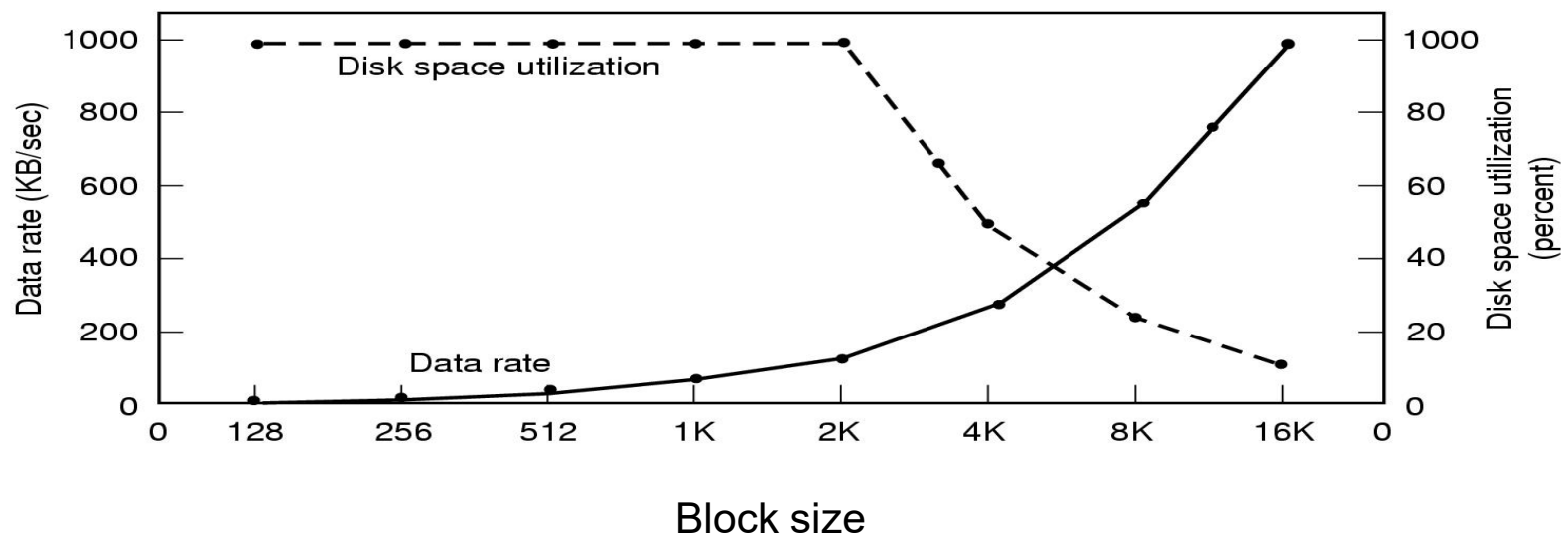
Manajemen Ruang Disk

- n byte
 - menyambung
 - blok
- Mirip dengan manajemen memori
 - Kontinu seperti partisi ukuran variabel
 - tapi berpindah (move) di disk sangat lambat!
 - Sehingga menggunakan blok
 - blok seperti paging
- Ukuran blok
- Blok bebas
- Quota disk

Ukuran Blok (Cluster)

- Laju data = Ukuran data / Waktu akses
 - Waktu akses = waktu seek + delay rotasi + waktu transfer
- Efisiensi ruang
 - kebanyakan file kecil (2KB)
- Blok besar
 - throughput lebih cepat, waktu seek lebih kecil
 - ruang terbuang (fragmentasi internal)
- Blok kecil
 - ruang terbuang lebih sedikit
 - lebih banyak waktu karena blok lebih banyak
- (Catatan, ukuran blok di disk umumnya 512 byte, tapi blok logik (*cluster*) sistem file dipilih saat formatting)

Ukuran blok



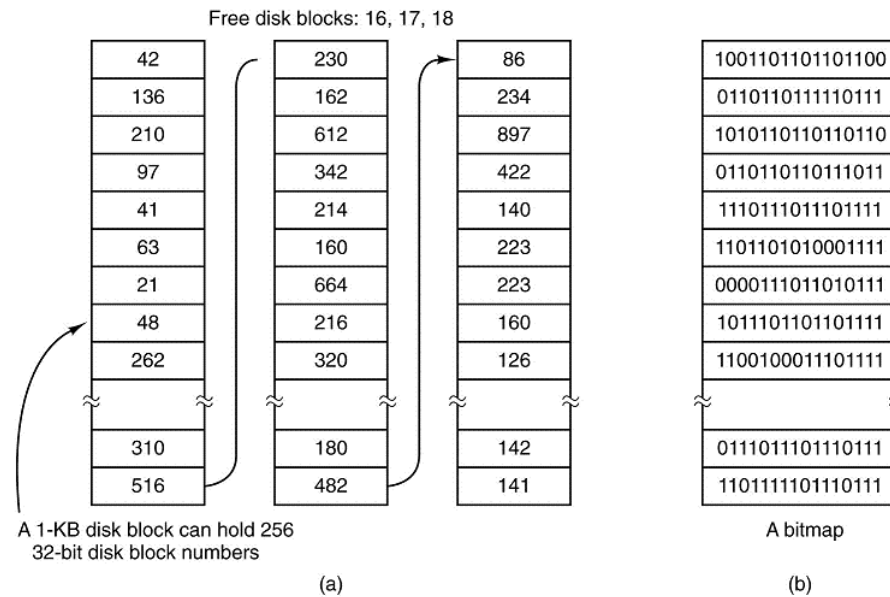
- Garis tegas adalah laju data dari disk
- Garis putus adalah efisiensi ruang disk
- Semua file 2KB

Pencatatan Blok Bebas

(note, these are stored on the disk)

- Dua metoda
 - linked list dari blok disk
 - bitmap dari blok disk
- Linked List dari Blok Bebas (banyak per blok)
 - blok 1KB, nomor blok disk 16 bit (2 byte)
 - Setiap blok menyimpan 512 nomor blok
 - Disk kosong 200 MB memerlukan nomor blok 200K, sehingga memerlukan 400 blok (400KB) untuk menyimpannya
 - Berapa jika disk kosong 20MB?
- Bit Map
 - Satu blok (tidak hanya yang kosong) memerlukan satu bit
 - Disk 200 MB mempunyai 200K blok, sehingga memerlukan 200K bit(25KB)
 - 1 bit vs. 16 bit

Pencatatan Blok Bebas



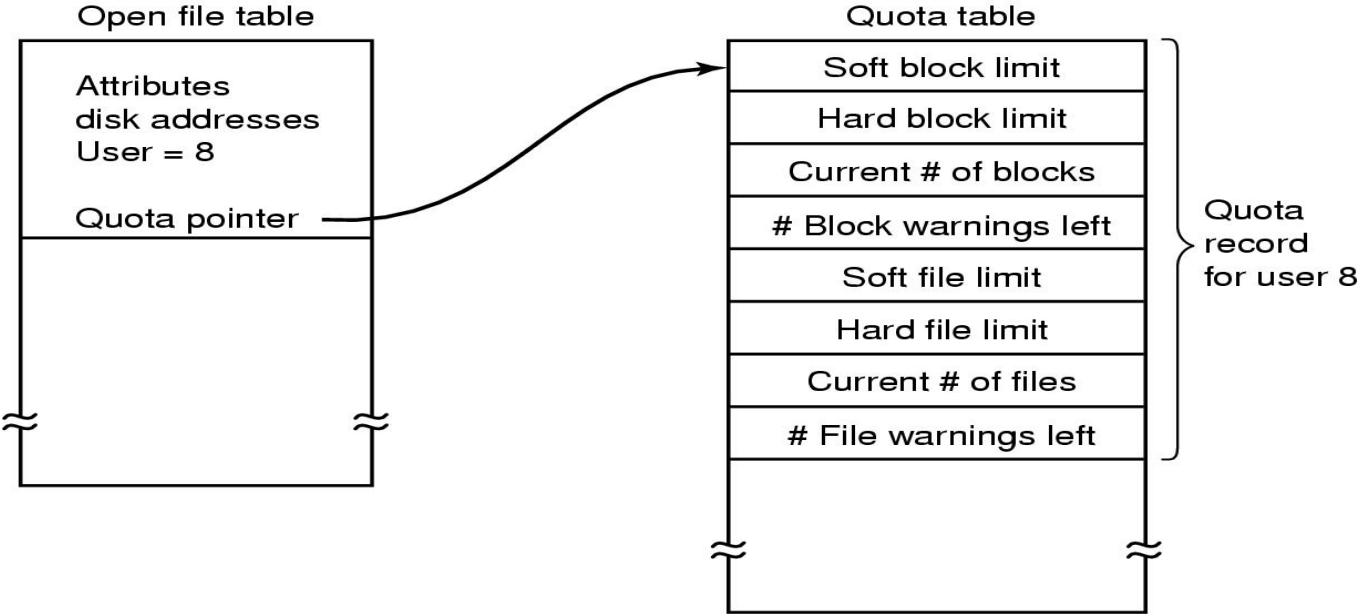
(a) Menyimpan daftar bebas di sebuah linked list

(b) Sebuah bit map

Quota Disk

- Tabel 1: tabel file open di memori
 - Ketika ukuran file berubah, kenakan ke user
 - user mengindeks ke tabel 2
- Tabel 2: quota record
 - Pemeriksaan soft limit, melebihi diperbolehkan dengan peringatan
 - hard limit tidak pernah dilewati
- Limit: blok, file, i-node

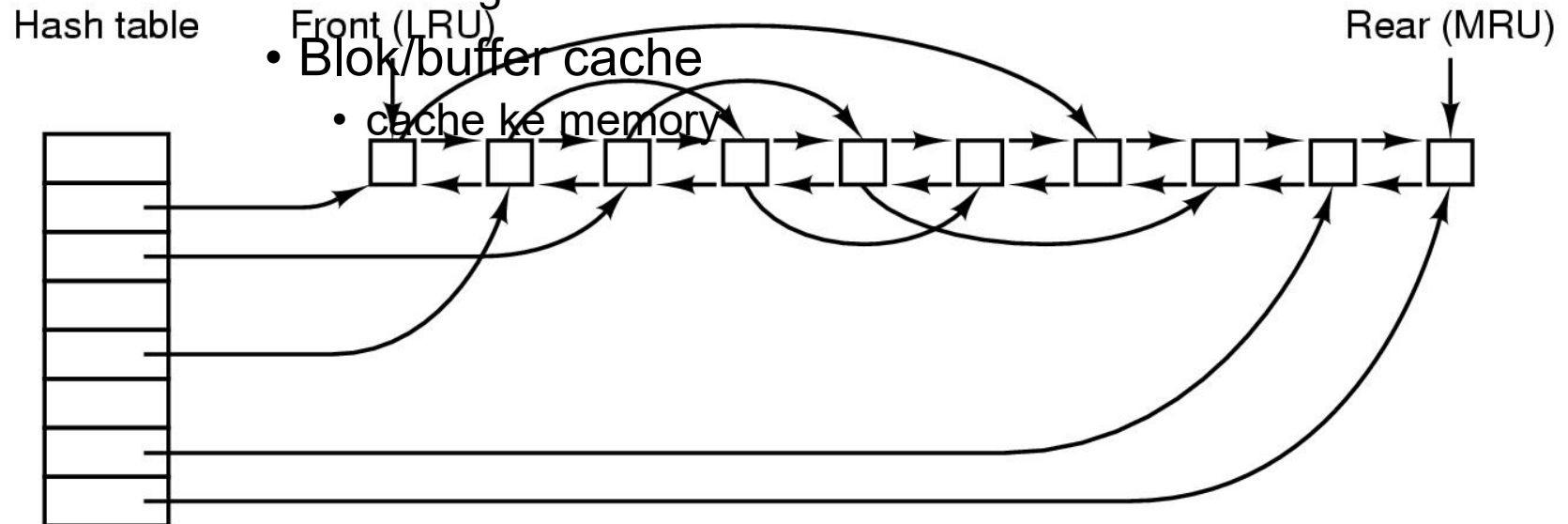
Quota Disk



Quota untuk mencatat setiap pemakaian disk oleh user

Caching

- Akses disk 100.000x lebih lambat dari memori
 - kurangi kebutuhan akses disk!
- Blok/buffer cache
 - cache ke memory



Caching

- Full cache? FIFO, LRU, 2nd chance ...
 - Tidak sesering paging
 - LRU secara tepat dapat dilakukan
- LRU kadang tidak mencukupi
 - crash dg i-node bisa menyebabkan kondisi tak konsisten
 - i-node di ekor diperlukan sementara sebelum ditulis ulang
- Enhanced LRU
 - Apakah blok ini kelihatannya akan diperlukan lagi segera?
 - Apakah blok ini vital untuk konsistensi dari sistem file?

Enhanced LRU

- Apakah blok ini vital untuk konsistensi dari sistem file?
 - Tulis segera!!
- Apakah blok ini akan diperlukan lagi segera?
 - jika tidak, simpan di awal daftar
 - jika ya, simpan di akhir daftar
- Sekali-kali tulis semua
 - sync

Outline

- Sudut Pandang User
 - File
 - Direktori
- Sudut Pandang OS
- Contoh:

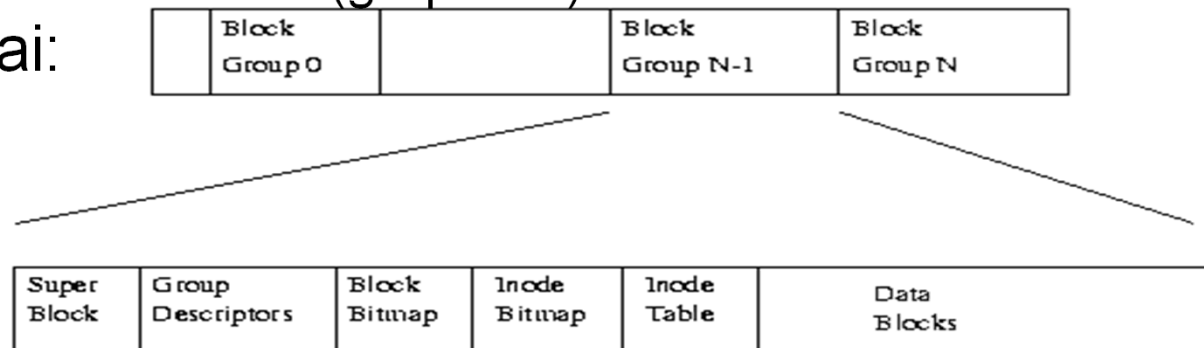
← selesai

← selesai

← selesai

Sistem file Linux : blok

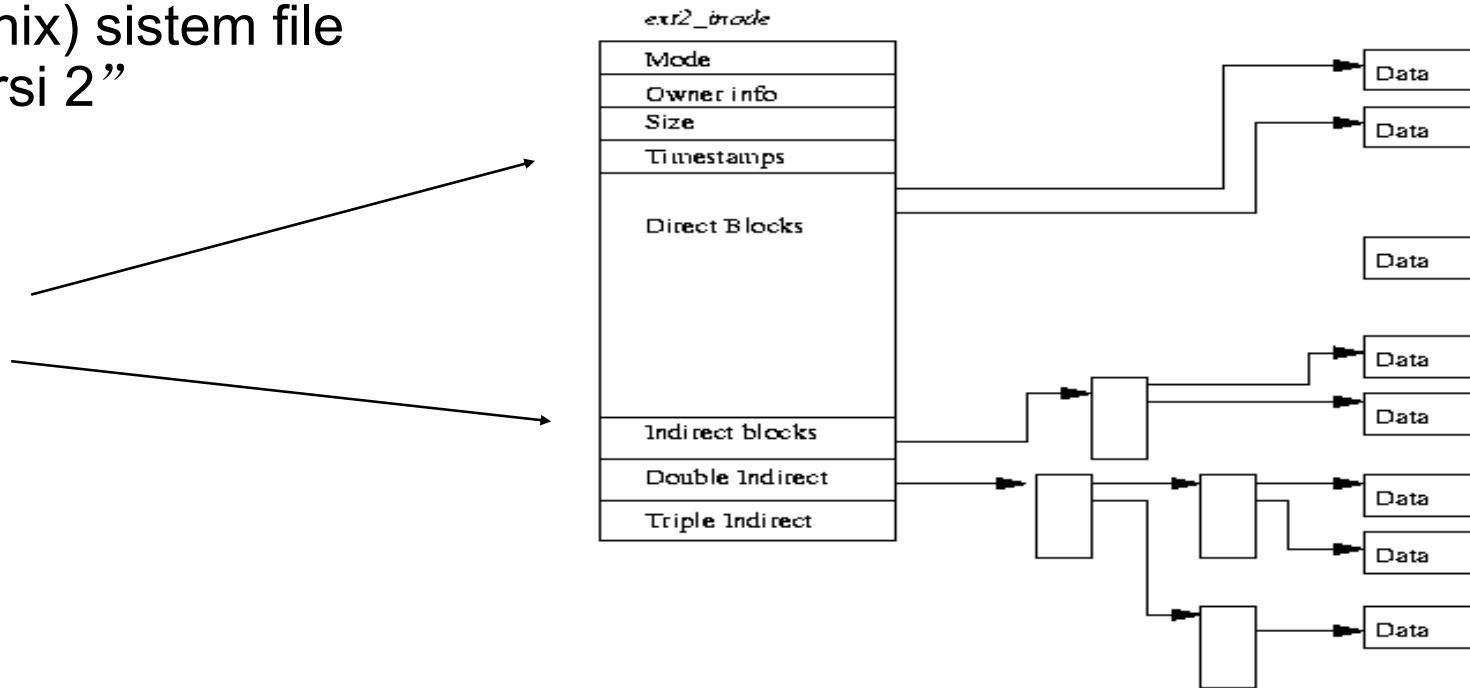
- Default blok 1 Kb
 - kecil!
- Untuk performa lebih tinggi
 - lakukan I/O dalam chunks (mengurangi request)
 - request cluster bersebelahan (grup blok)
- Grup mempunyai:
 - bit-map dari blok bebas dan I-node
 - copy dari super block



Sistem file Linux: ext2fs

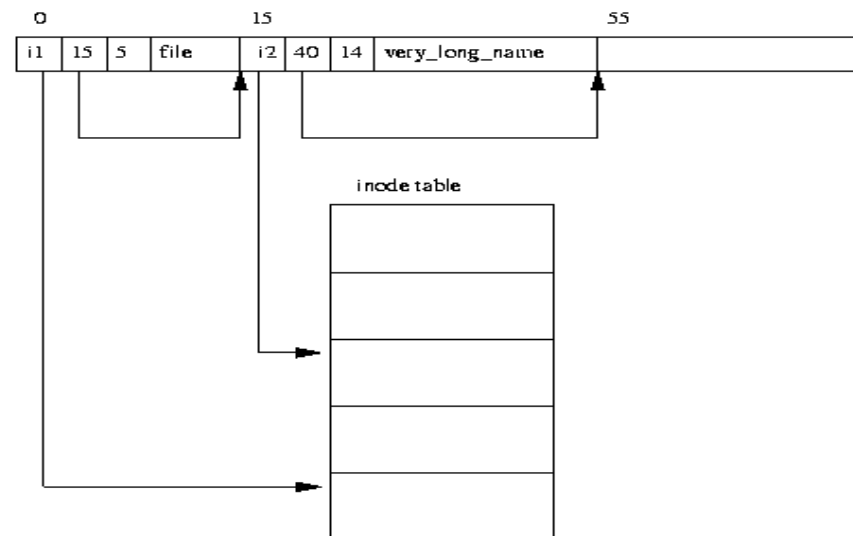
- “Extended (dari minix) sistem file versi 2”

12

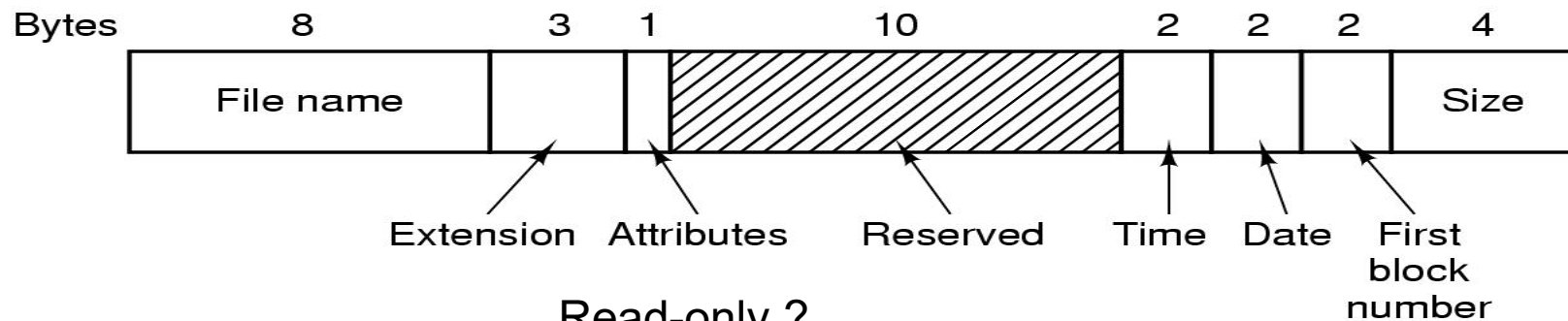


Sistem file Linux : direktori

- File khusus dengan nama dan inode



Sistem file MS-DOS (1)



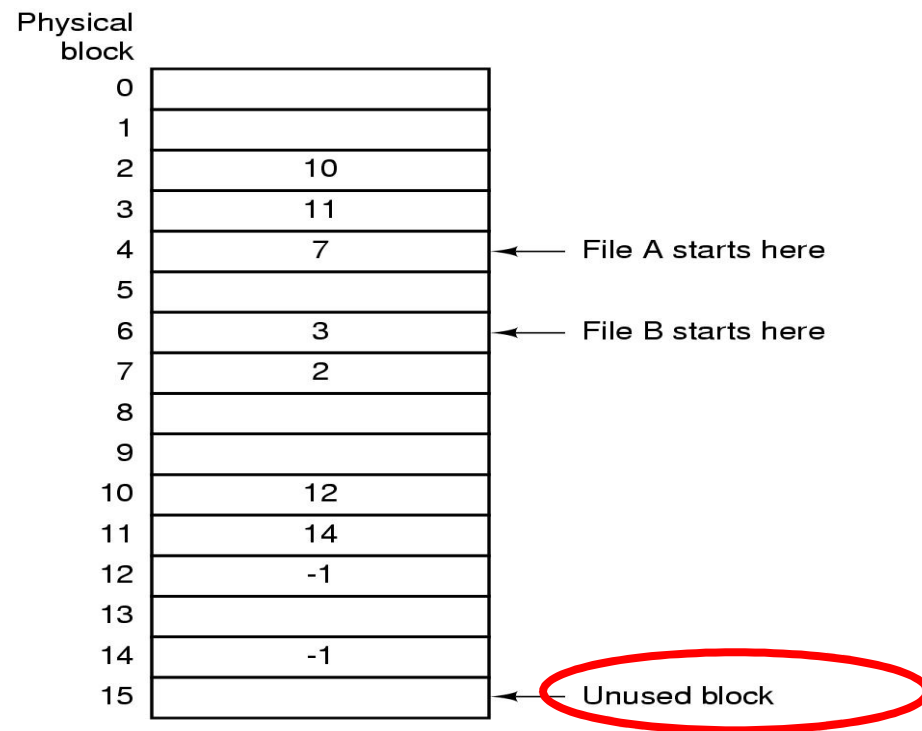
Read-only ?
Archive?
Hidden?
System file?
Masukan direktori/file MS-DOS

The MS-DOS File System (2)

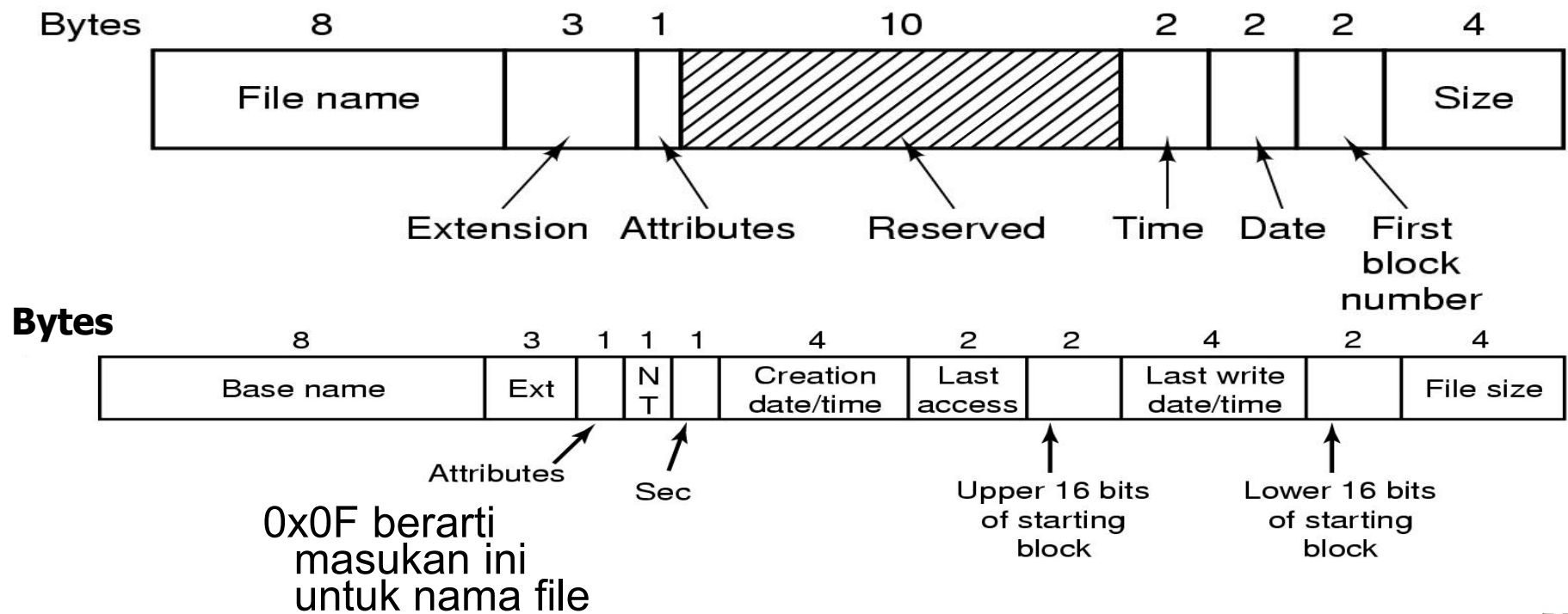
Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

- Partisi maksimum untuk berbagai ukuran blok
- Kotak kosong menyatakan tak didukung

Sistem file MS-DOS(3)



Sistem File Windows 98



Review

- Terangkan beberapa latar belakang terbentuknya sistem file.
- Apa yang dimaksud dengan MBR? Apa kegunaannya?
- Jelaskan dua cara menyimpan file
- Jika sebuah ukuran blok besar yang dipilih, apa yang menjadi kelebihan dan kekurangan utamanya?
- Apa metoda untuk mengatur ruang kosong di disk?