

CSGE602055 Operating Systems

CSF2600505 Sistem Operasi

Week 09: DISK, I/O, BIOS, Loader, & Systemd

Rahmat M. Samik-Ibrahim

University of Indonesia

<https://os.vlsm.org/>

Always check for the latest revision!

REV167 18-Oct-2018

Operating Systems 2018-2 (Room 3114)

R/M (Tu/Th 13-15) | I (Tu/Th 15-17)

Week	Schedule	Topic	OSC10
Week 00	04 Sep - 12 Sep 2018	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	13 Sep - 19 Sep 2018	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	20 Sep - 26 Sep 2018	Security, Protection, Privacy, & C-language	Ch. 16, 17
Week 03	27 Sep - 03 Oct 2018	File System & FUSE	Ch. 13, 14, 15
Week 04	04 Oct - 10 Oct 2018	Addressing, Shared Lib, & Pointer	Ch. 9
Week 05	11 Oct - 17 Oct 2018	Virtual Memory	Ch. 10
Reserved	18 Oct - 19 Oct 2018		
Mid-Term	24 Oct 2018	MidTerm (UTS): 09:00 - 11:30	
Week 06	30 Oct - 05 Nov 2018	Concurrency: Processes & Threads	Ch. 3, 4
Week 07	06 Nov - 12 Nov 2018	Synchronization & Deadlock	Ch. 6, 7, 8
Week 08	13 Nov - 21 Nov 2018	Scheduling	Ch. 5
Week 09	22 Nov - 28 Nov 2018	Disks, BIOS, Loader, & Systemd	Ch. 11
Week 10	29 Nov - 05 Dec 2018	I/O & Programming	Ch. 12
Reserved	06 Dec - 14 Dec 2018		
Final Extra	15 Dec - 22 Dec 2018 12 Jan 2019	Final (UAS): TBA Extra assignment	This schedule is subject to change.

The Weekly Check List

- ☐ **Resources:** <https://os.vlsm.org/>
 - ☐ **(THIS) Slides** — <https://github.com/UI-FASILKOM-OS/SistemOperasi/tree/master/pdf/>
 - ☐ **Demos** — <https://github.com/UI-FASILKOM-OS/SistemOperasi/tree/master/demos/>
 - ☐ **Extra** — BADAK.cs.ui.ac.id:///extra/
 - ☐ **Problems** — rms46.vlsm.org/2/195.pdf, [196.pdf](http://rms46.vlsm.org/2/196.pdf), ..., [205.pdf](http://rms46.vlsm.org/2/205.pdf)
- ☐ **Text Book:** any recent/decent OS book. Eg. **(OSC10)** Silberschatz et. al.: **Operating System Concepts**, 10th Edition, 2018.
- ☐ Encode your **QRC** with size upto 7cm x 7cm (ca. 400x400 pixels):
"OS182 CLASS ID SSO-ACCOUNT Your-Full-Name"
- ☐ For **Week 00**, send your **embedded QRC before the 2nd lecture**
<mailto:operatingsystems@vlsm.org>
With Subject: OS182 CLASS ID SSO-ACCOUNT Your-Full-Name
- ☐ Write your Memo (with QRC) **every week**.
- ☐ 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.
Eg. (Week00): `cp -r /extra/Week00/W00-demos/ W00-demos/`

Agenda

- 1 Start
- 2 Schedule
- 3 Agenda
- 4 Week 09
- 5 Week 09: DISK, I/O, BIOS, Boot, & Systemd
- 6 Disk Management
- 7 RAID
- 8 I/O
- 9 Legacy BIOS
- 10 UEFI
- 11 UEFI Boot
- 12 Operating System (Boot) Loader
- 13 GRUB Map
- 14 init (SYSV legacy)
- 15 UpStart - Ubuntu
- 16 The All New "systemd"
- 17 systemctl

- Characteristics of serial and parallel devices
- Abstracting device differences
- Buffering strategies
- Direct memory access
- Recovery from failures

¹Source: ACM IEEE CS Curricula 2013

Week 09 I/O: Learning Outcomes¹

- Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate. [Familiarity]
- Identify the relationship between the physical hardware and the virtual devices maintained by the operating system. [Usage]
- Explain buffering and describe strategies for implementing it. [Familiarity]
- Differentiate the mechanisms used in interfacing a range of devices (including hand-held devices, networks, multimedia) to a computer and explain the implications of these for the design of an operating system. [Usage]
- Describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted. [Usage]
- Identify the requirements for failure recovery. [Familiarity]
- Implement a simple device driver for a range of possible devices. [Usage]

¹Source: ACM IEEE CS Curricula 2013

- 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
 - Host-Attached Storage: via I/O
 - Network-Attached Storage (NAS): via distributed FS
 - Storage Area Network (SAN): dedicated Network
- Legacy Linux I/O Scheduling Algorithm.
 - Deadline Scheduler
 - Completely Fair Queueing (CFQ)

- Formatting
 - 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

Week 03: I/O, BIOS, Boot, & Systemd

- Reference: (OSC9-ch13 demo-w03)
- Overview
- I/O Hardware
- Application I/O Interface
- Kernel I/O Subsystem
- Transforming I/O Requests to Hardware Operations
- STREAMS
- BIOS
- Boot
- Systemd

I/O (1)

- Direct I/O vs. Memory Mapped I/O
- Interrupts: Non Maskable (NMI) vs Maskable (MI)
- DMA: Direct Memory Access
- I/O Structure:
 - Kernel (S/W).
 - I/O (S/W: Kernel Subsystem)
 - Driver (S/W)
 - Controller (H/W)
 - Device (H/W)
- I/O Streams
 - APP
 - HEAD
 - MODULES
 - DRIVER
 - H/W.

- I/O Interface Dimensions
 - Character-stream vs. Block;
 - Sequential vs. Random-access;
 - Sharable vs. Dedicated;
 - Parallel vs. Serial;
 - Speed;
 - Read Write – Read Only – Write Only.
 - Synchronous vs. Asynchronous;
 - Blocking vs. Non-Blocking.
- Where should a new algorithm be implemented?
 - APP?
 - Kenel?
 - Driver?
 - Controller?
 - HW?

BIOS, Boot, & Systemd

- Reference: (OSC9-ch13 demo-w03)
- 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
- I/O
 - Interrupt.
 - DMA.
 - ETC.

- 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).

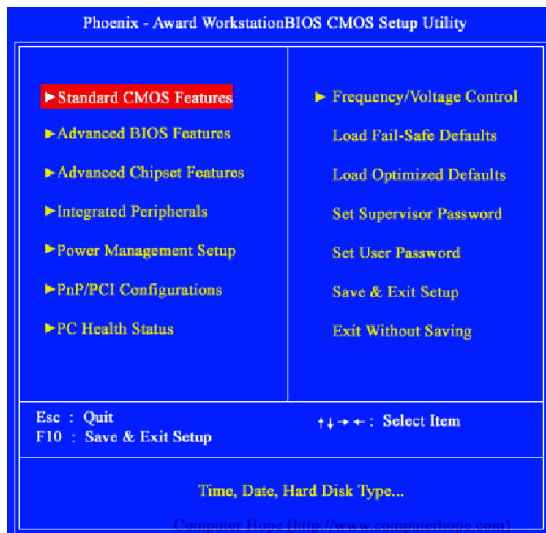


Figure: BIOS

- A Firmware Specification, not an Implementation!
- No (INT) service after boot.
- HII: Human Interface Infrastructure.
- Protected Mode.
- Flexible.
 - Technology of 2000s.
 - written 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**.



Figure: UEFI

Platform Initialization (PI) Boot Phases

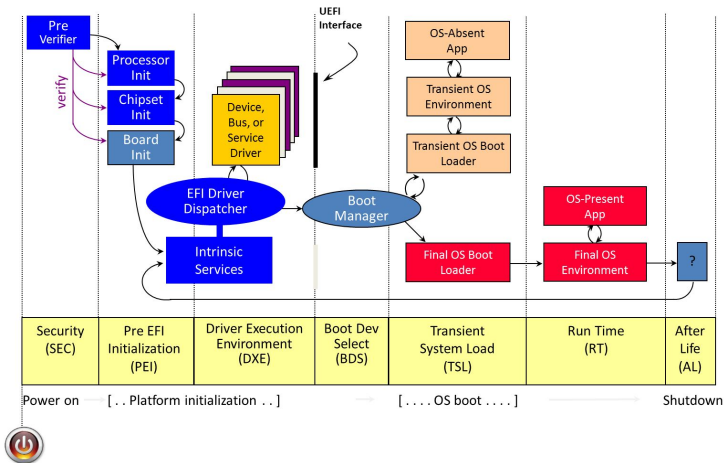


Figure: UEFI Boot Process¹.

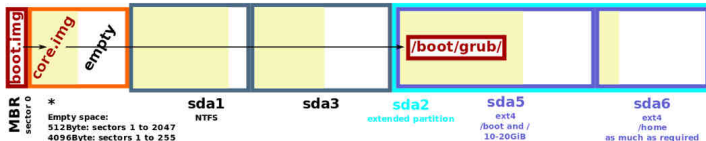
Operating System (Boot) Loader

- General
 - How/Where to start the operating system?
 - What to do?
 - How many ways to boot?
 - How many types of OS?
- GRUB/GRUB2: GRand Unified Boot system
 - Stage 1 (boot.img): MBR (Master Boot Record) – Where is everything
 - Stage 1.5 (core.img): generated from diskboot.img
 - Stage 2: Kernel Selection: Windows, Linux, BSD, etc.
- GRUB2
 - More flexible than GRUB legacy
 - More automated than GRUB legacy
- Disk Partition
 - MBR: Master Boot Record (1983).
 - GPT: GUID Partition Table (2010s).

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

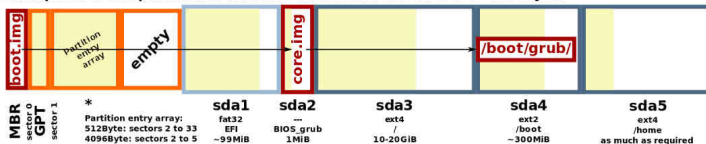


Figure: GRUB¹.

¹Source 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.

- 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 concurrency 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 -l; echo "===";
  systemd-analyze blame | head -15' \
'systemctl --full | wc -l; echo "===";
  systemctl --full | head -10' \
'systemctl list-units | wc -l; 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' \
do
...
```


PCH: Platform Controller Hub

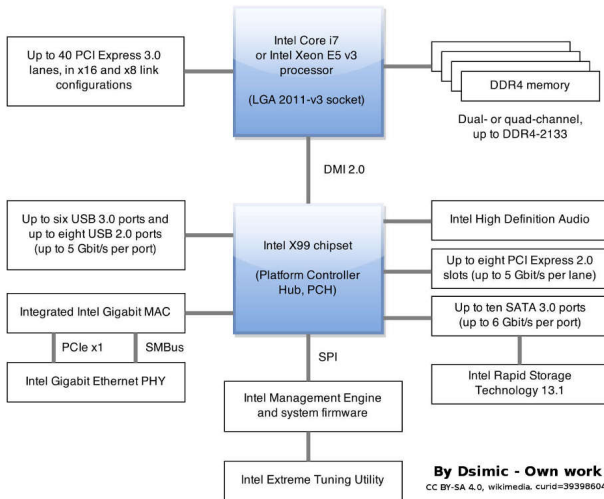


Figure: PCH: Platform Controller Hub

Some Terms

- PCH: Platform Controller Hub
- PCIe: Peripheral Component Interconnect Express — 32 bits for (16 * 1x or 8 * 2x or 4 * 4x or 2 * 8x or 1 * 16x) * (2 direction) lanes.
- DMI: Direct Media Interface. Eg. DMI 2.0 (2 GB/s; 4x)
- GT/s: GigaTransfers per second
- 1 KB (KiloByte) = 1000 bytes — 1 KiB (Kibibyte) = 1024 bytes¹
- SMB: System Management Bus
- SPI: Serial Peripheral Interface, a de facto standard bus.
- SATA: Serial AT Attachment. Eg. SATA 3.2 \approx 2 GB/s.
- DDR4 SDRAM: Double Data Rate Fourth-generation Synchronous Dynamic Random-Access Memory: 2 x DDR2 (DDR2 = 2 x DDR (DDR = 2 x SDRAM)). Eg. DDR4-3200 (8x SDRAM); Memory Clock: 400 MHz; Data Rate: 3200 MT/s; Module Name PC4-25600; Peak Transfer Rate: 25600 MB/s,

¹In IT tradition; 1 KB = 1024 bytes

The End

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