

File System (Interface) – Ch. 10

SISTIM OPERASI
(Operating System)
IKI-20230

Johny Moningka
(moningka@cs.ui.ac.id)

Fakultas Ilmu Komputer
Universitas Indonesia
Semester 2000/2001

File-System Interface

- Review (Bab 2): Storage Structure
- File Concept
- Access :Methods
- Directory Structure
- Protection
- Management File System

File System JM -2000/v1.1/2

File System Interface

■ Untuk kenyamanan user

- OS menyediakan sudut pandang logik yang sama bagi semua media (jenis storage):
 - Segi akses (read dan write) ke storage
 - Logical address: cara mengambil dan menyimpan data.
- Interface: standard interaksi dan manipulasi informasi untuk “unit unit storage” yang berbeda.
- Penulisan program: menyediakan standard operasi (system call, library) untuk melakukan akses ke isi file
 - Sekumpulan bit, byte, record, atau file.
 - Sederhana sampai “complex structure” (seperti index file, data base dll.)

File System JM -2000/v1.1/3

File Structure

- Tidak diperlukan: sekumpulan bytes yang menempati lokasi yang “contiguous” pada storage.
- Simple record structure
 - Lines: sebaris teks, dengan terminasi tanda akhir baris: “newline”.
 - Karakteristik: pada saat manipulasi perlu melacak tanda akhir baris.
 - Fixed length: logical terbagi dalam record yang panjang tetap.
 - Panjang data seragam: tidak ada overhead untuk mengambil data.
 - Kemungkinan space terbuang percuma.
 - Variable length: secara logical terbagi sesuai dengan banyaknya data.
 - Panjang tidak seragam: perlu prosesing dan penyimpanan informasi panjang “record”.
 - Efisien penggunaan space.
- Complex Structures: formatted document, indexing dll.
- Q: bagaimana OS menyimpan informasi tersebut.

File System JM -2000/v1.1/4

File Attributes

- **Nama:**
 - Identifikasi untuk user, dalam bentuk teks (human-readable form).
- **Tipe:**
 - Diperlukan untuk sistim yang mendukung berbagai jenis/isi file.
 - Informasi ini dapat digunakan oleh user/program untuk manipulasi isi file.
- **Lokasi:**
 - Pointer ke lokasi (fisik) dari file pada device (dalam bentuk posisi dari awal, atau nomor blok)
- **Size**
 - Besarnya isi file (misalkan dalam satuan bytes).

File System JM -2000/v1.1/5

File Attributes (2)

- **Proteksi**
 - Mengendalikan siapa saja yang dapat membaca, menulis (update), menghapus, execute dll. –
 - Proteksi penting untuk multiuser system.
- **Time, date, and user identification**
 - Monitor dan informasi lain dari file.
- **File attributes:** umumnya disimpan pada suatu direktori (file khusus) yang dikenal oleh OS.

File System JM -2000/v1.1/6

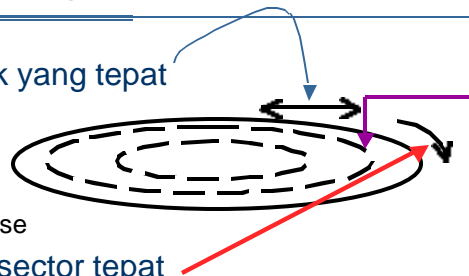
Review: Disk

- Faktor yang mempengaruhi waktu akses (latency) dari disk:
 - Seek
 - Rotation
 - Transfer

File System JM-2000/v1.1/7

Review: Disk (latency)

- **Seek: gerak disk arm ke track yang tepat**
 - best case: 0ms (on track)
 - worst: ~30-50ms (sepanjang disk)
 - average: 10-20ms, 1/3 worst case
- **Rotational delay: menunggu sector tepat berputar di bawah head**
 - best: 0ms (di atas sector)
 - worst: ~10ms (satu rotasi)
 - average: ~5ms (1/2 worst case)
- **Transfer bandwidth: kirim data satu sector ke controller.**
- **Total disk access: Seek + rotation + transfer time**
 - Read a single sector: $10\text{ms} + 5\text{ms} + 50\mu\text{s} \approx 16\text{ms}$



File System JM-2000/v1.1/8

Review: Disk – big slooowwww

DISK

- Minimal transfer: sector
- Write = sector
- Random akses: 10ms
 - worst case: 30 ms
- Sequential access: 20MB/s
- Cost: \$ 0.004 per MB ('2001 PC M-2)

Memory

- (umumnya) bytes
- byte, word
- 100 ns
 - rata-rata
- 200-1000MB/s
- \$ 0.35 per MB

File System JM-2000/v1.1/9

Review: OS & File Systems

- OS => file system sebagai interface dengan tempat penyimpanan tetap (storage)
- Realitas fisik (hardware):
 - Penyimpanan dalam bentuk blok data (sektor dari disk).
 - Identifikasi dengan nomor track, sector dan silinder.
 - Tidak ada proteksi/perlindungan akses.
- File system:
 - Akses dalam urutan arus (alamat) byte.
 - Nama file dapat diberikan pada sekumpulan data.
 - Proteksi milik (privacy) dan akses.

File System JM-2000/v1.1/10

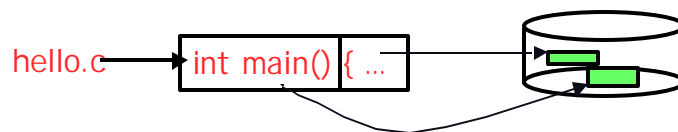
Review: File system

- **Bentuk isi file:**
 - Tidak terstruktur.
 - Kumpulan data dengan panjang variabel:
 - Baris (Text line).
 - Record variable length.
 - Kumpulan data dengan panjang tetap:
 - Record fixed length
- **Atribut untuk file (karakteristik)**
 - Nama
 - Panjang file
 - Time stamp
 - DII.

File System JM -2000/v1.1/11

Files: named bytes on disk

- **File abstraction:**
 - user's view: nama dalam sekumpulan urutan bytes



- File system view: kumpulan blok (cluster) data
- Terjemahkan nama & offset ke disc blok

offset:int → disk addr:int

File System JM -2000/v1.1/12

Review: File system

■ User (programmer) view:

- Kumpulan bytes (tidak ada struktur, mis. UNIX).
- Kumpulan records (IBM, RMS VMS).
- Kumpulan tables/index/objects (database, multimedia).
- Operasi dan manipulasi file.

■ OS (system) view:

- Kumpulan blok data (logical unit transfer dari/ke I/O storage): tipikal ukuran blok 4 Kbytes.
- Kumpulan header file, pointer ke blok data (sector) pertama dari file etc.
- Interface operasi dan manipulasi file: system call, data structure (table open file, etc).

File System JM -2000/v1.1/13

File Operation

■ Create file:

- (1) menemukan free space;
- (2) entry baru dibuat dalam tabel direktori yang mencatat nama dan lokasi; (3) ukuran yang diinisialisasi 0

■ Write file:

- (1) OS melihat ke direktori untuk mencari lokasinya dalam disk;
- (2) melakukan transfer dari memori ke lokasi dalam disk (suatu pointer digunakan sebagai penunjuk lokasi penulisan berikutnya);
- (3) entry dalam direktori di update

■ Read file:

- OS melakukan hal yang sama dengan penulisan file kecuali operasinya membaca dari lokasi dalam disk ke dalam memori

File System JM -2000/v1.1/14

File Operation (2)

■ Reposition dalam file (seek):

- (1) OS melihat ke direktori untuk mencari entry yang dimaksud, (2) pointer di set dengan harga (lokasi) tertentu yang diberikan

■ Delete file:

- (1) OS melihat ke direktori mencari entry dengan nama yang dimaksud; (2) kemudian membebaskan space yang teralokasi; (3) serta menghapus entry tsb

■ Truncate file:

- Sama dengan menghapus file kecuali entry tidak dihapuskan tapi ukuran file diisi 0; (2) free space tsb.

File System JM -2000/v1.1/15

Others file operation:

■ Operasi-operasi lain pada dasarnya dilakukan dengan kombinasi operasi-operasi dasar tadi, contoh:

- Append file, Rename file
- Get atribut file: mengambil informasi tertentu pada direktori file.
- Set atribut file: mengubah atribut file pada direktori file.

■ Operasi low level (tergantung OS):

- $\text{open}(F_i)$ – search the directory structure on disk for entry F_i , and move the content of entry to memory.
- $\text{close}(F_i)$ – move the content of entry F_i in memory to directory structure on disk.

File System JM -2000/v1.1/16

Example: Unix fopen

```
fd = open(const char *path, int oflag, /* mode_t mode */...);
```

- Membuat hubungan antara "file descriptor" dengan suatu file:
 - Return value merupakan nilai file descriptor untuk file tersebut (bilangan integer).
 - path: berisi string nama file (direktori/file).
 - oflag: flag untuk status open file
 - O_RDONLY Open for reading only.
 - O_WRONLY Open for writing only.
 - O_RDWR Open for reading and writing.
 - O_APPEND If set, the file offset is set to the end of the file prior to each write.
 - O_CREAT Creat file and set user id file to the process user id. ... dst.
- Operasi selanjutnya: `int read(fd, buffer, 128);`

File System JM -2000/v1.1/17

Open: File Table

- OS memelihara suatu tabel mencatat sejumlah informasi file-file yang open:
 - Dikaitkan dengan file descriptor, nilai integer yang menunjuk ke tabel (file handler).
 - Informasi file (atribut) diambil dari direktori dan disalin ke tabel.
 - File pointer: offset pada file yang akan diakses, offset default 0, awal dari isi file.
 - File open count: berapa banyak yang meng-open file tsb

File System JM -2000/v1.1/18

Access method (1)

■ Sequential Access

- Akses dilakukan secara berurut berdasarkan “logical urutan byte” dari file.
- Akses searah sehingga perlu “rewind” jika ingin mengakses data sebelumnya.
- Sequential Access
 - read next*
 - write next*
 - reset (rewind)*
 - read # of records.(no. read after last write).*

File System JM -2000/v1.1/19

Access method (2)

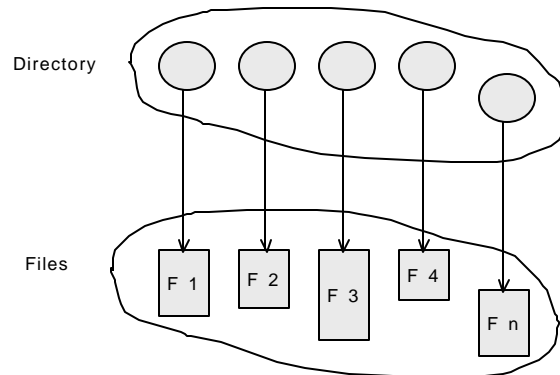
■ Direct Access (random access)

- Akses dilakukan bisa pada posisi mana saja dalam file
 - Direct Access
 - read n*
 - write n*
 - position to n (dikenal istilah: seek)*
 - read next*
 - write next*
 - rewrite n*
- n* = relative block number

File System JM -2000/v1.1/20

Directory Structure

Directory: kumpulan “nodes” (entry) informasi untuk semua files yang berada pada direktori tersebut.



File System JM -2000/v1.1/21

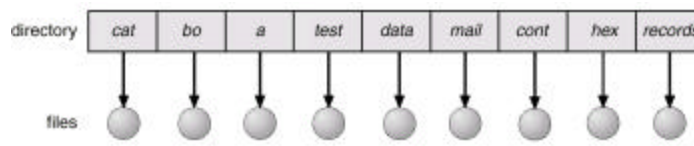
Information in a Directory

- Name
- Type
- Address (lokasi blok)
- Current length
- Maximum length
- Date last accessed
- Date last updated
- Owner ID (user ID)
- Protection information

File System JM -2000/v1.1/22

Single-Level Directory

- A single directory for all users.

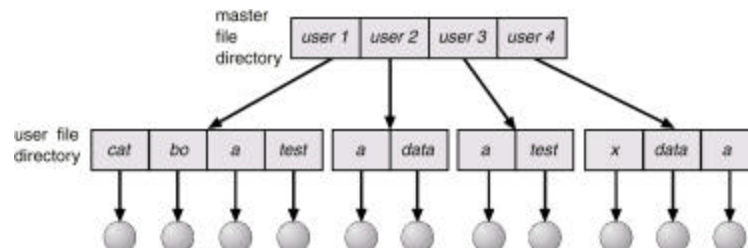


- Naming problem: file dengan nama sama?
- Grouping problem: kumpulan file untuk user/aplikasi tertentu?.

File System JM -2000/v1.1/23

Multi-Level Directory (Tree)

- Separate directory for each user.



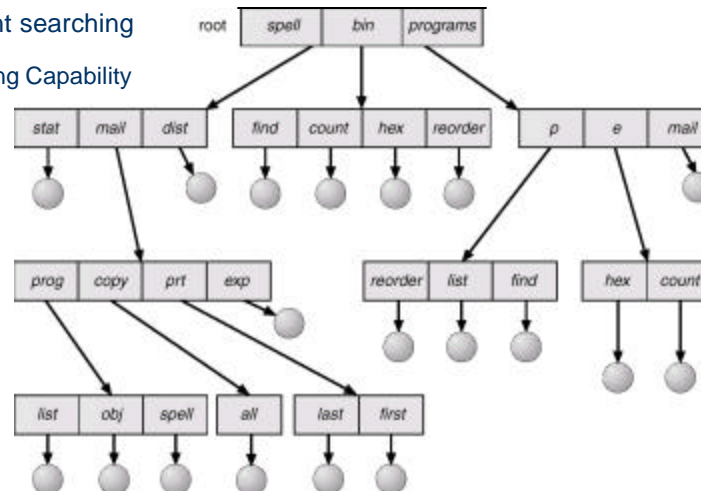
- Path name: pencarian file dari root
- Nama file dapat sama pada direktori (user) berbeda

File System JM -2000/v1.1/24

Tree-Structured Directories

Efficient searching

Grouping Capability



File System JM-2000/v1.1/25

Example: UNIX

```

■ caplin% ls foodir
a.out*      catdir.txt  foonext/    main.c
subdir/     test.c

• caplin% od -c foodir

00000000 \0 023 013 002 \0 \f \0 001 . \0 \0 \0 \0 \0 < 017
00000020 \0 \f \0 002 . . \0 \0 \0 023 ) \t \0 020 \0 006
00000040 s u b d i r \0 \0 \0 023 8 V \0 020 \0 007
00000060 f o o n e x t \0 \0 \0 023 013 016 \0 020 \0 006
00001000 t e s t . c \0 \0 \0 023 013 8 \0 020 \0 006
00001120 m a i n . c \0 \0 \0 023 013 9 \0 020 \0 005
.....
    
```

File System JM-2000/v1.1/26

Protection

- File owner/creator should be able to control:
 - what can be done
 - by whom
- Types of access
 - Read
 - Write
 - Execute
 - Append
 - Delete
 - List

File System JM-2000/v1.1/27

UNIX: protection

- Mode of access: read, write, execute
- Three classes of users

a) owner access	7	⇒	RWX 1 1 1
b) groups access	6	⇒	RWX 1 1 0
c) public access	1	⇒	RWX 0 0 1

- Mode (protection) file tersebut: 761 => atribut dari file.

File System JM-2000/v1.1/28