

# CSGE602055 Operating Systems

## CSF2600505 Sistem Operasi

### Week 03: File System & FUSE

C. BinKadal

Sendirian Berhad

<https://docos.vlsm.org/Slides/os03.pdf>

Always check for the latest revision!

REV426: Wed 13 Nov 2024 04:00

# OS242<sup>3</sup>): Operating Systems Schedule 2024 - 2

Week	Topic <sup>1)</sup>	OSC10 <sup>2)</sup>
Week 00	Overview (1), Assignment of Week 00	Ch. 1, 2
Week 01	Overview (2), Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	File System & FUSE	Ch. 13, 14, 15.
Week 04	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	Virtual Memory	Ch. 10.
Week 06	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	Scheduling + W06/W07	Ch. 5.
Week 09	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	I/O & Programming	Ch. 12.

<sup>1)</sup> For schedule, see <https://os.vlsm.org/#idx02>

<sup>2)</sup> Silberschatz et. al.: **Operating System Concepts**, 10<sup>th</sup> Edition, 2018.

<sup>3)</sup> This information will be on **EVERY** page two (2) of this course material.

# STARTING POINT — <https://os.vlsm.org/>

- ☐ **Text Book** — Any recent/decent OS book. Eg. (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10<sup>th</sup> Edition, 2018. (See <https://codex.cs.yale.edu/avi/os-book/OS10/>).
- ☐ **Resources** (<https://os.vlsm.org/#idx03>)
  - ☐ **SCELE** — <https://scele.cs.ui.ac.id/course/view.php?id=3841>.  
The enrollment key is **XXX**.
  - ☐ **Download Slides and Demos from GitHub.com** —  
(<https://github.com/os2xx/docos/>)  
[os00.pdf \(W00\)](#), [os01.pdf \(W01\)](#), [os02.pdf \(W02\)](#), [os03.pdf \(W03\)](#), [os04.pdf \(W04\)](#), [os05.pdf \(W05\)](#),  
[os06.pdf \(W06\)](#), [os07.pdf \(W07\)](#), [os08.pdf \(W08\)](#), [os09.pdf \(W09\)](#), [os10.pdf \(W10\)](#).
  - ☐ **Problems**  
[195.pdf \(W00\)](#), [196.pdf \(W01\)](#), [197.pdf \(W02\)](#), [198.pdf \(W03\)](#), [199.pdf \(W04\)](#), [200.pdf \(W05\)](#),  
[201.pdf \(W06\)](#), [202.pdf \(W07\)](#), [203.pdf \(W08\)](#), [204.pdf \(W09\)](#), [205.pdf \(W10\)](#).
  - ☐ **LFS** — <http://www.linuxfromscratch.org/lfs/view/stable/>
  - ☐ **This is How Me Do It!** — <https://doit.vlsm.org/>
    - ☐ PS: "Me" rhymes better than "I", duh!

# Agenda

- 1 Start
- 2 OS242 Schedule
- 3 Agenda
- 4 Week 03
- 5 OSC10 (Silberschatz) Chapter 13, 14, and 15
- 6 File System Interface
- 7 File System Organization
- 8 FHS: Filesystem Hierarchy Standard
- 9 Devices
- 10 File System Implementation
- 11 File System Internals

# Week 03 File System & FUSE: Topics<sup>1</sup>

- Files: data, metadata, operations, organization, buffering, sequential, nonsequential
- Directories: contents and structure
- File systems: partitioning, mount/unmount, virtual file systems
- Standard implementation techniques
- Memory-mapped files
- Special-purpose file systems
- Naming, searching, access, backups
- Journaling and log-structured file systems

---

<sup>1</sup>Source: ACM IEEE CS Curricula

# Week 03 File System & FUSE: Learning Outcomes<sup>1</sup>

- Describe the choices to be made in designing file systems. [Familiarity]
- Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each. [Usage]
- Summarize how hardware developments have led to changes in the priorities for the design and the management of file systems. [Familiarity]
- Summarize the use of journaling and how log-structured file systems enhance fault tolerance. [Familiarity]

---

<sup>1</sup>Source: ACM IEEE CS Curricula

# OSC10 (Silberschatz) Chapter 13: File-System Interface, Chapter 14: File System Implementation, and Chapter 15: File System Internals

- OSC10 Chapter 13

- File Concept
- Access Methods
- Disk and Directory Structure
- Protection
- Memory-Mapped Files

- OSC10 Chapter 14

- File-System Structure
- File-System Operations
- Directory Implementation
- Allocation Methods
- Free-Space Management
- Efficiency and Performance
- Recovery
- Example: WAFL File System

- OSC10 Chapter 15

- File Systems
- File-System Mounting
- Partitions and Mounting
- File Sharing
- Virtual File Systems
- Remote File Systems
- Consistency Semantics
- NFS

# File System Interface

- File Concept
  - File Attributes: Name, Id, Type, Location, Size, Protection, Time Stamp: create, last modified, last accessed.
  - File Operation
    - Create/Delete/Truncate
    - Open/Close
    - Read/Write
  - File Types: Executable, Object, Source Code, Library, Markup, Markdown, Archive, Compressed.
  - File Structure: No Structure (just a string).
  - Access Methods: Sequential vs Direct Access
- Directory and Disk Structure
  - Three-Structured Directories
  - Directory Operation: create/delete, search/list, rename, traverse
  - Path Name: Absolute vs. Relative
  - FS Mounting vs. Volume Based System
- File Sharing
- Protection: Access Control (eg. -rwx-x-x)



# File System Organization

- Disk Partition
  - One Disk — Many Partitions
  - Many Disks — One Partitions
  - Many Disks — Many Partitions
  - One Partition — One File System (Volume)
- Mounting vs. Volumes

```
demo@badak:~$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda2	9515660	1435776	7573468	16%	/
/dev/sdb1	32895760	12156672	19045036	39%	/usr
/dev/sdc1	412322216	79695252	311639116	21%	/home
udev	10240	0	10240	0%	/dev
tmpfs	16508828	0	16508828	0%	/dev/shm
tmpfs	6603532	8880	6594652	1%	/run
tmpfs	5120	0	5120	0%	/run/lock
tmpfs	16508828	0	16508828	0%	/sys/fs/cgroup
tmpfs	3301768	0	3301768	0%	/run/user/1002

```
demo@badak:~$
```

# FHS: Filesystem Hierarchy Standard

- Source (URL) [http://refspecs.linuxfoundation.org/FHS\\_3.0/fhs-3.0.pdf](http://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf)
- A file placement guidelines/requirements for GNU/Linux-like OS.

FILES	shareable (multiple hosts)	unshareable (single hosts)
static (read only, except for update)	/usr, /opt	/etc, /boot
variable (r/w)	/var/mail, /var/spool/news	/var/run, /var/lock

- The Root File System (Required)

Directory	Description
/bin	Essential command binaries
/boot	Static files of the boot loader
/dev	Device files
/etc	Host-specific system configuration
/lib	Essential shared libraries and kernel modules
/media	Mount point for removable media
/mnt	Mount point for mounting a filesystem temporarily
/opt	Add-on application software packages
/run	Data relevant to running processes
/sbin	Essential system binaries
/srv	Data for services provided by this system
/tmp	Temporary files
/usr	Secondary hierarchy
/var	Variable data

- Specific Options

Directory	Description
/home	User home directories (optional)
/lib<qual>	Alternate format essential shared libraries(optional)
/root	Home directory for the root user (optional)

- The /usr Hierarchy

Directory	Description
/usr/bin	Most user commands (required)
/usr/lib	Libraries (required)
/usr/local	Local hierarchy (empty after main installation) (required)
	/usr/local/{bin etc games include lib man sbin share src} (required)
/usr/sbin	Non-vital system binaries (required)
/usr/share	Architecture-independent data (required)
	/usr/share/{man misc} (required)
	/usr/share/{color dict doc games info locale} (optional)
	/usr/share/{nls ppd sgml terminfo tmac xml zoneinfo} (optional)
/usr/games	Games and educational binaries (optional)
/usr/include	Header files included by C programs (optional)
/usr/libexec	Binaries run by other programs (optional)
/usr/lib<qual>	Alternate Format Libraries (optional)
/usr/src	Source code (optional)

- The /var Hierarchy

Directory	Description
/var/cache	Application cache data (required)
/var/lib	Variable state information (required) /var/lib/misc (required)
/var/local	Variable data for /usr/local (required)
/var/lock	Lock fileslogLog files and directories (required)
/var/opt	Variable data for /opt (required)
/var/run	Data relevant to running processes (required)
/var/spool	Application spool data (required)
/var/tmp	Temporary files preserved between system reboots (required)
/var/backups	(reserved names, do not use)
/var/cron	(reserved names, do not use)
/var/messages	(reserved names, do not use)
/var/preserve	(reserved names, do not use)
/var/account	Process accounting logs (optional)
/var/crash	System crash dumps (optional)
/var/games	Variable game data (optional)
/var/mail	User mailbox files (optional)
/var/yp	Network Information Service (NIS) database files(optional)

- (Mostly) Linux

Directory	Description
/proc	Kernel and process information virtual filesystem
/sys	Kernel and system information virtual filesystem
/usr/include	Header files included by C programs
/usr/src	Source code
/var/spool/cron	cron and at jobs

- the `/dev/` directory
  - `/etc/fstab`: configuration of filesystems
  - `/etc/mtab` → `/proc/mounts`: mounted filesystems
  - `/proc/swaps`: swap filesystems
  - `df`: checking disk space and filesystems
  - Device Major and Minor Numbers
  - UUID - Universally Unique Identifier (128 bits)
  - GUID - Globally Unique Identifiers: `ls -al /dev/disk/by-uuid`
  - practically is NOT guaranteed unique
  - FUSE: Filesystem in Userspace
  - More Storage Structure
    - `tmpfs` — a temporary file storage, stored in RAM that grows and shrinks.
    - `objfs` — dynamic kernel object filesystem.
    - `ctfs` — (creating, controlling, and observing) contract file system .
    - `loopfs` — loop filesystem allows to dynamically allocate loop devices.
    - `procfs` — proc filesystem presents information about processes.
    - `ufs` — the original Unix Filesystem (before Linux ext2).
    - `zfs` — the Zettabyte Filesystem is both a volume manager and a file system.

# A Typical Ubuntu 20.04 Work Station

```
cbkadal@ubuntu2004:~$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	8138664	0	8138664	0%	/dev
tmpfs	1634140	1948	1632192	1%	/run
tmpfs	8170684	210348	7960336	3%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	8170684	0	8170684	0%	/sys/fs/cgroup
tmpfs	1634136	76	1634060	1%	/run/user/1000
/dev/sda1	98304	33523	64781	35%	/boot/efi
/dev/sda3	286082372	78565916	207516456	28%	/altfs/ntfs
/dev/sda5	32999120	9181772	22111364	30%	/altfs/linux1
/dev/sda6	38186548	12054612	24162428	34%	/altfs/linux2
/dev/sda7	126265680	13342928	106465768	12%	/
/dev/sdb2	62216964	13238156	45788588	23%	/var
/dev/sdb3	3532259904	2605226568	747535200	78%	/home
/dev/loop0	101632	101632	0	100%	/snap/core/10859
/dev/loop1	65920	65920	0	100%	/snap/gtk-common-themes/1513
/dev/loop2	66432	66432	0	100%	/snap/gtk-common-themes/1514
/dev/loop3	678016	678016	0	100%	/snap/intellij-idea-community/273
/dev/loop4	679040	679040	0	100%	/snap/intellij-idea-community/270
/dev/loop5	52352	52352	0	100%	/snap/snap-store/498
/dev/loop6	223232	223232	0	100%	/snap/gnome-3-34-1804/60
/dev/loop7	267008	267008	0	100%	/snap/kde-frameworks-5-core18/32
/dev/loop8	166784	166784	0	100%	/snap/gnome-3-28-1804/145
/dev/loop9	102784	102784	0	100%	/snap/kotlin/57
/dev/loop10	52352	52352	0	100%	/snap/snap-store/518
/dev/loop11	56832	56832	0	100%	/snap/core18/1988
#####	##### TL;DR #####		# #####		
/dev/loop18	56832	56832	0	100%	/snap/core18/1944
/dev/loop19	142080	142080	0	100%	/snap/chromium/1506

# File Systems Implementation

- File System Layers / Structure
  - Application Programs
  - Logical File Systems
  - File-Organization Module
  - Basic File Systems
  - I/O Control
  - Hardware Device
- File System Implementation
- File Control Block
- FS In Memory Structure
- VFS: Virtual File Systems
  - How to support multiple File Systems
  - I.e. How to support multiple `open()/close()` `read()/write()` operations



# Implementation and Allocation Method

- Directory Implementation
  - Linear List
  - Hash Table
- Allocation Method
  - Contiguous
  - Linked
  - Indexed
  - Combined Scheme
- Free Space Management
- Performance & Efficiency
- Unified Buffer Cache
- Recovery
- Log Structured File System

- File Systems
- File-System Mounting
- Partitions and Mounting
- File Sharing
- Virtual File Systems
- Remote File Systems
- Consistency Semantics
- NFS