CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 00: Overview 1, Virtualization & Scripting

C. BinKadal

Sendirian Berhad

https://docOS.vlsm.org/Slides/os00.pdf
Always check for the latest revision!

REV418: Tue 30a Jan 2024 22:00

OS241[3]): Operating Systems Schedule 2023 - 2

Week	Topic[1])	OSC10[2])
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.

[1]) For schedule, see https://os.vlsm.org/#idx02

[2]) Silberschatz et. al.: **Operating System Concepts**, 10^{th} 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**, 10^{th} 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=3743. 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/
  - **OSP4DISS** — https://osp4diss.vlsm.org/
  - **This is How Me Do It!** — https://doit.vlsm.org/
    - PS: "Me" rhymes better than "I", duh!

Agenda

Agenda (2)

Week 00 Overview I: Topics[1]

- Role and purpose of the operating system
- Functionality of a typical operating system
- Mechanisms to support client-server models, hand-held devices
- Design issues (efficiency, robustness, flexibility, portability, security, compatibility)
- Influences of security, networking, multimedia, windowing systems
- Structuring methods (monolithic, layered, modular, micro-kernel models)
- Abstractions, processes, and resources
- Concepts of application program interfaces (APIs)
- The evolution of hardware/software techniques and application needs
- Device organization
- Interrupts: methods and implementations
- Concept of user/system state and protection, transition to kernel mode

[1] Source: ACM IEEE CS Curricula

Week 00 Overview I: Learning Outcomes (1)[1]

- Explain the objectives and functions of modern operating systems [Familiarity]
- Analyze the tradeoffs inherent in operating system design [Usage]
- Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve. [Familiarity]
- Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems. [Familiarity]
- Identify potential threats to operating systems and the security features design to guard against them. [Familiarity]
- Explain the concept of a logical layer. [Familiarity]

[1] Source: ACM IEEE CS Curricula

Week 00 Overview I: Learning Outcomes (2)[1]

- Explain the benefits of building abstract layers in hierarchical fashion. [Familiarity]
- Describe the value of APIs and middleware. [Assessment]
- Describe how computing resources are used by application software and managed by system software. [Familiarity]
- Contrast kernel and user mode in an operating system. [Usage]
- Discuss the advantages and disadvantages of using interrupt processing. [Familiarity]
- Explain the use of a device list and driver I/O queue. [Familiarity]

[1] Source: ACM IEEE CS Curricula

How to contact the Lecturer

- **Always introduce yourself**.
  - State your "`GitHubAccount`", "`Student ID`", "`Hypervisor`", and "`OS class`".
- Post a question/query on **SCELE** — (The enrollment key is **XXX**): https://scele.cs.ui.ac.id/course/view.php?id=3614.
- For SIAK related questions, use email with Subject:**[OS]** `rms46(AT)ui.ac.id`.
  - **DO NOT** send an email for assignment-related questions.

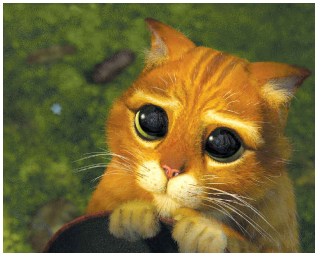
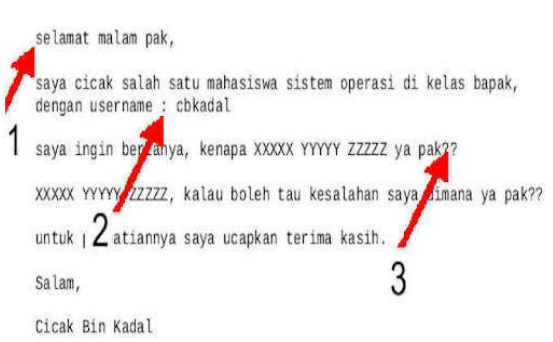


Figure: Never ever whine and pretend like this[1]!

[1] "Puss in Boot" is a DreamWorks/Paramount Picture character.

Emails between a "Gen Z" and a "Babyboomer"



Hallo Gen Z Zaman Now!

1 Kalimat baru seharusnya selalu dimulai dengan huruf besar.

2 Tanda baca seperti ":" (titik dua) seharusnya tanpa spasi. "Ini betul: ", "Ini salah : ".

3 Mengapa sampai lebih dari satu tanda-tanya??????????????

Salam,

Babyboomer.

Assessment

- **11 Weekly Assignments @ 11.11 points**.
  - Assignments will vary from week to week.
  - The assignment deadline will be by the end of every week. See https://os.vlsm.org/#idx02.
  - Check your points regularly at https://academic.ui.ac.id/
  - See also, https://os.vlsm.org/Log/.
  - **DO NOT COMPLAIN** weeks after!
- You need to log your weekly activities!
  - See https://doit.vlsm.org/ETC/logCodes.txt
  - See https://cbkadal.github.io/os241/TXT/mylog.txt
  - **4 SKS** (Units) means 12 hours (720 minutes) per week!
  - The average time allocation for each weekly assignment is 425 minutes—only 45% of the four SKS (units) load.
  - Most of the time (44 %) will be spent on the weekly assignment.

Average Time Allocation

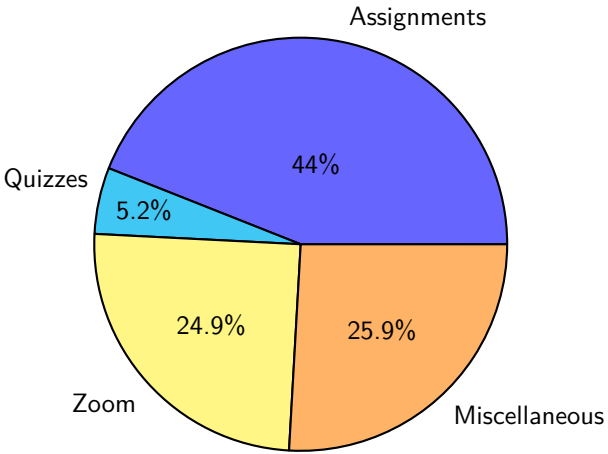


Figure: Operating Systems classes (2021-2022) student time allocation chart

NFT: Non-Fungible Tests

- Midterm (UTS)
  - The Midterm (UTS) is mandatory **IF** your SCORE is less than 25.0 before UTS. Otherwise, you need to register if you want to take UTS.
  - The UTS result will replace the worst grade of Assignment 00-05, even if the result is less.
- Final Term (UAS)
  - The final term (UAS) is mandatory **IF** your SCORE is less than 55.0 before UAS. Otherwise, you need to register if you want to take UAS.
  - The UAS result will replace the worst grade of Assignments 06-10, even if the result is less.
- Both UTS and UAS can only be held offline in an exam room.
- You can read an A4 size MEMO – reciprocal – written in your **handwriting**.
- Failure to show up on the day of the exam without reason and evidence will get a score of "0".

Final Grade (1)

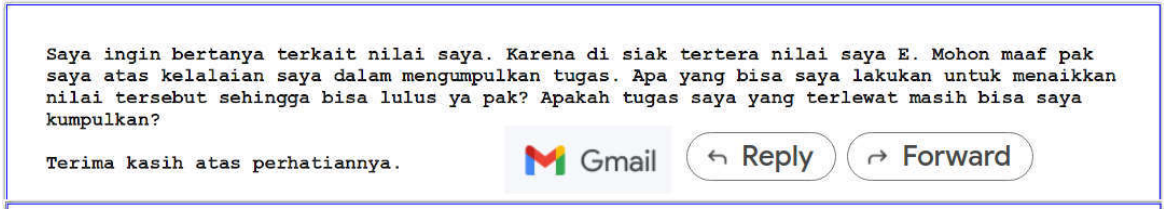
- The final grade will be the best 9 out of 11 assignments.
- **Two (2) "spare" assignments will be more than enough!**
- In case of emergency, contact your Academic Advisor!
- C-2C (C minus to C)
  - Up to 5 points, only if:
    - your grade is between 50.00 and 55.00, and
    - you have a "good" track record.
- Score Range

85 - ... = A	80 - 85 = A-	75 - 80 = B+	70 - 75 = B
65 - 70 = B-	60 - 65 = C+	55 - 60 = C	50 - 55 = D or C[1]
40 - 50 = D	30 - 40 = E	20 - 30 = E	00 - 20 = E

[1] C-2C: terms and conditions apply — void where prohibited by law.

The eternal recurring chronic problem

How to avoid receiving emails like the following at the end of the semester after grades have been published?



- Do not ask for any dispensations like a broken computer, circumcision (sunat), cold, competitions (including Gemastik), deadline extension, influenza, lame excuses, marriage, mourning, power failure, remedial, return to the village (mudik), slow network (lemot), two-semester evaluation, umrah, weddings, etc.
- It also includes: "It is not my fault but of $\{X : X \in \mathit{Lecturer} \parallel \mathit{Fasilkom} \parallel \mathit{UI} \parallel \mathit{Kampus\ Merdeka} \parallel \mathit{Immigration} \parallel \mathit{Foreign\ Embassy} \parallel \mathit{else}\}$."

Grade Examples

Mata Kuliah: CSCM602055 - Sistem Operasi

Kelas: 696070 - Sistem Operasi A, B, C, INT

[redacted]

-1 = no exam

Nama	NPM	W00	W01	W02	W03	W04	W05	W06	W07	W08	W09	W10	UTS	UAS	C-2C	TOTAL	
1234567890	1234567890	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	0	99.99	A
1234567891	1234567891	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	-1	-1	0	99.99	A
1234567892	1234567892	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	0	0	0	99.99	A
1234567893	1234567893	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	0	0	-1	-1	0	99.99	A
1234567894	1234567894	0	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	0	-1	-1	0	99.99	A
1234567895	1234567895	6.01	6.01	6.01	6.01	8.31	6.01	5.55	10.55	5.55	5.55	5.55	-1	-1	0	60.01	C+
1234567896	1234567896	6.01	6.01	6.01	6.01	8.31	6.01	5.55	10.55	5.55	5.55	5.55	0	0	0	59.55	C
1234567897	1234567897	6.01	6.01	6.01	6.01	8.31	6.01	5.55	10.55	5.55	5.55	5.55	11.1	11.1	0	70.65	B
1234567898	1234567898	6.21	4.88	6.21	3.11	6.21	6.21	6.21	6.01	6.01	6.01	6.01	-1	-1	0	55.09	C
1234567899	1234567899	6.21	4.88	6.21	3.11	6.21	6.21	6.21	6.01	6.01	6.01	6.01	0	0	1.05	55.01	C
1234567900	1234567900	6.21	4.88	6.21	3.11	6.21	6.21	6.21	6.01	6.01	6.01	6.01	11.1	11.1	0	65.27	B-

The Three-Strikes Rule

- All major academic rules violations will be handled directly by the Faculty of Computer Science, University of Indonesia.
- "Accidents" may happen. There will be warnings for the first two minor violations.
- Your final grade will be reduced for the third warning.
- Your final grade will be reduced to "D" for the fourth warning.
- Five (5) or more warnings will be considered as a significant academic-rules violation.

AIN'T DIFFICULT, lah!

Figure: Even this Goat will get "C" at the end of the semester!

Study From Anywhere?

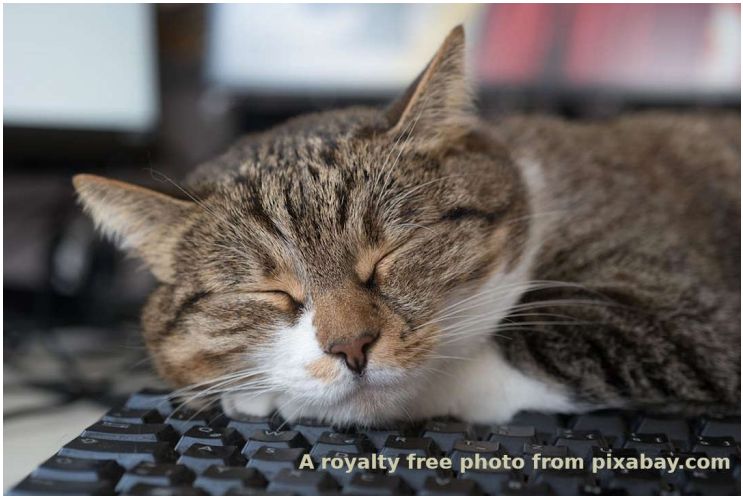
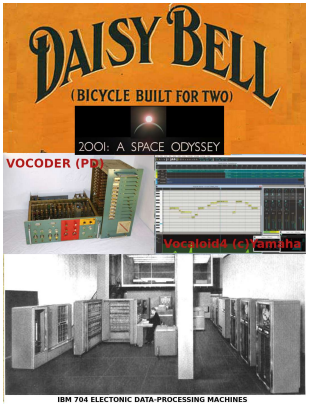


Figure: Who is on Zoom? What? I don't know! Why? Because! Today, I Don't Give a Darn!

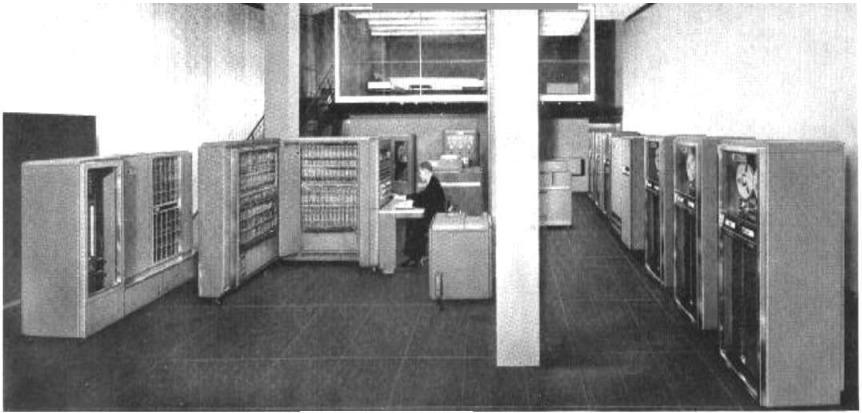
Prelude: Daisy Bell – Bicycle Built for Two




```
Daisy, Daisy,
Give me your answer, do!
I'm half crazy,
All for the love of you!
It won't be a stylish marriage,
I can't afford a carriage,
But you'll look sweet on the seat
Of a bicycle built for two!
```

YouTube (`https://youtu.be/TXK_cE9AqAI`). A choir (emulation) of VOCODER (pre WW2), IBM704 (1950s) and Vocaloid4 (2014). See also the classical movie 2001: A Space Odyssey.

IBM 704 at Los Alamos National Laboratory in the 1950s



IBM 704 ELECTONIC DATA-PROCESSING MACHINES

Estimate price (2020 value): USD 8,000,000.
Weight: 8800 kg — Electricity: ca. 200 kWatt — 42000 flops — 128 kbytes (eq.) core memory — 64 kbytes (eq.) drum memory — 3 Mbytes (eq.) Tape Unit.

Xiaomi 12 Pro – 12 GB / 256 GB

Figure: Source: Mi Indonesia (2024)

Out of Topic/Intermezzo/Segue

- Semiconductor Scalling:
  - Process Shrink: $10\mu m$ (1971), $250nm$ (1996), $10nm$ (2016), $5nm$ (2020), $3nm$ (2022).
  - Smaller Devices means:
    - Less space.
    - Less power consumption.
    - More density.
- Indonesia:
  - Fairchild Semiconductor Indonesia.
  - National Semiconductor Indonesia.
  - Minister of Manpower (Menteri Tenaga Kerja) 1983–1988.
- Technology:
  - SoC: System on a Chip.
  - SiP: System in a Package.
  - Fab/Foundry: Taiwan Semiconductor Manufacturing Company (TSMC), Ltd.
    - Have No Fab? It is OK! E.g., Marvell Technology, Inc (1995).
  - Lithography: ASML Holding, N.V: Advanced Semiconductor Materials Lithography.
  - Optics: Carl Zeiss SMT GmbH (This is NOT Optik Seis, Duh :).

TSMC Logic Nodes

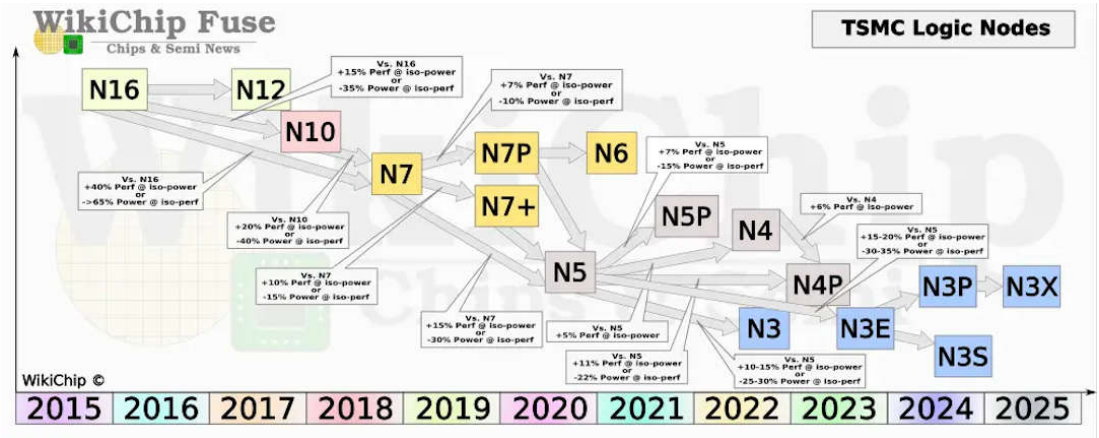


Figure: Source: WikiChip

The Computing Diciplines

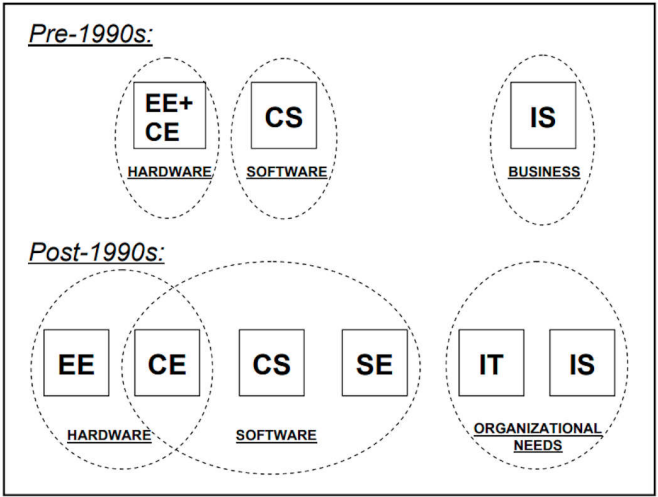


Figure 2.1. Harder Choices: How the Disciplines Might Appear to Prospective Students

Computing Curricula 2005 Page 12

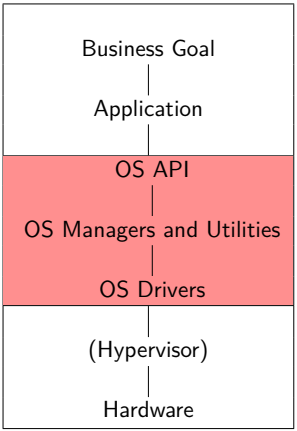
Lessons from the Development of the Boeing 787 Dreamliner

- 1997: Boeing acquired the nearly bankrupt McDonnell Douglas.
  - Result: "Boeing honorable name" with "McDonnell Douglas Greedy Culture."
- 2003: Boeing announced the Boeing 787 Dreamliner project.
- 2007: An "empty skeleton" prototype was rolled out on schedule. Many fuselage parts were temporarily attached.
  - Result: as expected, its stock price rose sharply.
- 2009: maiden flight after multiple delays.
  - Problem: Boeing and its partners have had no experience with many new technologies.
- 2011: Enter into service, but the problems did not go away.
  - Result: The budget increased from US$ 5billion to more than US$ 30billion.
- 2018: Problems did not go away but were overshadowed by the Boeing 737 MAX problems.
- Lesson learned?

LFS: Linux From Scratch (Week 00 — Week 10)

- THIS IS HOW WE DOIT!
- http://www.linuxfromscratch.org/lfs/view/stable/
- To build a GNU/Linux system from scratch (source code).
- To learn a GNU/Linux system inside out.
- To use a Virtual Machine.
- A Chicken and Egg dependency problem:
  - It would be best if you had the tools to build an Operating System.
  - You need an Operating System to build tools.
  - To build a cross-toolchain (compiler and its libraries).
  - To build cross utilities using the cross-toolchain.
  - To build an Operating System in a chroot environment.
  - To do iterations (if necessary).
- How deep would you like to know of a "real" Operating System?
- Whatever, however, from Week 00 to Week 10!
- **YOU** decide!

What defines an Operating System? (The Three Layers Model)



- The Three Layers Model
  - An Operating System is between your Application and your Hardware (or Hypervisor).
    - OS API: Application Programming Interface
    - OS Resources Managers and Utilities: Process, Scheduler, Dispatcher, (Virtual) Memory, Disk, I/O, Network, Security, Protection, etc.
    - OS Device Drivers: controls devices
  - Remember that your future "**Business Goal**" may not directly relate to an Operating System at all!

OSC10 (Silberschatz) Chapter 1 and 2

- OSC10 Chapter 1
  - What Operating Systems Do
  - Computer-System Organization
  - Computer-System Architecture
  - Operating-System Operations
  - Resource Management
  - Security and Protection
  - Virtualization
  - Distributed Systems
  - Kernel Data Structures
  - Computing Environments
  - Free/Libre and Open-Source Operating Systems
- OSC10 Chapter 2
  - Operating System Services
  - User and Operating System-Interface
  - System Calls
  - System Services
  - Linkers and Loaders
  - Why Applications are Operating System Specific
  - Operating-System Design and Implementation
  - Operating System Structure
  - Building and Booting an Operating System
  - Operating System Debugging

Remember Computer Organization (POK/DDAK)?

- You should understand:
  - von Neumann Model.
  - Buses, Bridges, Transfer Rate, Clock.
  - Memory: DDR, DDR-2, DDR-3, DDR-3+ ...
  - Cache, Buffer, Spool, & Pipelining.
  - Direct Memory Access (DMA).
  - Port & Memory Mapped I/O.
  - CPU: (privilege/kernel/supervisor mode) vs. (user mode).
  - Physical (Hardware) Limitation.
  - Priority: Read vs. Write.
  - Interrupts: Polling & Vectored.
  - Multiprocessors: Symmetric vs. Asymmetric.
  - Multicore & Multithreading.
  - Clustered Systems.
  - Numbers: base 2, base 8, base 10, base 16.
    - Base 2: 110010101010_2
    - Base 8: $01234567_8 = 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111_2$
    - Base 10: 012 345 679

Physics 101: Signal Length (E.g. 3 GHz)

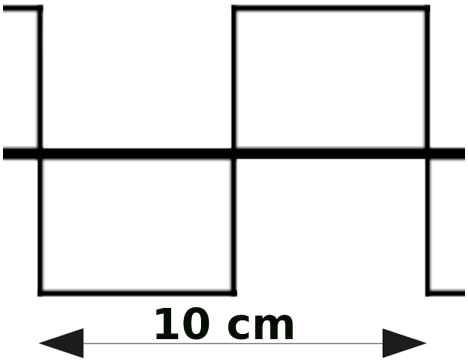


Figure: What is the length of a 3 GHz signal?

Physics 101: Safe Distance for 3 GHz

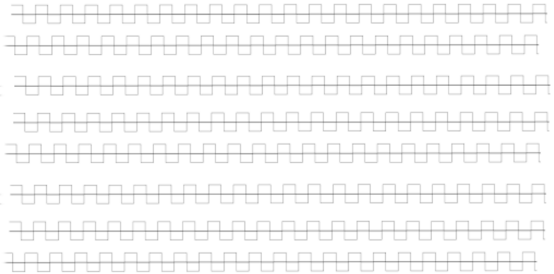
Figure: Safe Distance

Physics 101: Serial vs. Parallel Transmission

- Serial Transmission
  - Longer Distance
  - Easy to implement

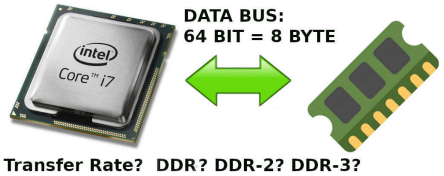
(serial)

- Parallel Transmission
  - Faster
  - Not easy to implement



(parallel)

Transmission Rate (E.g. **BUS**: 64 bit/133 MHz)



- E.g. **BUS**: 64 bit, **Clock**: 133 MHz
  - SDRAM (Synchronous Dynamic RAM): 1 transmission/cycle.
    Transfer Rate = `64/8 byte x 133M x 1 = 1064 Mbyte/s`.
  - DDR (Double Date Rate): 2 transmission/cycle.
    Transfer Rate = `64/8 byte x 133M x 2 = 2128 Mbyte/s`.
  - DDR-2 (Double Date Rate 2): 4 transmission/cycle.
    Transfer Rate = `64/8 byte x 133M x 4 = 4256 Mbyte/s`.
  - DDR-3 (Double Date Rate 3): 8 transmission per cycle.
    Transfer Rate = `64/8 byte x 133M x 8 = 8512 Mbyte/s`.
  - DDR-3+ = DDR-3 with a better clock rate, lower voltage, and greater capacity.

CPU: SuperVisor Mode

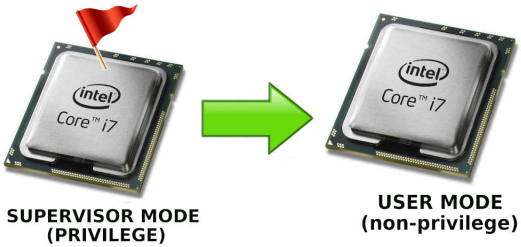


Figure: SuperVisor (Privilege) Mode to User Mode

- SuperVisor Mode
  - A.k.a. Kernel Mode, Privilege Mode.
  - Initial STATE (Mode) of a CPU (Power On).
  - STATE (Mode) after Interrupt.
  - All operations are allowed, including to switch to User Mode!

CPU: User Mode

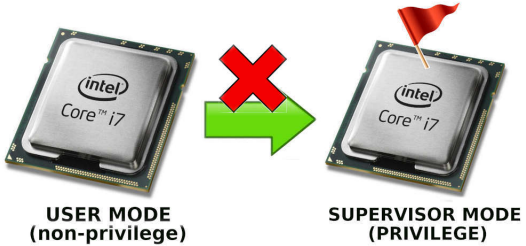


Figure: User Mode to SuperVisor (Privilege)

- User Mode
  - It is not allowed to switch back to SuperVisor Mode.
  - It is not allowed to access I/O directly.
  - It is not allowed to modify the Interrupt Vector.
  - It is allowed to request Interrupt.

Can you read a Block Diagram?

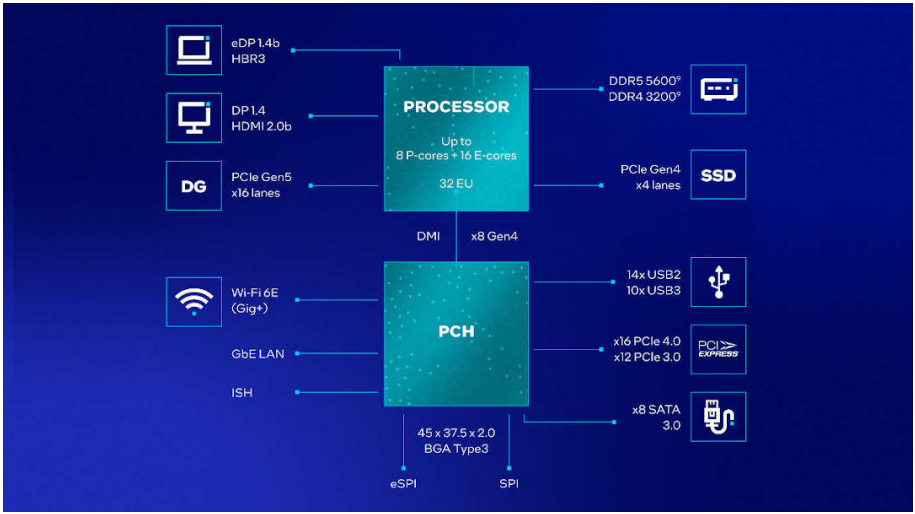


Figure: Block Diagram

Block Diagram

- eDP: Embedded DisplayPort, for internal displays.
- DDI: Digital Display Interface
  - DP: Display Port
  - HDMI: High-Definition Multimedia Interface
- DMI: Direct Media Interface
- PCIe: Peripheral Component Interconnect express
- eSPI: Enhanced Serial Peripheral Interface
- SPI: Serial Peripheral Interface
- SMBus: System Management Bus
- HD Audio: High Definition Audio
- USB: Universal Serial Bus

What is an APIC?!

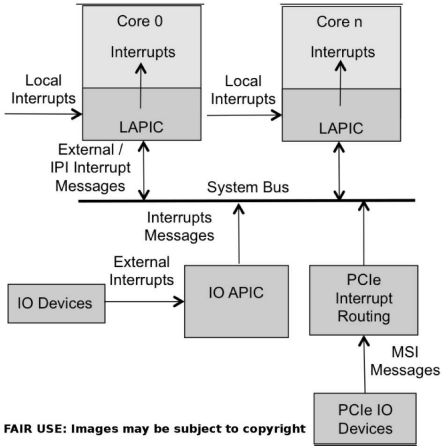


Figure: APIC (Advanced Programmable Interrupt Controller)

And, what is "Interrupt Handling"?

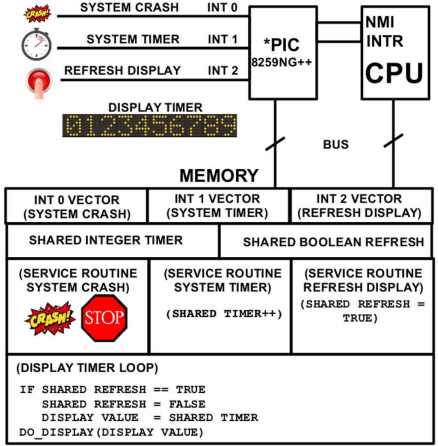


Figure: Interrupt Handling with PIC (Programmable Interrupt Controller)

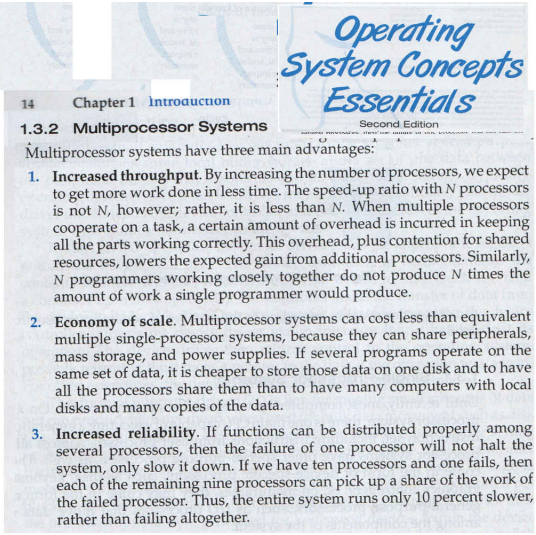
The Operating System Managers

- Process Manager:
  - Creating/Deleting; Suspending/Resuming; Synchronization; Communication; Scheduling
- Memory Manager:
  - Tracking; Move In/Move Out; Allocating/Deallocating.
- Storage/File System Manager:
  - Create/Delete; Open/Close; Read/Write.
- Mass Storage Manager:
  - Scheduling; Allocating; Free Space.
- I/O Manager:
  - Buffering; Caching; Spooling.
  - Interfacing (driving).
- Protecting & Security Manager:
  - Protecting.
  - Security.

Any idea what these following terms mean?!

- Scripting: bash, regex, sed, awk
- Security and Protection
- File System
- Data Structure in a (logical) Memory
- Virtual Memory
- Concurrency
- Synchronization
- Mass Storage
- UEFI, GRUB, and systemd
- I/O
- I/O Programming

Week 00: QUIZ Example #1 (from OSC2e)



Operating System Concepts Essentials
Second Edition

14 Chapter 1 Introduction

1.3.2 Multiprocessor Systems

Multiprocessor systems have three main advantages:

1. **Increased throughput**. By increasing the number of processors, we expect to get more work done in less time. The speed-up ratio with N processors is not N, however; rather, it is less than N. When multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly. This overhead, plus contention for shared resources, lowers the expected gain from additional processors. Similarly, N programmers working closely together do not produce N times the amount of work a single programmer would produce.
2. **Economy of scale**. Multiprocessor systems can cost less than equivalent multiple single-processor systems, because they can share peripherals, mass storage, and power supplies. If several programs operate on the same set of data, it is cheaper to store those data on one disk and to have all the processors share them than to have many computers with local disks and many copies of the data.
3. **Increased reliability**. If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. If we have ten processors and one fails, then each of the remaining nine processors can pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether.

True or False?

The advantages of a multiprocessor system include: increased throughput, economy of scale, and increased reliability.

(from MidTerm 2016)

Week 00: QUIZ Example #2 (from OSC10)

Preface

Operating systems are an essential part of any computer system. Similarly, a course on operating systems is an essential part of any computer science education. This field is undergoing rapid change, as computers are now prevalent in virtually every arena of day-to-day life—from embedded devices in automobiles through the most sophisticated planning tools for governments and multinational firms. Yet the fundamental concepts remain fairly clear, and it is on these that we base this book.

- **T/F** Operating systems are an essential part of any computer system.
- **T/F** Operating systems are not an essential part of any computer system.
- **T/F** A course on operating systems is essential to any computer science education.
- **T/F** A course on operating systems is optional to any computer science education.
- **T/F** The Operating System field is undergoing rapid change, as computers are now prevalent in virtually every arena of day-to-day life.
- **T/F** The Operating System field is not undergoing rapid change, as computers are now prevalent in virtual machines.

Week 00: QUIZ Example #3

- **TRUE/FALSE**
  The best way to get any help is to send an email to `rms46 AT ui.ac.id`.
- **TRUE/FALSE**
  Questions regarding assignments should be posted at SCELE.
- **TRUE/FALSE**
  Making a **PUSS IN BOOT** face is increasing the chance to get a better deal.
- **TRUE/FALSE**
  Anyone can appeal any time, even after the (official) final grade is announced on SIAK.
- **TRUE/FALSE**
  There are bonus points for early assignment submission.

Assignments

- You need to run "VirtualBox" or "UTM" on a computer with more than 4GB RAM and at least 64 GB disk space.
- Each weekly assignment will be due seven days after it is announced. The weekly schedule will be at https://os.vlsm.org/#idx02.
- Use the **"GitHub web interface"** for the Week 00 assignment. However, starting Week 01, you need to understand **"pull, add, commit, push, and ssh-keys"**.
- Submit (push) the assignments to https://github.com/. If you still don't have one, you must sign up for a GitHub account. More information will follow.
- See the assignment list at https://demos.vlsm.org/#idx000.

Course Highlights and Syllabus

Coverage

This is an introduction to a modern operating systems course. It will cover general overview, computer architecture review, operating system overview, GNU/Linux CLI, scripting, C language overview, protection, security, privacy, systemd, I/O, addressing and pointers, memory management, processes and threads, virtual memory, synchronization, mutual exclusion, deadlock, CPU scheduling algorithms, file systems, and I/O programming.

Student-Centered

This course is student-centered where responsibility is in the hands of the students. Students are expected to be prepared for the class meeting.

GNU/Linux

Students will have a thorough understanding of how GNU/Linux provides services by using a Command Line Interface.

Week 00 Overview I: Topics[1]

- Role and purpose of the operating system
- Functionality of a typical operating system
- Mechanisms to support client-server models, hand-held devices
- Design issues (efficiency, robustness, flexibility, portability, security, compatibility)
- Influences of security, networking, multimedia, windowing systems
- Structuring methods (monolithic, layered, modular, micro-kernel models)
- Abstractions, processes, and resources
- Concepts of application program interfaces (APIs)
- The evolution of hardware/software techniques and application needs
- Device organization
- Interrupts: methods and implementations
- Concept of user/system state and protection, transition to kernel mode

[1] Source: ACM IEEE CS Curricula

Week 00 Overview I: Learning Outcomes (1)[1]

- Explain the objectives and functions of modern operating systems [Familiarity]
- Analyze the tradeoffs inherent in operating system design [Usage]
- Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve. [Familiarity]
- Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems. [Familiarity]
- Identify potential threats to operating systems and the security features design to guard against them. [Familiarity]
- Explain the concept of a logical layer. [Familiarity]

[1] Source: ACM IEEE CS Curricula

Week 00 Overview I: Learning Outcomes (2)[1]

- Explain the benefits of building abstract layers in hierarchical fashion. [Familiarity]
- Describe the value of APIs and middleware. [Assessment]
- Describe how computing resources are used by application software and managed by system software. [Familiarity]
- Contrast kernel and user mode in an operating system. [Usage]
- Discuss the advantages and disadvantages of using interrupt processing. [Familiarity]
- Explain the use of a device list and driver I/O queue. [Familiarity]

[1] Source: ACM IEEE CS Curricula

Week 01 Overview II: Topics[1]

- Intelectual Property Rights (IPR)
- Software Licenses and Free Software
- Operating System Services and Interfaces
- System Calls and System Programming
- Types of virtualization (including Hardware/Software, OS, Server, Service, Network)
- Hypervisors
- Portable and cost of virtualization; emulation vs. isolation
- Cloud services: IAAS, PAAS and Platform APIs, SAAS
- Introduction to Scripting and REGEX.

[1]Source: ACM IEEE CS Curricula

Week 01 Overview II: Learning Outcomes[1]

- Explain the concept of virtual memory and how it is realized in hardware and software. [Familiarity]
- Discuss hypervisors and the need for them in conjunction with different types of hypervisors. [Usage]
- Differentiate emulation and isolation. [Familiarity]
- Evaluate virtualization trade-offs. [Assessment]
- Discuss the importance of elasticity and resource management in cloud computing. [Familiarity]
- Explain the advantages and disadvantages of using the virtualized infrastructure. [Familiarity]

[1]Source: ACM IEEE CS Curricula

Week 02 Security & Protection: Topics[1]

- Overview of system security
- Cyber Security Introduction
- Policy/mechanism separation
- Security methods and devices
- Protection, access control, and authentication
- Backups
- Safety and Privacy
- Threads
- Cryptography: (Symmetric and Asymmetric) Encryption,
- C Language

[1]Source: ACM IEEE CS Curricula

Week 02 Security & Protection: Learning Outcomes[1]

- Articulate the need for protection and security in an OS (cross-reference IAS/Security Architecture and Systems Administration/Investigating Operating Systems Security for various systems). [Assessment]
- Summarize the features and limitations of an operating system used to provide protection and security [Familiarity]
- Explain the mechanisms available in an OS to control access to resources [Familiarity]
- Carry out simple system administration tasks according to a security policy, for example creating accounts, setting permissions, applying patches, and arranging for regular backups [Usage]

[1]Source: ACM IEEE CS Curricula

Week 03 File System & FUSE: Topics[1]

- 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

[1] Source: ACM IEEE CS Curricula

Week 03 File System & FUSE: Learning Outcomes[1]

- 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]

[1]Source: ACM IEEE CS Curricula

Week 04 Addressing: Topics[1]

- Bits, bytes, and words
- Numeric data representation and number bases
- Representation of records and arrays

[1] Source: ACM IEEE CS Curricula

Week 04 Addressing: Learning Outcomes[1]

- Explain why everything is data, including instructions, in computers. [Familiarity]
- Explain the reasons for using alternative formats to represent numerical data. [Familiarity]
- Describe the internal representation of non-numeric data, such as characters, strings, records, and arrays. [Familiarity]

[1] Source: ACM IEEE CS Curricula

Week 05 Virtual Memory: Topics[1]

- Review of physical memory and memory management hardware
- Virtual Memory
- Caching
- Memory Allocation
- Memory Performance
- Working sets and thrashing

[1] Source: ACM IEEE CS Curricula

Week 05 Virtual Memory: Learning Outcomes[1]

- Explain memory hierarchy and cost-performance trade-offs. [Familiarity]
- Summarize the principles of virtual memory as applied to caching and paging. [Familiarity]
- Describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction). [Familiarity]
- Defend the different ways of allocating memory to tasks, citing the relative merits of each. [Assessment]
- Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed. [Assessment]
- Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem. [Familiarity]

[1]Source: ACM IEEE CS Curricula

Week 06 Concurrency: Topics[1]

- States and state diagrams
- Structures (ready list, process control blocks, and so forth)
- Dispatching and context switching
- The role of interrupts
- Managing atomic access to OS objects
- Implementing synchronization primitives
- Multiprocessor issues (spin-locks, reentrancy)

[1]Source: ACM IEEE CS Curricula

Week 06 Concurrency: Learning Outcomes (1)[1]

- Describe the need for concurrency within the framework of an operating system. [Familiarity]
- Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks. [Usage]
- Summarize the range of mechanisms that can be employed at the operating system level to realize concurrent systems and describe the benefits of each. [Familiarity]
- Explain the different states that a task may pass through and the data structures needed to support the management of many tasks. [Familiarity]

[1] Source: ACM IEEE CS Curricula

Week 06 Concurrency: Learning Outcomes (2)[1]

- Summarize techniques for achieving synchronization in an operating system (e.g., describe how to implement a semaphore using OS primitives). [Familiarity]
- Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system. [Familiarity]
- Create state and transition diagrams for simple problem domains. [Usage]

[1] Source: ACM IEEE CS Curricula 2023 (beta)

Week 07 Synchronization & Deadlock: Topics[1]

- Shared Memory and Critical Section
- Consistency, and its role in programming language guarantees for data-race-free programs
- Message passing: PtPo vs Multicast, Blocking vs non-blocking, buffering.

[1] Source: ACM IEEE CS Curricula

Week 07 Synchronization & Deadlock: Learning Outcomes[1]

- Use mutual exclusion to avoid a given race condition. [Usage]
- Give an example of an ordering of accesses among concurrent activities (e.g., program with a data race) that is not sequentially consistent. [Familiarity]
- Use semaphores to block threads [Usage]

[1] Source: ACM IEEE CS Curricula

Week 08 Scheduling: Topics[1]

- Preemptive and non-preemptive scheduling
- Schedulers and policies
- Processes and threads
- Deadlines and real-time issues

[1]Source: ACM IEEE CS Curricula

Week 08 Scheduling: Learning Outcomes[1]

- Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes. [Usage]
- Describe relationships between scheduling algorithms and application domains. [Familiarity]
- Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O. [Familiarity]
- Describe the difference between processes and threads. [Usage]
- Compare and contrast static and dynamic approaches to real-time scheduling. [Usage]
- Discuss the need for preemption and deadline scheduling. [Familiarity]
- Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing. [Usage]

[1]Source: ACM IEEE CS Curricula

Week 09 Storage, Firmware, Bootloader, & Systemd: Topics[1]

- Storage
- Storage Arrays
- BIOS
- Loader
- Systemd

[1]Source: ACM IEEE CS Curricula

Week 09 Storage, Firmware, Bootloader, & Systemd: Learning Outcomes[1]

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

[1]Source: ACM IEEE CS Curricula

Week 10 I/O & Programming: Topics[1]

- Characteristics of serial and parallel devices
- Abstracting device differences
- Buffering strategies
- Direct memory access
- Recovery from failures
- I/O Programming
- Network Programming

[1] Source: ACM IEEE CS Curricula

Week 10 I/O & Programming: Learning Outcomes[1]

- 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]
- I/O Programming [Usage]
- Network Programming [Usage]

[1] Source: ACM IEEE CS Curricula

Week 00: Summary

- What is an Operating System?
  - Definition: Resource Allocator & Control Program.
  - Why taking an Operating System class?
- Computer Organization Review
- The Manager Set
  - Process Manager, Memory Manager, I/O Manager, Storage Manager.
- Security and Protection
- Virtualization
  - Hypervisor type 0, 1, 2
  - Paravirtualization, Emulators, Containers.
  - VCPU: Virtual CPU
  - Virtualization Implementation:
    - Trap-and-Emulate mode
    - Binary Translation mode

TIPS (1)

- See also https://rms46.vlsm.org/2/221.pdf.
- Register yourself via Google Forms as soon as possible. No need to wait until your SIAK is approved (RMS)!
- For any administrative issues, contact SEKRE at building B, 2^{nd} floor – especially for absences, illness, sick letters, follow-up exams, etc. Please do not contact the **Lecturer** (RMS).
- Please complete the follow-up/paper work within six (6) working days (RMS).

TIPS (2)

- Study the Operating System Concept book, which deals with the material that will be discussed that week (MIM). Make a summary of the material in your Memo (IP).
- You should understand every single problem of the past examinations. Write down all hints in your "**MEMO**" (MHP).
- You are allowed to bring a sheet of MEMO for the midterm (UTS) and a sheet for the finalterm (UAS) (RMS).
- You should understand every single line of the "**DEMOS**" (MHP).
- You should ask **the lecturer** or anyone, anything you do not understand (TA).
- The **ASDOS** are the lectures's helper, not your personal tutors (RMS).

Special Thanks

Special thanks for writing and reviewing this material to:
Anisha Inas Izdihar (AII), Benedictus Alvin (BA), Dennis Al Baihaqi Walangadi, Dionisius Baskoro Samudra, Eugene Brigita Lauw, Ibnu Sofian Firdaus (ISF), Irmanpen Panjaitan (IP), Ivana Irene Thomas (IIT), Marcia Nadin Pramasiwi, Michael Giorgio Wirawan (MGW), Muhamad Yoga Mahendra, Muhammad Afkar (MA), Muhammad Hanif Pratama (MHP), Muhammad Iqbal Mahendra (MIM), Muhammad Krishertanto Adityapu, M. Ikhsan Kurniawan (MIK), Nixi Sendya Putri (NSP), Raihan Mahendra Sutanto (RM), Rizki Leonardo (RL), Shavira Adeva (SA), Stefan Mayer Sianturi (SMS), Thrisnadevany Amalia (TA), Zhelia Alifa (ZA);

The End

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