# CSF2600505 Sistem Operasi
# CSGE602055 Operating Systems
# Week 00: Overview 1

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

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

REV240 14-Sep-2020

# Operating Systems 2020-2[2]) — **PJJ from HOME**
# ZOOM: International [Tue 08-10] — A/Matrix [Tue 10-12]

| **Week** | **Schedule** | **Topic** | **OSC10**[1]) |
|---|---|---|---|
| Week 00 | 15 Sep - 21 Sep 2020 | Overview 1, Virtualization & Scripting | Ch. 1, 2, 18. |
| Week 01 | 22 Sep - 28 Sep 2020 | Overview 2, Virtualization & Scripting | Ch. 1, 2, 18. |
| Week 02 | 29 Sep - 05 Oct 2020 | Security, Protection, Privacy, & C-language. | Ch. 16, 17. |
| Week 03 | 06 Oct - 12 Oct 2020 | File System & FUSE | Ch. 13, 14, 15. |
| Week 04 | 13 Oct - 19 Oct 2020 | Addressing, Shared Lib, & Pointer | Ch. 9. |
| Week 05 | 20 Oct - 26 Oct 2020 | Virtual Memory | Ch. 10. |
| Week 06 | 27 Oct - 16 Nov 2020 | Concurrency: Processes & Threads | Ch. 3, 4. |
| | 29 Oct 2020 | Maulid Nabi | |
| Week 07 | 17 Nov - 23 Nov 2020 | Synchronization & Deadlock | Ch. 6, 7, 8. |
| Week 08 | 24 Nov - 30 Nov 2020 | Scheduling + W06/W07 | Ch. 5. |
| Week 09 | 01 Dec - 07 Dec 2020 | Storage, Firmware, Bootloader, & Systemd | Ch. 11. |
| Week 10 | 08 Dec - 16 Dec 2020 | I/O & Programming | Ch. 12. |
| | 09 Dec 2020 | Pil Kada | |

[1]Silberschatz et. al.: **Operating System Concepts**, $10^{th}$ Edition, 2018.

[2]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 also http://codex.cs.yale.edu/avi/os-book/OS10/.
- **Resources**
  - **SCELE** — https://scele.cs.ui.ac.id/course/view.php?id=3020. The enrollment key is **XXX**.
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/
  - **Problems** — https://rms46.vlsm.org/2/: 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).
- **Build your own Virtual Guest** https://osp4diss.vlsm.org/

# Agenda

1. Start
2. Schedule
3. Agenda
4. How to contact the Lecturer
5. Assessment
6. The Three-Strikes Rule
7. Week 00: Review
8. Assignments
9. Schedule
10. Week 00 Assignment #1: Public Repository "os202"
11. Week 00 Assignment #2: Start Week 00 Log
12. Week 00 Assignment #3 & #4
13. Week 00 Assignment #5: Course Registration
14. Course Highlights and Syllabus

# Agenda (2)

# How to contact the Lecturer[2]

| |
|---|
| For Q & A, use WhatsApp Group **OperatingSystems** (info +62-881-456-**XXXX**) |
| Email (Subject:**[HELP]**) operatingsystems@vlsm.org State your "Name", "ID", and "OS class". |
| **SCELE** — (The enrollment key is **XXX**): https://scele.cs.ui.ac.id/course/view.php?id=3020. |

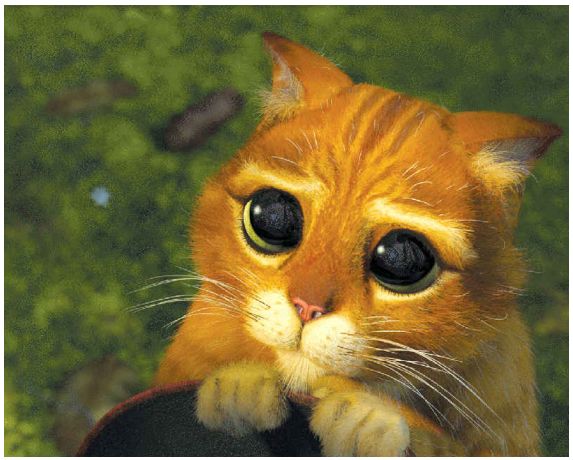


Figure: Never ever whine and pretend like this[1]!

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

[2] FYI: King Goerge II founded the University of Goettingen in 1734.

# Assessment

- **4 SKS** (Units) means 12 hours per week!
  - You need to log your weekly activities!
- **11** (weekly) assigments @ 11.11 points.
  - Assignments will vary from week to week.
  - The assignment deadline will be by the end of every week (see page 2).
  - See the check list at the end of this presentation.
- Final grade: the sum of the best 9 out of 11 assignments.

| | | | |
|---|---|---|---|
| 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 |

  - **C-2C:** upto 5 points.
    - Only if your grade is between 50.00 and 55.00 and you have a "good" track record.
  - Check your points regularly at `https://academic.ui.ac.id/` and **DO NOT COMPLAIN** weeks after! See also, `https://os.vlsm.org/`.

[1] Terms and Conditions apply. Void where prohibited by law.

# The Three-Strikes Rule


- All major academic-rules violations will be handled directly by the Faculty of Computer Science, University of Indonesia.
- "Accidents" may happens. 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 warnings (and more) will be considered as a major academic-rules violation.

# AIN'T DIFFICULT, lah!

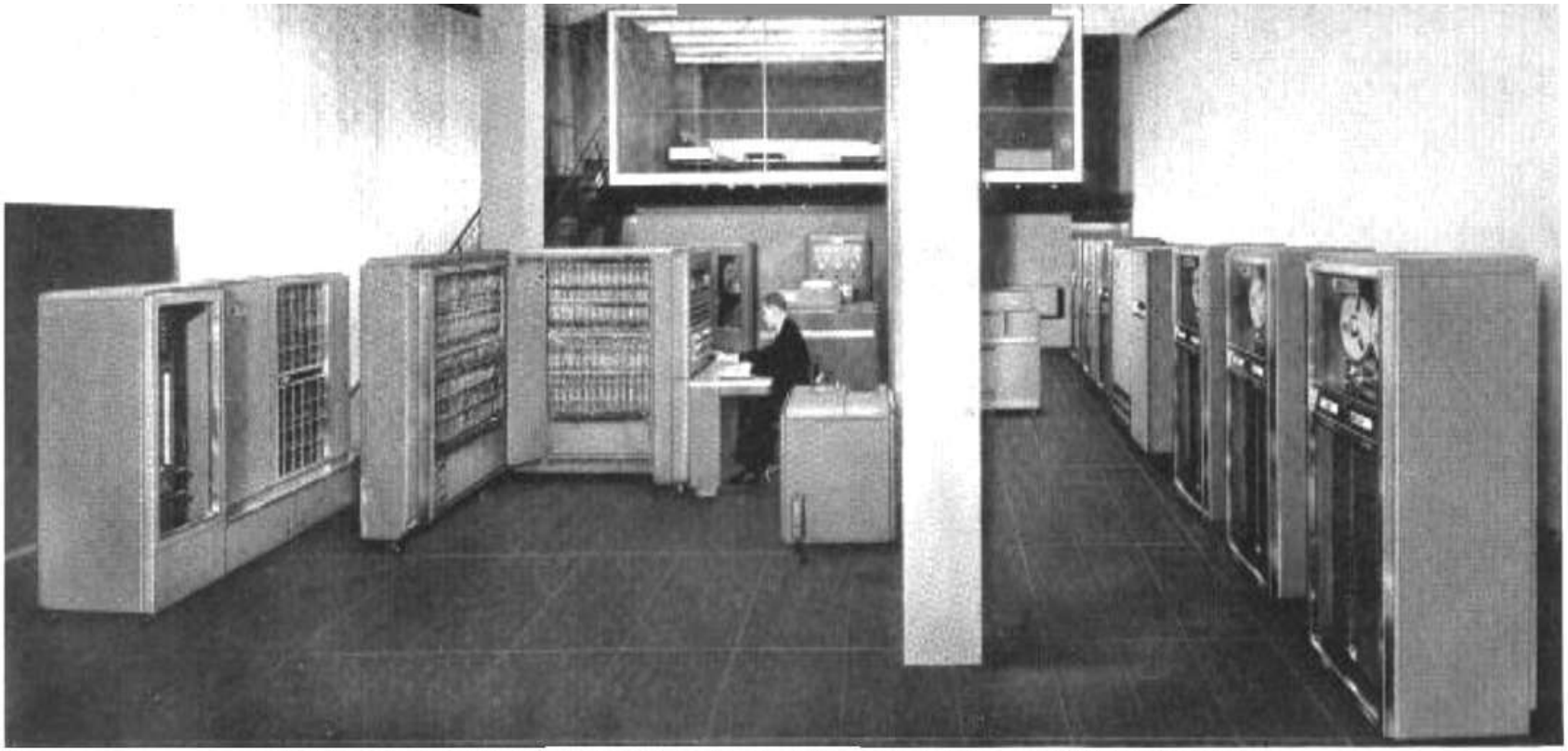

Figure: Even this Goat will get "C" at the end of the semester!

# 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!

See also https://youtu.be/TXK_cE9AqAI.

# IBM 704 at Los Alamos National Laboratory in the 1950s



**IBM 704 ELECTONIC DATA-PROCESSING MACHINES**

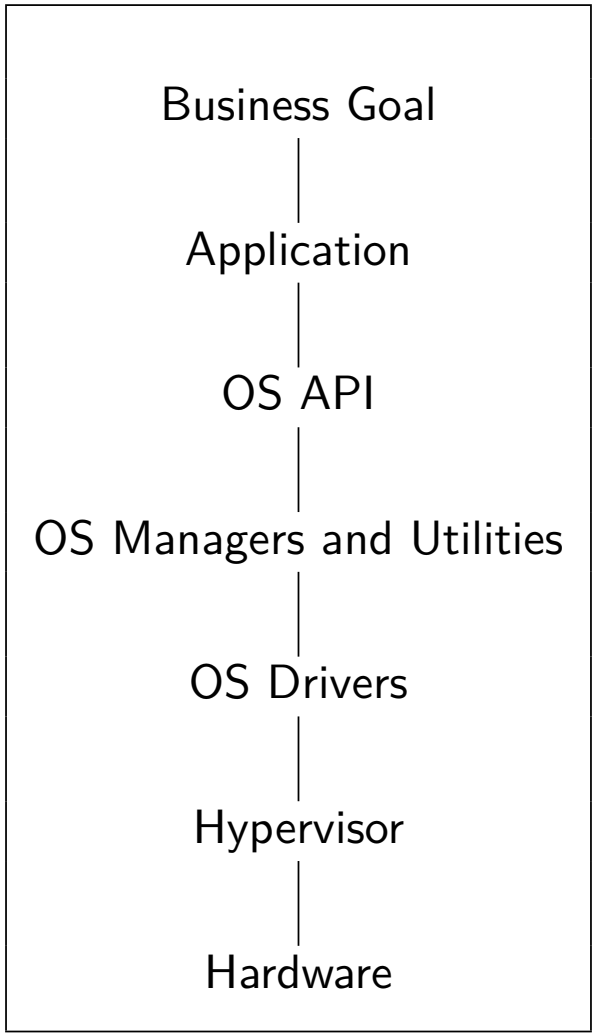
Estimate price (2020 value): USD 8,000,000.
Weight: 8800 kg — Electricity: ca. 200 kWatt — 42000 flops — 128 kbytes (eq.) core memory — 64 kbytes (eq.) drum memory — 3 Mbytes (eq.) Tape Unit.

# QS855, 256GB, 12 GB, 48+12 MP, 6.4”, 4000 mAh


Estimate price (2020 value): Rp. 8,000,000.

# Week 00: Review

- What is an Operating System?
- Why taking an Operating System class?



# Remember Computer Organization (POK/DDAK)?

- You should understand:
  - von Neumann Model.
  - Buses, Bridges, Transfer Rate, Clock.
  - Memory: DDR, DDR-2, 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
    - Base 16: $9AB\ CDEF_{16} = 1001\ 1010\ 1011\ \ 1100\ 1101\ 1110\ 1111_2$

# Can you read a Block Diagram?

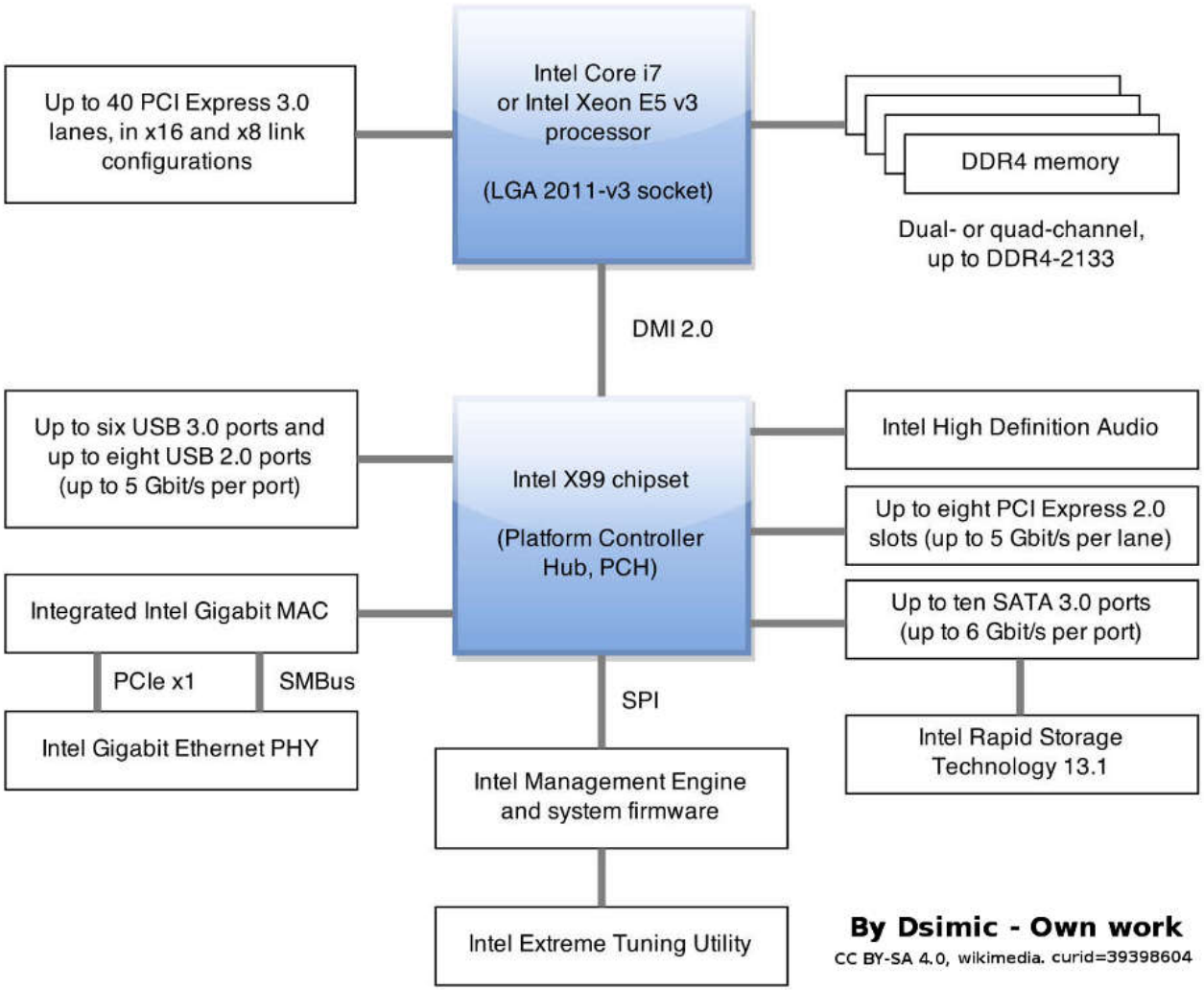



Figure: Block Diagram

# What is an APIC?!

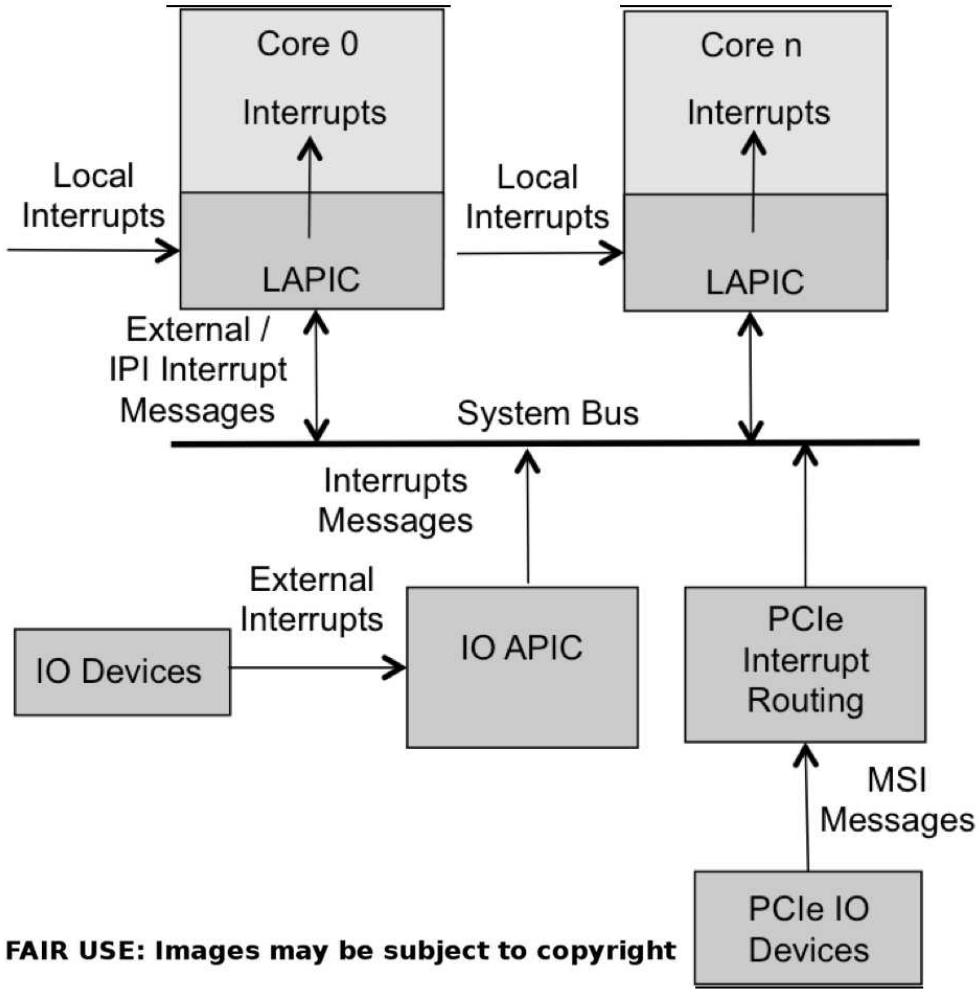



Figure: APIC (Advanced Programmable Interrupt Controller)

# And, what is "Interupt Handling"?

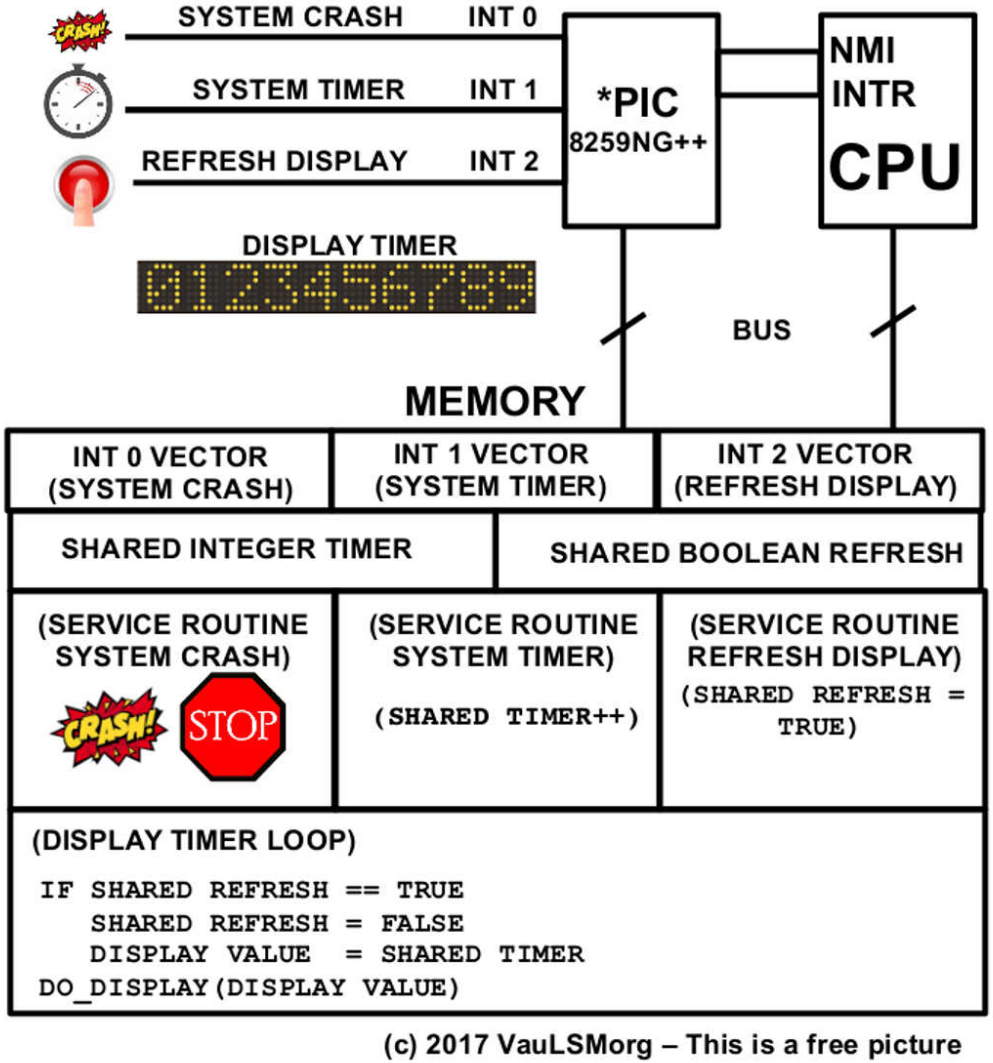



Figure: Interupt Handling with PIC (Programmable Interrupt Controller)

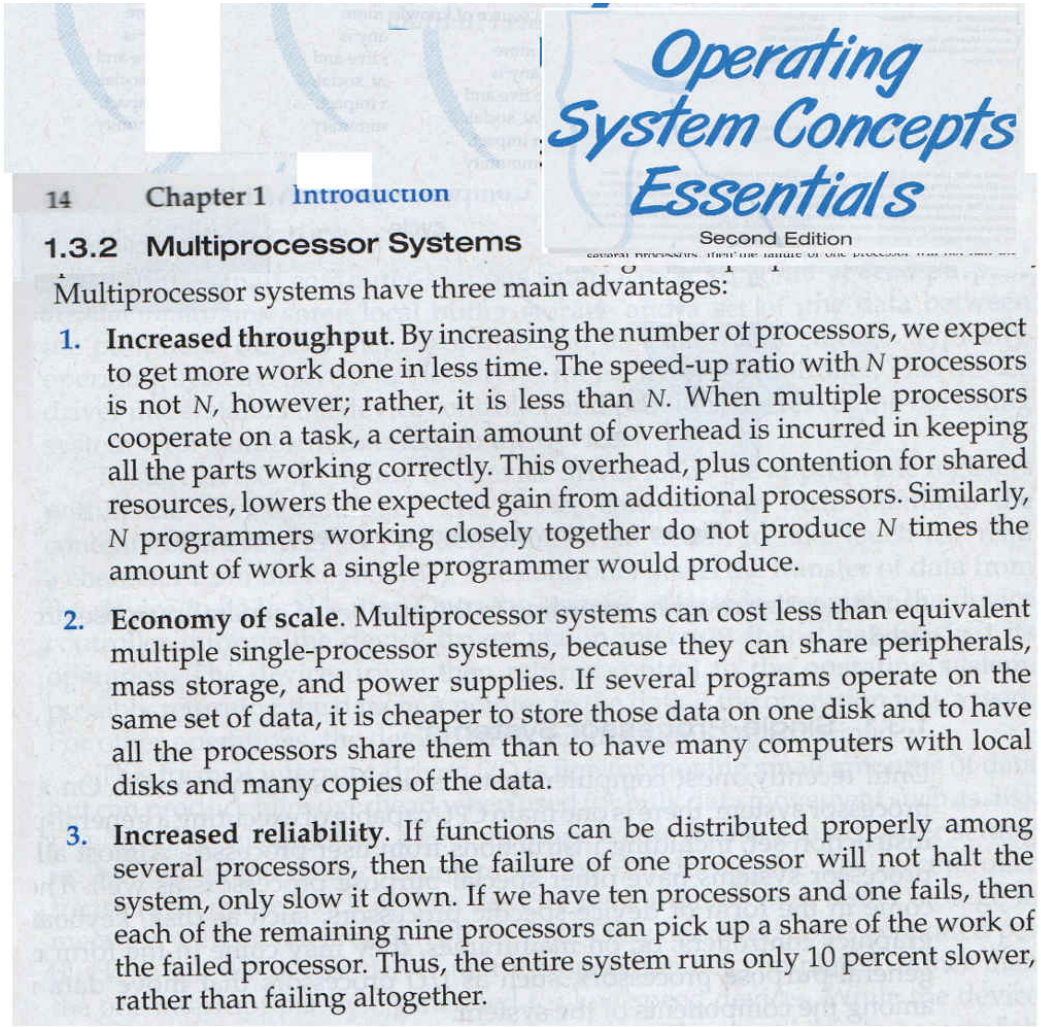
# The Operating System Managers

- Process Manager:
  - Creating/Deleting; Suspending/Resuming; Synchronization; Communication; Schedulling
- Memory Manager:
  - Tracking; Move In/Move Out; Allocating/Deallocating.
- Storage/File System Manager:
  - Create/Delete; Open/Close; Read/Write.
- Mass Storage Manager:
  - Schedulling; 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: Problem Example (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.

Figure: **T / F** The advantages of a multiprocessor system include: increased throughput, economy of scale, and increased reliability (Week00 2016-1).

# Assignments

- There will no mid-term (UTS) nor final-term (UAS). Instead, there will be 11 weekly assignments. Your grade will be taken from the best 9 out of 11 assignments.
- Each assignment deadline will be by the end of that "week". The weekly schedule is always on page 2!
- Understand how to "**add, commit, and push**" will be an advantage. But, you are allowed to use the "github web interface" for the Week 00 assignment.
- Submit (push) the assignments to `github.com`. If you still don't have one, you need to sign up for a `github.com` account. More information will follow.
- The will be a "check list" at the end of this presentation.
- By popular demand, the weekly schedule will be repeated on the following page!

# Operating Systems 2020-2[2]) — **PJJ from HOME**
# ZOOM: International [Tue 08-10] — A/Matrix [Tue 10-12]

| **Week** | **Schedule** | **Topic** | **OSC10**[1]) |
|---|---|---|---|
| Week 00 | 15 Sep - 21 Sep 2020 | Overview 1, Virtualization & Scripting | Ch. 1, 2, 18. |
| Week 01 | 22 Sep - 28 Sep 2020 | Overview 2, Virtualization & Scripting | Ch. 1, 2, 18. |
| Week 02 | 29 Sep - 05 Oct 2020 | Security, Protection, Privacy, & C-language. | Ch. 16, 17. |
| Week 03 | 06 Oct - 12 Oct 2020 | File System & FUSE | Ch. 13, 14, 15. |
| Week 04 | 13 Oct - 19 Oct 2020 | Addressing, Shared Lib, & Pointer | Ch. 9. |
| Week 05 | 20 Oct - 26 Oct 2020 | Virtual Memory | Ch. 10. |
| Week 06 | 27 Oct - 16 Nov 2020 | Concurrency: Processes & Threads | Ch. 3, 4. |
| | 29 Oct 2020 | Maulid Nabi | |
| Week 07 | 17 Nov - 23 Nov 2020 | Synchronization & Deadlock | Ch. 6, 7, 8. |
| Week 08 | 24 Nov - 30 Nov 2020 | Scheduling + W06/W07 | Ch. 5. |
| Week 09 | 01 Dec - 07 Dec 2020 | Storage, Firmware, Bootloader, & Systemd | Ch. 11. |
| Week 10 | 08 Dec - 16 Dec 2020 | I/O & Programming | Ch. 12. |
| | 09 Dec 2020 | Pil Kada | |

[1]Silberschatz et. al.: **Operating System Concepts**, $10^{th}$ Edition, 2018.

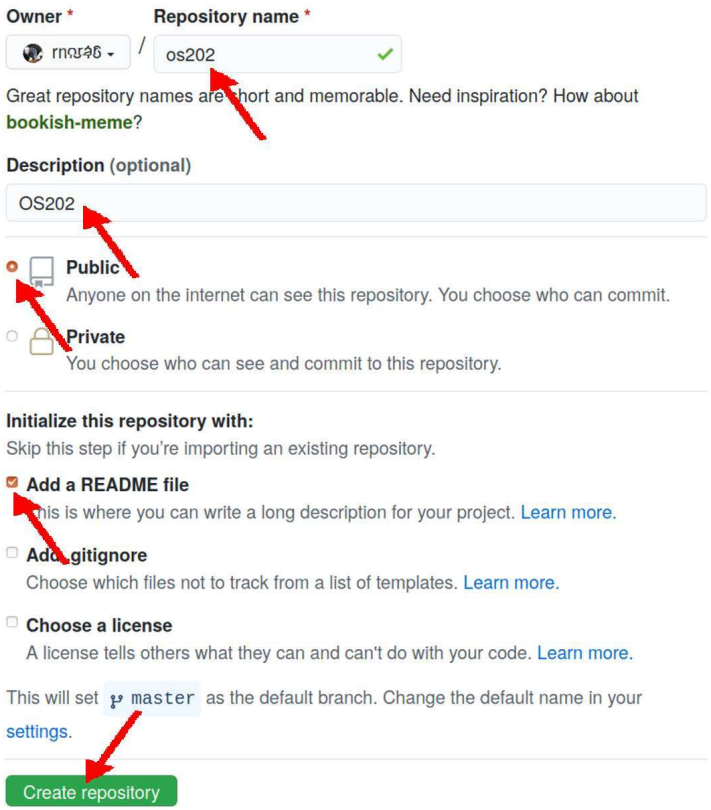
[2]This information will be on **EVERY** page two (2) of this course material.

# 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 also http://codex.cs.yale.edu/avi/os-book/OS10/.
- **Resources**
  - **SCELE** — https://scele.cs.ui.ac.id/course/view.php?id=3020. The enrollment key is **XXX**.
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/
  - **Problems** — https://rms46.vlsm.org/2/: 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).
- **Build your own Virtual Guest** https://osp4diss.vlsm.org/

# Week 00 Assignment #1: Public Repository "os202"

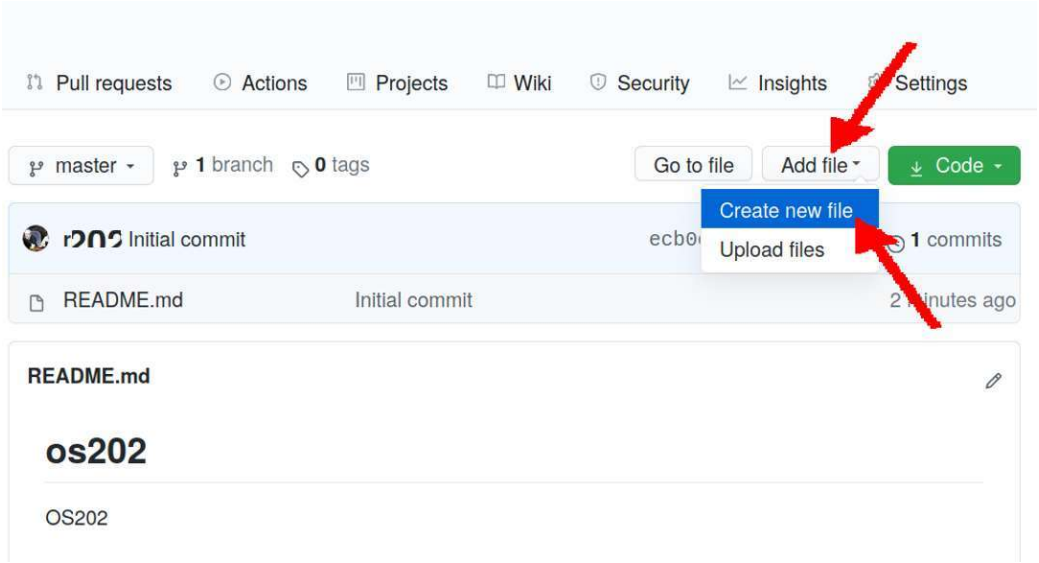
- Visit `github.com`
  - (IF NO ACCOUNT) **SIGN UP** (`https://github.com/join`).
  - (ELSE) **SIGN IN** (`https://github.com/login`).
  - Create a new repository ("os202"):



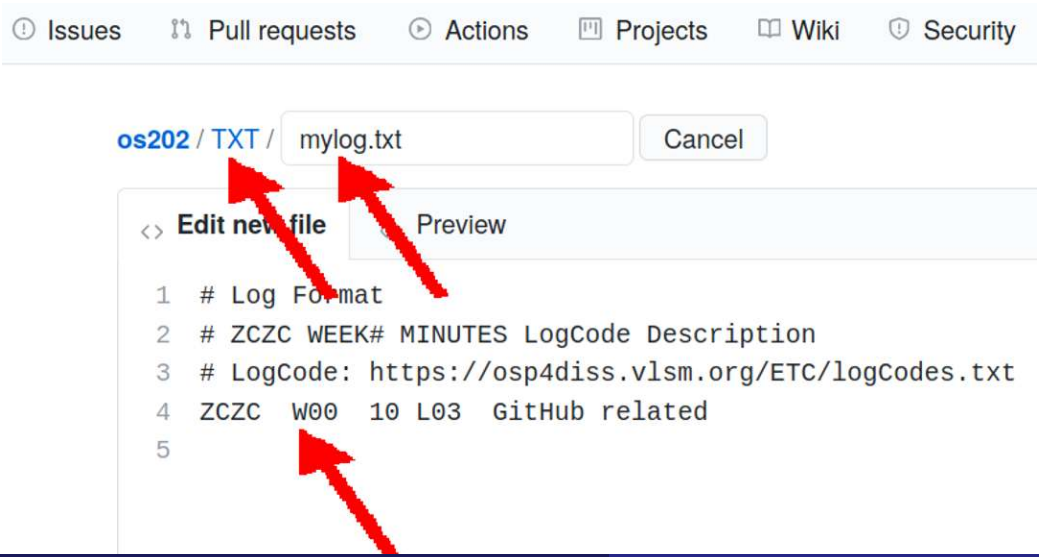

    - **Repository name**: "os202" (all small).
    - **Public**: Anyone can see this repository.

# Week 00 Assignment #2: Start Week 00 Log (1)

`Add file` → `Create new file`

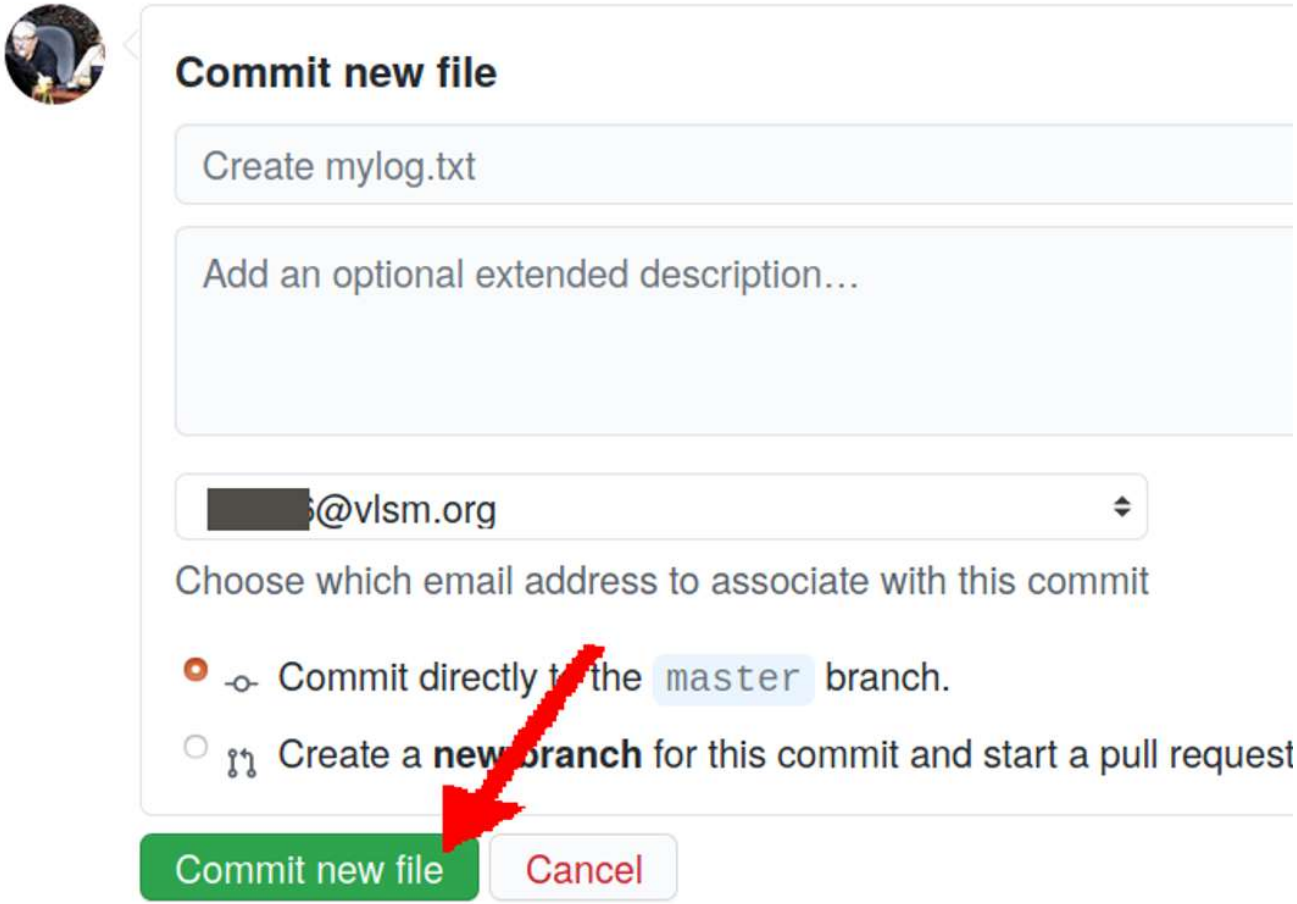



**Folder/File**: `"TXT/mylog.txt"` (Eg. Week-00 10 minutes doing GitHub (L03))



# Week 00 Assignment #2: Start Week 00 Log (2)

Commit a new file



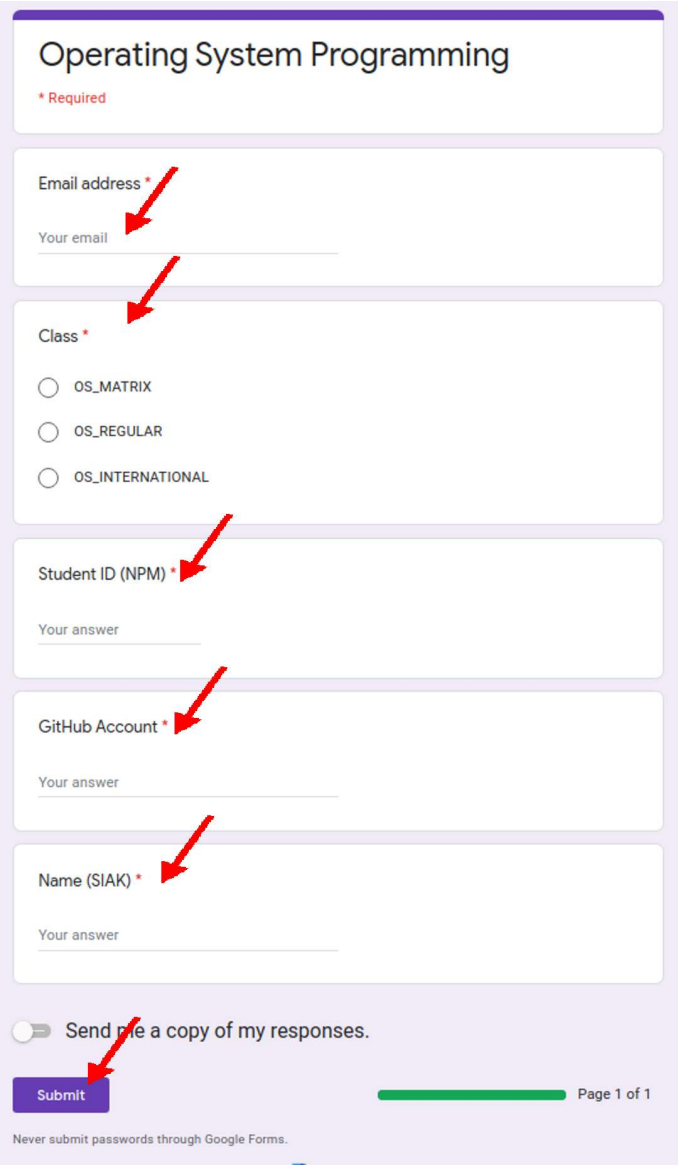
# Week 00 Assignment #3 & #4

- Assignment #3: Create Your GitHub Page
  - Do GSGS[1]. Find out how to create your GitHub Page!
  - Eg. if your GitHub account is "`cbkadal`" (Cicak Bin Kadal).
  - The GitHub repository will be:
    - `https://github.com/cbkadal/os202/`.
  - The GitHub Page will be:
    - `https://akunGitHub.github.io/os202/`.
- Assignment #4: The Weekly TOP 10 LIST
  - **Read any** recent and decent Operating Systems TextBook chapter(s) that are related to this week topic.
  - Write a **TOP 10 LIST** about what points of this week that you think are important.
  - This Week 00 TOP 10 LIST will be merged with the TOP 10 LIST of Week 01.

[1] Google Sana (There) Google Sini (Here)

# Week 00 Assignment #5: Course Registration

Fill in your Email Address, Class, Student ID, GitHub Account, and Name. Details (Google Form link) will be available at **SCELE**



# 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 2013

# 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 2013

# 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 2013

# Week 01 Overview II: Topics[1]

- Types of virtualization (including Hardware/Software, OS, Server, Service, Network)
- Paging and virtual memory
- Virtual file systems
- 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 2013

# 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 virtualized infrastructure. [Familiarity]

[1] Source: ACM IEEE CS Curricula 2013

# Week 02 Security & Protection: Topics[1]

- Overview of system security
- Policy/mechanism separation
- Security methods and devices
- Protection, access control, and authentication
- Backups

[1] Source: ACM IEEE CS Curricula 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

# 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 2013

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

- Storage
- Storage Arrays
- BIOS
- Loader
- Systemd

[1] Source: ACM IEEE CS Curricula 2013

# 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 2013

# 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 2013

# 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]

# Week 00: Summary

- What is an Operating Systems?
  - 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)

- 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 6 working days (RMS).
- Prepare the weekly MEMO as completely as possible. You should have mastered the material at the beginning of the week (RMS).
- Study the Operating System Concept book which deals with the material will be discussed that week (MIM). Make a summary of 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 up to 6 sheets of MEMOs for the midterm (UTS) and up to 5 sheets of MEMOs for the final term (UAS) (RMS).
- You should understand every single line of the "**DEMOS**" (MHP).

# TIPS (2)

- You should ask **the lecturer** or anyone, anything you do not understand (TA).

# Special Thanks

**Special thanks** for the early version of this writing to:

Anisha Inas Izdihar (AII), Benedictus Alvin (BA), Ibnu Sofian Firdaus (ISF), Irmanpen Panjaitan (IP), Ivana Irene Thomas (IIT), Michael Giorgio Wirawan (MGW), Muhammad Afkar (MA), Muhammad Hanif Pratama (MHP), Muhammad Iqbal Mahendra (MIM), 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);

See also `https://rms46.vlsm.org/2/221.pdf`.

# Week 00: Check List

- Visit **Pengumuman Akademis:** `https://scele.cs.ui.ac.id/`
- Check this out:
  - **Starting Point:** `https://os.vlsm.org/`
  - **SCELE:** `https://scele.cs.ui.ac.id/course/view.php?id=3020`
  - **OSP4DISS:** `https://osp4diss.vlsm.org/`
- Week 00: Assignment (more details in **os00.pdf**).
  1. Create GitHub Public Repository "os202".
  2. Start Week 00 Log.
  3. Create your "os202" GitHub Page.
  4. Write a **TOP 10 LIST** about what points that you think are important. This Week 00 TOP 10 LIST will be merged with the TOP 10 LIST of Week 01.
  5. Course Registration (check **SCELE**).
- The "Assignment Day" is every Thursday morning.
- The "Deadline Day" will be the beginning of the week after.
- Revisit/add your own Weekly Log.
- **Study Hard**! No more TABULA RASA. Jolan Tru!
- This page is `https://os.vlsm.org/Slides/check00.pdf`.

# The End

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