

```
(c) Copyright Oak Technology
Device name      : MSCD000
Transfer Mode   : Programmed I/O
Drive 0:        : Port= 170 (Secondary Channel), Master  IRQ=15
Firmware Version : BFEE
```

```
MSCDEX Version 2.23
Copyright(c) Microsoft Corp. 1986-1993. All rights reserved.
Drive D: = Driver MSCD000 unit 0
```

# OPERATING SYSTEM

```
C:\>dir
Volume in drive C is MS-DOS_6
Volume Serial Number is 3340-AB44
Directory of C:\

DOS             <DIR>                12/01/98    12:43
COMMAND.COM    54,645  31/05/94    7:22
WINDOWS        <DIR>                13/04/98    0:17
AUTOEXEC.BAT   124    13/04/98    2:21
                220    13/04/98    4:08
```

## IN OUR CLASSROOM



# FILE SYSTEM INTERFACE



PRIO HANDOKO, S.KOM., M.T.I.

## CAPAIAN PEMBELAJARAN

- Mahasiswa memahami konsep *file system*
- Mahasiswa memahami struktur, tipe, dan metode akses berkas yang dikelola oleh sistem operasi

### Agenda.

- File Concept
- Access Methods
- Directory and Disk Structure
- File-System Mounting
- File Sharing
- File Protection

## FILE CONCEPT

- Sistem komputer dapat menyimpan informasi dari beragam media penyimpanan



**MAGNETIC DISKS**



**MAGNETIC TAPES**

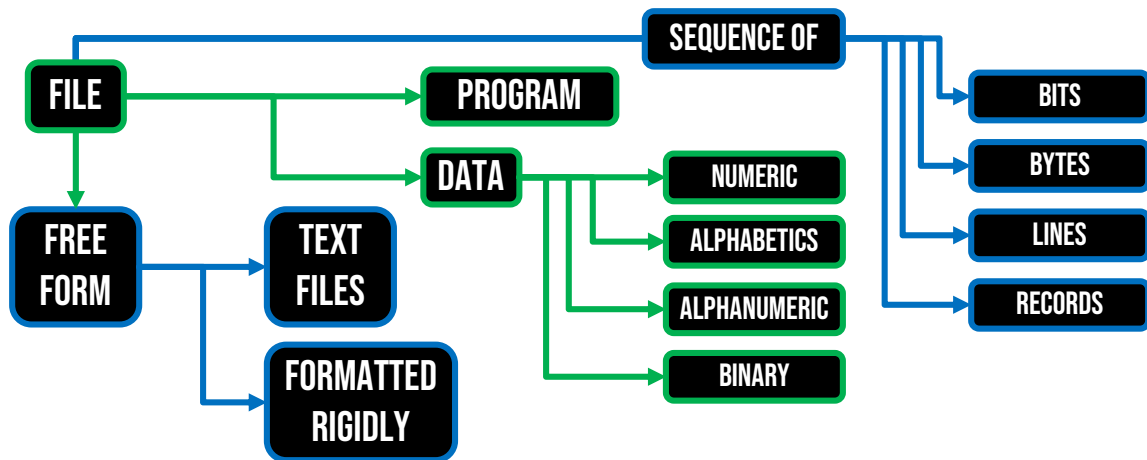


**OPTICAL DISKS**

## FILE CONCEPT

- OS menyediakan *uniform logical view* informasi media penyimpanan
- OS menguraikan properti fisik setiap perangkat penyimpanan untuk mendefinisikan bentuk logis unit penyimpanan yang disebut dengan *file*
- File dipetakan oleh OS ke dalam perangkat fisik (*physical devices*)
- Media penyimpanan pada umumnya bersifat *nonvolatile*

## FILE CONCEPT



7

## FILE CONCEPT

### Struktur File.

Struktur file tergantung pada tipe *file*.

- Text file, berisikan sekumpulan karakter yang berurutan dan diorganisasikan ke dalam sebuah baris
- Source file, berisikan urutan *subroutine* dan fungsi
- Object file, berisikan urutan *byte* yang diorganisasikan ke dalam sebuah blok yang dapat dipahami oleh sistem penghubung
- Executable file, rangkaian kode yang dimuatkan oleh *loader* ke dalam memori dan dieksekusi

8

## FILE ATRIBUTES

**“WHEN A FILE IS NAMED, IT BECOMES INDEPENDENT OF THE PROCESS, THE USER, AND EVEN THE SYSTEM THAT CREATED IT”**

```
-rw-rw-r--   1 pbg  staff   31200  Sep 3 08:30  intro.ps
drwx-----   5 pbg  staff    512    Jul 8 09:33  private/
```

9

## FILE ATRIBUTES

- **Nama file**, informasi yang mewakili isi sebuah file
- **Id file**, informasi yang bersifat unik dan hanya dapat dipahami oleh sistem
- **Tipe file**, informasi mengenai kepemilikan file terhadap sebuah proses atau aplikasi
- **Lokasi file**, informasi yang menunjukkan lokasi file dalam perangkat

10

## FILE ATRIBUTES

- **Ukuran file**, informasi mengenai besar file yang dinyatakan dalam bytes, word atau block
- **Proteksi file**, informasi mengenai mekanisme kontrol untuk melakukan pengaksesan terhadap file
- **Tanggal, waktu & ID Pengguna**, informasi mengenai waktu pembuatan file (*created*), modifikasi file (*modified*) dan waktu penggunaan file

11

## FILE OPERATIONS

**“A FILE IS AN ABSTRACT DATA TYPE. TO DEFINE A FILE PROPERLY, WE NEED TO CONSIDER THE OPERATIONS THAT CAN BE PERFORMED ON FILES”**

Operasi-operasi yang dapat dilakukan terhadap file adalah:

- **CREATE** > SPACE & DIRECTORY

12

## FILE OPERATIONS

- **WRITE** ➤ Membuat SYSTEM CALL (nama file & informasi yang akan dituliskan dalam file) & sistem harus terus menjaga sebuah WRITE POINTER lokasi selanjutnya penulisan file dilakukan dan harus terus terbaharui
- **READ** ➤ Menggunakan SYSTEM CALL (nama file & lokasi file dalam memori) & sistem harus terus menjaga sebuah READ POINTER lokasi selanjutnya pembacaan file dilakukan dan harus terus ter-update

13

## FILE OPERATIONS

- **REPOSITION** ➤ SEARCH lokasi direktori yang sesuai dengan *entry file* & melakukan CURRENT-FILE-POSITION POINTER untuk mereposisi file agar dapat dimuatkan sebuah nilai
- **DELETE** ➤ SEARCH file yang akan dihapus, RELEASE ruang pengisian agar dapat digunakan oleh file lain dan ERASE DIRECTORY ENTRY

14

## FILE OPERATIONS

- ▶ **TRUNCATE** Menyimpan ATTRIBUTE file agar dapat digunakan kembali dibandingkan harus menciptakan lagi file yang baru

Untuk tetap menjaga informasi file-file yang tengah digunakan atau dalam kondisi terbuka, OS menyimpan semua informasi file-file tersebut dalam sebuah OPEN-FILE TABLE dan akan menghapusnya dari tabel ketika file tersebut telah selesai digunakan

15

## FILE TYPES

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing

file type	usual extension	function
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

16



## FILE STRUCTURES

Karena setiap sistem memiliki struktur file yang berbeda-beda, maka harus dibuat sebuah mekanisme yang dapat mengubah struktur file tersebut agar dapat digunakan oleh sistem yang berbeda.

Cara yang dapat dilakukan adalah dengan membuat sebuah *compiler*

17

## INTERNAL FILE STRUCTURES

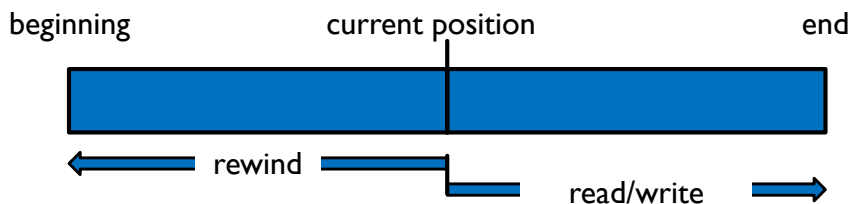
- Menempatkan sebuah *offset* dalam sebuah file menjadi hal yang rumit untuk OS, hal ini dikarenakan ukuran ruang penyimpanan file *physical* media penyimpanan tidak selalu sesuai dengan ukuran ruang kebutuhan *logic* sebuah file.
- Cara yang dapat dilakukan untuk merapikan record-record *logic* ke dalam ukuran ruang penyimpanan file *physical* media penyimpanan adalah dengan melakukan PACKING.

18

## ACCESS METHODS

### 1. Sequential Access

Proses R/W (read/write) dilakukan *record per record* dimulai dari *record* pertama hingga *record* terakhir



19

## ACCESS METHODS

### 2. Direct Access

- File membuat sebuah *fixed logical address* yang memungkinkan program melakukan R/W cepat secara acak
- File diperlakukan sebagai sebuah urutan angka yang menyatakan block atau *record* dan tidak ada batasan urutan dalam melakukan proses R/W
- Bermanfaat dalam proses pengaksesan informasi dalam jumlah besar (contoh: *database*)

20

## ACCESS METHODS

- Operasi *file* harus dimodifikasi menjadi urutan *block* data sebagai parameter pengaksesan
- Nomor urut *block* data/*record* merupakan sebuah *relative block number*

21

## DIRECTORY AND DISK STRUCTURE

OS dapat memanfaatkan seluruh ruang yang dimiliki perangkat penyimpanan sebagai ruang untuk file sistem:

- menggunakan seluruh kapasitasnya
- membaginya menjadi beberapa bagian/partisi untuk memudahkan pengontrolan

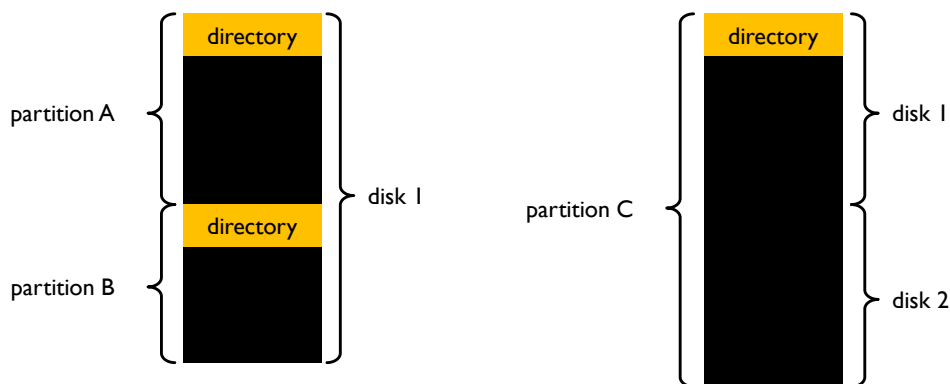
22

## DIRECTORY AND DISK STRUCTURE

- Pemartisian sangat berguna untuk:
  - membatasi ukuran *file* sistem yang berdiri sendiri
  - mencegah dimuatkannya berbagai tipe *file* sistem ke dalam perangkat yang sama
  - mencegah perangkat digunakan oleh user yang berbeda, seperti *swap space* atau partisi yang belum diformat (*unformatted partition/raw partition*)

23

## DIRECTORY AND DISK STRUCTURE



a typical file-system organization

24

# DIRECTORY STRUCTURE

## DIRECTORY and DISK STRUCTURE

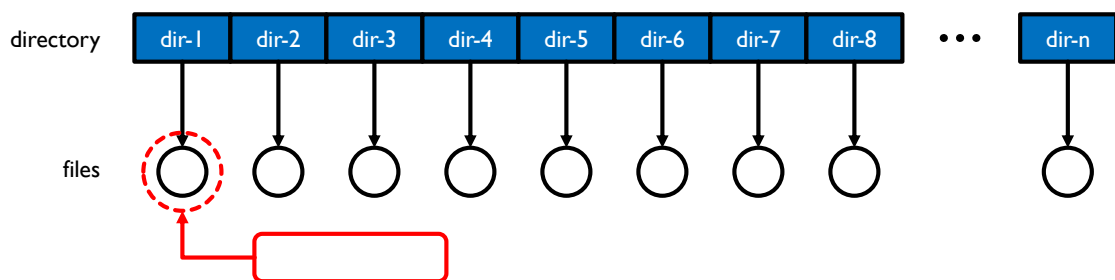
Struktur direktori terbagi menjadi 4 skema yang paling umum digunakan untuk mendefinisikan struktur logika sebuah direktori:

- Single-level directory
- Two-level directory
- Tree-structured directories
- Acyclic-graph directories

25

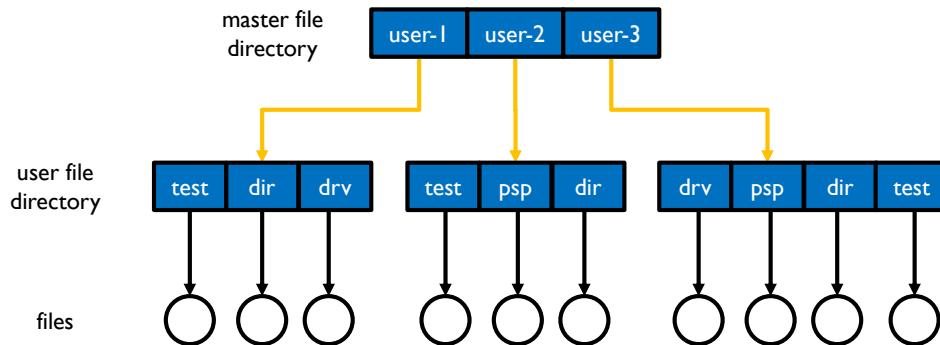
## DIRECTORY STRUCTURE | SINGLE LEVEL DIRECTORY

Struktur direktori dimana semua file berada dalam sebuah direktori sehingga mudah untuk memberikan dukungan.



26

## DIRECTORY STRUCTURE | TWO LEVEL DIRECTORY



27

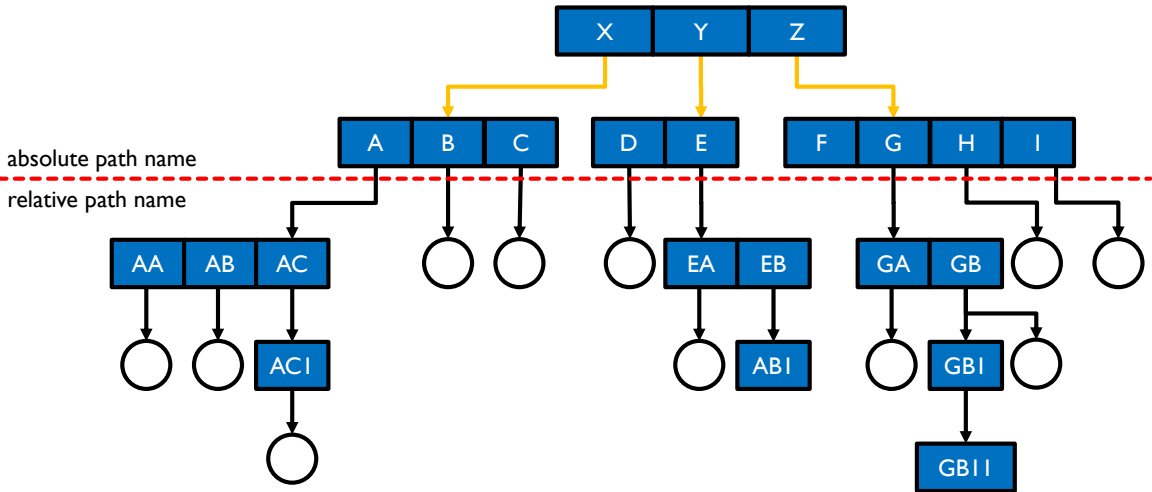
## DIRECTORY STRUCTURE | TWO LEVEL DIRECTORY

### Kekurangan.

- Meskipun struktur ini secara efektif memisahkan antar user satu dengan lainnya, tetapi pemisahan ini akan merugikan jika para *user* menginginkan untuk menyelesaikan sebuah *task* secara bersamaan atau saling melakukan akses terhadap *file* diantara *user*
- Beberapa sistem tidak mengizinkan sebuah *file* milik sebuah *user* diakses oleh *user* lain, karena jika diijinkan, maka *user* harus memiliki kemampuan untuk menamai *file* pada direktori *user* yang berbeda

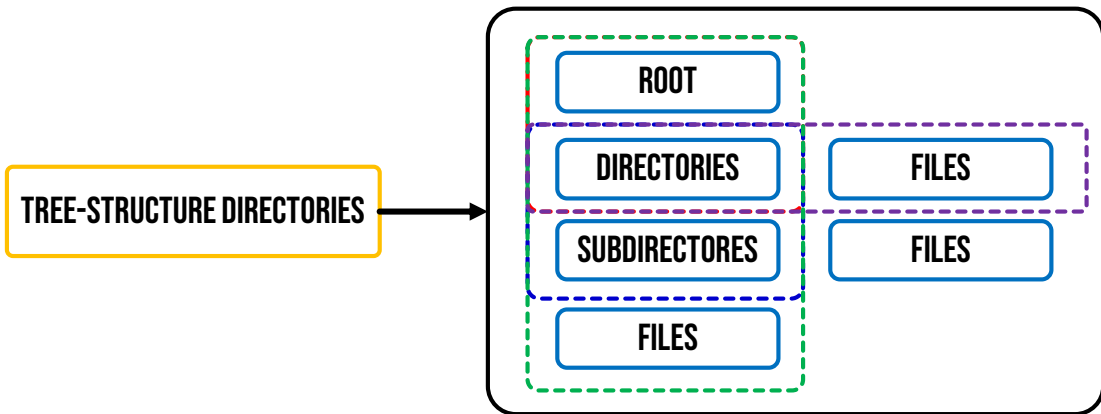
28

# DIRECTORY STRUCTURE | THREE LEVEL DIRECTORY



# DIRECTORY STRUCTURE | THREE LEVEL DIRECTORY

## Tree-Structured Directories.



## DIRECTORY STRUCTURE | THREE LEVEL DIRECTORY

### Directory Management.

#### Operasi Directory.

- Search
- Create
- Delete
- List
- Rename

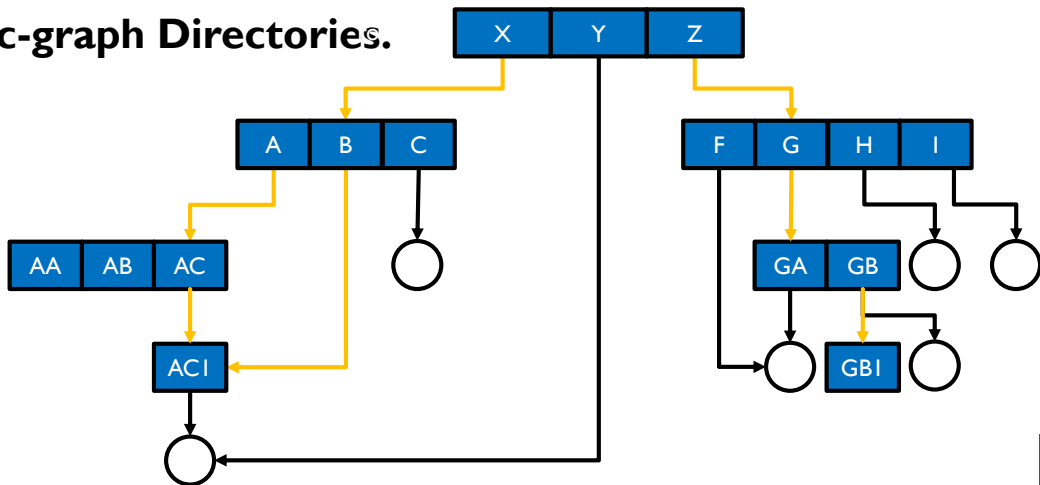


File Directory

31

## DIRECTORY STRUCTURE | THREE LEVEL DIRECTORY

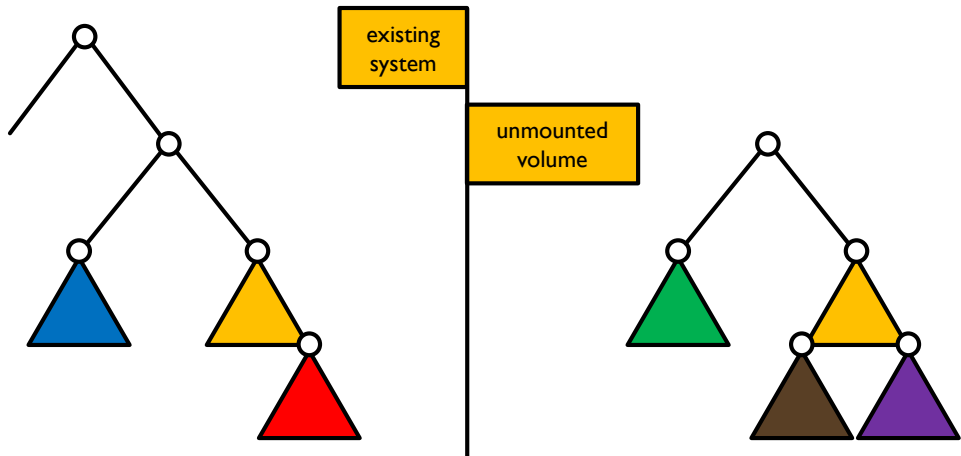
### Acyclic-graph Directories.



32



## FILE-SYSTEM MOUNTING



33

## FILE SHARING

Terbagi menjadi 2:

### I. **Multiple Users.**

- Dibutuhkan lebih banyak atribut file dan direktori
- Beberapa sistem telah berkembang untuk menggunakan konsep kepemilikan file (atau direktori) → *owner* (atau *user*) dan *group*

34

## FILE SHARING

- *Owner* memiliki kemampuan untuk mengubah atribut dan pemberian akses serta menentukan siapa yang lebih berhak terhadap *file*
- ID *owner* dan grup disimpan sebagai atribut sebuah file (atau direktori)

35

## FILE SHARING

### 2. Remote File Systems

- Metode yang digunakan:
  - *ftp* (*file transfer protocol*) → untuk saling bertukar file antar mesin/sistem
  - ***distributed file system*** (DFS) → direktori *remote* terlihat di sistem lokal
  - ***world wide web*** (www) → *ftp* dengan jangkauan yang lebih luas dan menggunakan jenis akses ***anonymous***

36

## FILE PROTECTION

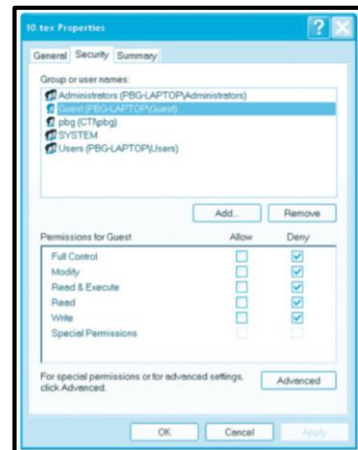
### Alasan:

1. Kebutuhan proteksi berbanding lurus dengan kemampuan akses terhadap file → *controlled access*
2. Membatasi sejumlah tipe file yang dapat melakukan akses file

37

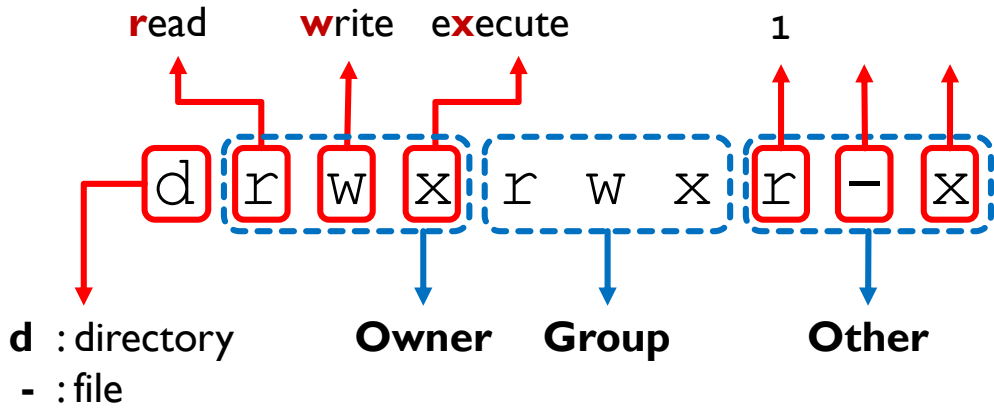
## FILE PROTECTION

3. Setiap user (*owner, group, other*) memiliki tipe akses ke file atau direktori yang berbeda → *access-control list (ACL)*



38

## FILE PROTECTION



39

## FILE SYSTEM INTERFACE



UNTIL NEXT WEEK...