

CSCE604227 System Programming
CSCE604227 Pemrograman Sistem
Week 03: FUSE: Filesystem in Userspace

C. BinKadal

Sendirian Berhad

<https://docs.vlsm.org/SPSlides/sp03.pdf>

Always check for the latest revision!

REV025: Wed 24 Jul 2024 19:00

SP241¹): System Programming

Week	Topic
Week 00	Overview
Week 01	Linux Kernel and Programming Interface
Week 02	Revisit Linux From Scratch
Week 03	FUSE: Filesystem in Userspace
Week 04	GetOpt
Week 05	Autoconf and Automake
Week 06	Boxing/Unboxing
Week 07	Sync, SETUID, and MMAP
Week 08	Kernel Modules I
Week 09	Kernel Modules II
Week 10	Kernel Modules III

¹) This information will be on **EVERY** page two (2) of this course material.

- Text Book** — The Linux Programming Interface, 2010, No Starch Press, ISBN 978-1-59327-220-3 — <https://man7.org/tlpi/>.
- Resources**
 - SCELE** — <https://scele.cs.ui.ac.id/course/view.php?id=3742>.
The enrollment key is **XXX**.
 - Download Slides and Demos from GitHub.com**
<https://github.com/os2xx/docOS/>:
[sp00.pdf \(W00\)](#), [sp01.pdf \(W01\)](#), [sp02.pdf \(W02\)](#), [sp03.pdf \(W03\)](#),
[sp04.pdf \(W04\)](#), [sp05.pdf \(W05\)](#), [sp06.pdf \(W06\)](#), [sp07.pdf \(W07\)](#),
[sp08.pdf \(W08\)](#), [sp09.pdf \(W09\)](#), [sp10.pdf \(W10\)](#).
 - LFS** — <http://www.linuxfromscratch.org/lfs/view/stable/>
 - OSP4DISS** — <https://osp4diss.vlsm.org/>
 - This is How Me DO IT!** — <https://doit.vlsm.org/>
 - PS: "Me" rhymes better than "I" duh!

Agenda

- 1 Start
- 2 Schedule
- 3 Agenda
- 4 File System Interface
- 5 File System Organization
- 6 FHS: Filesystem Hierarchy Standard
- 7 Devices
- 8 File System Implementation
- 9 File System Internals
- 10 One More Time

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/msgspgs	(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

```
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
#####	##### TL;DR #####			# #####	
/dev/loop18	56832	56832	0	100%	/snap/core18/1944
/dev/loop19	142080	142080	0	100%	/snap/chromium/1506

- 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

- Example: long options with `getopt_long()`
 - TLPI Source Code <https://man7.org/tlpi/code/>
 - URL <https://osp4diss.vlsm.org/osp-122.html>
- Example: `autoconf` — A Small Hello World
 - GitHub <https://github.com/UI-FASILKOM-OS/sharesp241>
 - URL <https://osp4diss.vlsm.org/osp-123.html>