## CSGE602055 Operating Systems CSF2600505 Sistem Operasi Week 09: Storage, BIOS, Loader, & Systemd

#### Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

https://os.vlsm.org/
Always check for the latest revision!

REV188 06-Feb-2019

## Operating Systems 2019-1

A (Rm 3114) [Tu/Th 10-12] — B (Rm 3114) [Tu/Th 13-15] — C (Rm 3114) [Tu/Th 16-18] — D (Rm 2401) [Tu/Th 10-12] — E (Rm 2306) [Tu/Th 13-15]

Week	Schedule	Topic	OSC10
Week 00	07 Feb - 13 Feb 2019	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	14 Feb - 20 Feb 2019	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	21 Feb - 27 Feb 2019	Security, Protection, Privacy,	Ch. 16, 17
		& C-language	
Week 03	28 Feb - 06 Mar 2019	File System & FUSE	Ch. 13, 14, 15
Week 04	12 Mar - 18 Mar 2019	Addressing, Shared Lib, & Pointer	Ch. 9
Week 05	19 Mar - 25 Mar 2019	Virtual Memory	Ch. 10
Mid-Term	23-30 Mar 2019 (tba)	MidTerm (UTS)	
Week 06	02 Apr - 08 Apr 2019	Concurency: Processes & Threads	Ch. 3, 4
Week 07	09 Apr - 15 Apr 2019	Synchronization & Deadlock	Ch. 6, 7, 8
Week 08	16 Apr - 22 Apr 2019	Scheduling	Ch. 5
Week 09	23 Apr - 29 Apr 2019	Storage, BIOS, Loader, & Systemd	Ch. 11
Week 10	30 Apr - 06 May 2019	I/O & Programming	Ch. 12
Reserved	07 May - 17 May 2019		
Final	18-25 May 2019 (tba)	Final (UAS)	This schedule is
Extra	27 Jun 2019	Extra assignment confirmation	subject to change.

## The Weekly Check List

Resources: https://os.vlsm.org/			
☐ (THIS) Slides — https:			
//github.com/UI-FASILKOM-OS/SistemOperasi/tree/master/pdf/			
☐ <b>Demos</b> — https://github.com/UI-FASILKOM-OS/SistemOperatree/master/demos/			
			<ul><li>□ Extra — BADAK.cs.ui.ac.id:///extra/</li><li>□ Problems — https://rms46.vlsm.org/2/:</li></ul>
195.pdf (Week 00), 196.pdf (Week 01), 197.pdf (Week 02),			
198.pdf (Week 03), 199.pdf (Week 04), 200.pdf (Week 05),			
201.pdf (Week 06), 202.pdf (Week 07), 203.pdf (Week 08),			
204.pdf (Week 09), 205.pdf (Week 10).			
☐ <b>Text Book</b> : any recent/decent OS book. Eg. <b>(OSC10)</b> Silberschatz			
et. al.: <b>Operating System Concepts</b> , 10 <sup>th</sup> Edition, 2018.			
☐ Encode your <b>QRC</b> with size upto 7cm x 7cm (ca. 400x400 pixels):			
"OS182 CLASS ID SSO-ACCOUNT Your-Full-Name"			
☐ Write your Memo (with QRC) <b>every week</b> .			
☐ Login to badak.cs.ui.ac.id via kawung.cs.ui.ac.id for at least			
10 minutes every week. Copy the weekly demo files to your own home			
directory.			
Fg. (Week00): cp -r /extra/Week00/W00-demos/ W00-demos/			

## Agenda

- Start
- Schedule
- Agenda
- 4 Week 09
- 5 Week 09: Storage, BIOS, Loader, & Systemd
- Storage Management
- RAID
- 8 Legacy BIOS
- UEFI
- Operating System (Boot) Loader
- GRUB Map
- init (SYSV legacy)
- UpStart Ubuntu
- The All New "systemd"
- systemctl
- 16 The End

## Week 09 Storage, BIOS, Loader, & Systemd: Topics<sup>1</sup>

- Storage
- Storage Arrays
- BIOS
- Loader
- Systemd

<sup>&</sup>lt;sup>1</sup>Source: ACM IEEE CS Curricula 2013

# Week 09 Storage, BIOS, Loader, & Systemd: Learning Outcomes<sup>1</sup>

- Storage [Usage]
- Storage Arrays [Usage]
- BIOS [Usage]
- Loader [Usage]
- Systemd [Usage]

<sup>&</sup>lt;sup>1</sup>Source: ACM IEEE CS Curricula 2013

## Week 09: Storage, BIOS, Loader, & Systemd

- Reference: (OSC10-ch11)
- Mass-Storage Structure
  - Obsolete: Magnetic Tape, Disket
  - Until When?: Magnetic Disk, DVD
  - Until When?: Mechanical Disk Arm Scheduling
  - Solid-State Disks (SSD)
  - (What is a) Flash Disk
- Attached-Storage
  - ullet Host-Attached Storage: via I/O
  - Network-Attached Storage (NAS): via distributed FS
  - Storage Area Network (SAN): dedicated Network

## Storage Management

- Formating
  - Low Level (Physical)
  - High Level (FS)
- Boot Block
- Disk Partition
  - "MBR"-scheme
    - upto 4 primary partition
    - upto 2 TB disk
  - "GPT"-scheme
    - "unlimited" partition
    - "unlimited" disk
    - redundancy
- Swap Space Management: On Partition or FS?

## RAID: Redundant Array of In\* Disks

- RAID 0, 1, 5, 6, 10, 100
- Note (http://www.commodore.ca/windows/raid5/raid5.htm):
  - RAID was created to enhance data performance, reliability and availability.
  - Striping, parity checking and mirroring are three primary functions of RAID systems.
  - RAID performs its functions transparent to the operating system.
  - Systems are typically defined by ranks consisting of five disks each connected to one or two Disk Array Controllers.
  - Different RAID levels provide varying degrees of speed and data protection.
- Problems with RAID
- Stable-Storage Implementation

## BIOS, Boot, & Systemd

- Firmware
  - BIOS: Basic Input Output System.
  - UEFI: Unified Extensible Firmware Interface.
  - ACPI: Advanced Configuration and Power Interface.
- Operating System (Boot) Loader
  - BOOTMGT: Windows Bootmanager / Bootloader.
  - LILO: Linux Loader.
  - GRUB: GRand Unified Bootloader.
- Operating System Initialization
  - Init (legacy)
  - UpStart
  - Systemd

## Legacy BIOS

- Check Settings.
- Initialize CPU & RAM.
- POST: Power-On Self-Test.
- Initialize ports, LANS, etc.
- Load a Boot Loader.
- Handover to the Boot Loader.
- Provides "Native" (obsolete) Drivers only (not loadable).
- Provides "INT" services .
- Limitation.
  - Technology of 1970s.
  - 16 bits software.
  - 20 bits address space (1 MB).
  - 31 bits disk space (2 TB).

#### **BIOS**

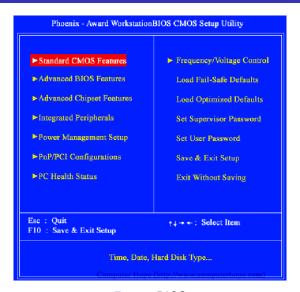


Figure: BIOS

#### **UEFI**

- A Firmware Specification, not an Implementation!
- No (INT) service after boot.
- HII: Human Interface Infrastructure.
- Protected Mode.
- Flexible.
  - Technology of 2000s.
  - writen in C.
  - (third party) loadable drivers and tools.
  - Emulate Legacy BIOS transition (MBR block, INT service).
  - UEFI Shell: environment shell for diagnostic (no need for DOS).
- Problems
  - Who controls the Hardware?
  - Is "Secure Boot" a good thing?
  - How about a NASTY/LOCKING/TROJAN UEFI implementation?
  - Different DRIVERS.

#### **UEFI**



Figure: UEFI

#### **UEFI** Boot

## Platform Initialization (PI) Boot Phases



Figure: UEFI Boot Process<sup>1</sup>.

## Operating System (Boot) Loader

- General
  - How/Where to start the operating system?
  - What to do?
  - How many ways to boot?
  - How many types of OS?
- Disk Partition
  - MBR: Master Boot Record (1983).
  - GPT: GUID (Globally Unique Identifiers) Partition Table (2010s).
- GRUB: GRand Unified Boot system
  - Stage 1: a small boot.img inside the MBR.
  - Stage 1.5 (core.img): FS drivers after MBR.
  - Stage 2: Kernel Selection: Windows, Linux, BSD, etc.
- GRUB2
  - More flexible than GRUB legacy.
  - More automated than GRUB legacy.
  - Accept MBR and GPT.
  - Stage 1.5 (core.img): generated from diskboot.img.
  - No 1024 cylinder restriction.

## **GRUB Map**

#### **GNU GRUB 2**

Locations of boot.img, core.img and the /boot/grub directory

Example 1: an MBR-partitioned harddisc with sector size of 512 or 4096Bytes



Example 2: a GPT-partitioned harddisc with sector size of 512 or 4096Bytes

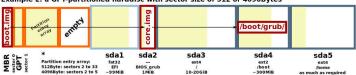


Figure: GRUB<sup>1</sup>.

<sup>&</sup>lt;sup>1</sup>Source Shmuel Csaba Otto Traian 2013

## init (SYSV legacy)

- File: /etc/inittab.
- Folders: /etc/rcX.d X = runlevel.
  - Seven (7) different runlevels:
    - 0 (shutdown).
    - 1 (single-user/admin).
    - 2 (multi-user non net).
    - 3 (standard).
    - 4 (N/A).
    - 5 (3+GUI).
    - 6 (reboot).
  - SXX-YYY: Start
  - KXX-YYY: Kill.
- One script at a time in order.
- dependency is set manually.

## UpStart - Ubuntu

- Developer: Ubuntu.
- Folder: /etc/init/.
- Control: initctl.
  - initctl list listing all processes managed by upstart.
- better support for hotplug devices.
- cleaner service management.
- faster service management.
- asynchronous.

## The All New "systemd"

- Replaces (SYSV) init and UpStart.
  - better concurency handling: Faster!
  - better dependencies handling: No more "S(tarts)" and "K(ills)".
  - better crash handling: automatic restart option.
  - better security: group protection from anyone including superusers.
  - simpler config files: reliable and clean scripts.
  - hotplug: dynamic start/stop.
  - supports legacy systems (init).
  - overhead reducing.
  - unified management way for all distros.
  - bloated: doing more with more resources.
  - linux specific: NOT portable.

### systemctl

```
for II in
   'systemctl list-unit-files | head -8; echo "(...)";
       systemctl list-unit-files| tail -8' \
   'systemd-analyze blame | wc -1; echo "===";
       systemd-analyze blame | head -15' \
   'systemctl --full | wc -1; echo "===";
       systemctl --full | head -10' \
   'systemctl list-units | wc -1; echo "===";
       systemctl list-units | head -10' \
   'systemctl list-units |grep .service|wc -l;echo "===";
       systemctl list-units|grep .service|head -10' \
   'systemctl list-units | grep ssh.service' \
   'systemctl status ssh.service' \
   'systemctl is-enabled ssh' \
   'journalctl' \
   'journalctl -b' \
dο
```

#### The End

- $\square$  This is the end of the presentation.
- imes This is the end of the presentation.
- This is the end of the presentation.