

CSGE602055 Operating Systems

CSF2600505 Sistem Operasi

Week 03: File System & FUSE

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

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

Always check for the latest revision!

REV337 30-Aug-2021

OS212³): Operating Systems 2021 - 2

OS A	OS B	OS C	OS INT
Every first day of the Week, Quiz#1 and Quiz#2 : 07:15-08:00			
Monday/Thursday 13:00 — 14:40	Monday/Thursday 15:00 - 16:40	Monday/Thursday 13:00 — 14:40	Monday/Wednesday 08:00 — 09:40

Week	Schedule & Deadline ¹⁾	Topic	OSC10 ²⁾
Week 00	30 Aug - 05 Sep 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	06 Sep - 12 Sep 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	13 Sep - 19 Sep 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	20 Sep - 26 Sep 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	27 Sep - 03 Oct 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	04 Oct - 10 Oct 2021	Virtual Memory	Ch. 10.
Week 06	11 Oct - 17 Oct 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	01 Nov - 07 Nov 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	08 Nov - 14 Nov 2021	Scheduling + W06/W07	Ch. 5.
Week 09	15 Nov - 21 Nov 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	22 Nov - 28 Nov 2021	I/O & Programming	Ch. 12.

¹⁾ The **DEADLINE** of Week 00 is 05 Sep 2021, whereas the **DEADLINE** of Week 01 is 12 Sep 2021, and so on...

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

³⁾ 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**, 10th Edition, 2018. See also <https://www.os-book.com/OS10/>.
- ❑ **Resources**
 - ❑ **SCELE OS212** — <https://scele.cs.ui.ac.id/course/view.php?id=3268>.
The enrollment key is **XXX**.
 - ❑ **Download Slides and Demos from GitHub.com**
<https://github.com/UI-FASILKOM-OS/SistemOperasi/>:
[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/>
 - ❑ **OSP4DISS** — <https://osp4diss.vlsm.org/>
 - ❑ **DOIT** — <https://doit.vlsm.org/001.html>

Agenda

- 1 Start
- 2 Schedule
- 3 Agenda
- 4 Week 03
- 5 File System Interface
- 6 File System Organization
- 7 FHS: Filesystem Hierarchy Standard
- 8 Devices
- 9 File System Implementation
- 10 File System Internals
- 11 Week 03: Check List
- 12 The End

Week 03 File System & FUSE: Topics¹

- 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

¹Source: ACM IEEE CS Curricula 2013

Week 03 File System & FUSE: Learning Outcomes¹

- 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]

¹Source: ACM IEEE CS Curricula 2013

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
- 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/msgs	(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
 - BBFS: Big Brother File System
- More Storage Structure
 - `tmpfs`
 - `objfs`
 - `ctfs`
 - `lofs`
 - `procfs`
 - `ufs`
 - `zfs`

A Typical Ubuntu 20.04 Work Station

```
rms46@pamulang1:~$ 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
/dev/loop12	33152	33152	0	100%	/snap/snapd/11107
/dev/loop13	100736	100736	0	100%	/snap/core/10823
#####	##### 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

Week 03: Check List (Deadline: 26 Sep 2021).

- ☐ Week 03 Token: **OS212W03**
- ☐ This page is <https://os.vlsm.org/Slides/check03.pdf>.
- ☐ More details: <https://osp4diss.vlsm.org/W03.html>.
- ☐ Assignment Check List:
 - ① Read OSC-10 (chapter 13 + chapter 14 + chapter 15)
 - ② Try Demos Week 02 and Week 03.
 - ③ Check if your ".bash_aliases" file is up-to-date. (See [OSP4DISS](#)).
 - ④ Visit <https://os.vlsm.org/GitHubPages/>. Review **Last Week TOP 10 List** and pick at least 3 out of your 10 next neighbors.
 - ⑤ Create your **TOP 10 List** of Week 03 (See <https://cbkadal.github.io/os212/W03/>). Do not use lecture material. Please be more creative!
 - ⑥ Update your log (e.g. <https://cbkadal.github.io/os212/TXT/mylog.txt>).
 - ⑦ Download <https://os.vlsm.org/WEEK/W03.tar.bz2.asc>. The passphrase will follow. The result ("W03-FUSE.txt") should be placed into a "W03/" folder and tarballed as "myW03.tar.bz2.asc"
 - ⑧ Update bash script (e.g. <https://cbkadal.github.io/os212/TXT/myscript.sh>).
 - ⑨ Make [SHA256SUM](#) and sign it (detached, armor) as [SHA256SUM.asc](#).

The End

- ☐ This is the end of the presentation.
- ☒ This is the end of the presentation.
 - This is the end of the presentation.