CSF2600505 Sistem Operasi
CSGE602055 Operating Systems
Week 00: Overview 1

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

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

REV350 27-Sep-2021

OS212[4]): Operating Systems 2021 - 2

OS A	OS B	OS C	OS INT
Every first day of the Week, **Quiz#1:** (07:40-07:50) and **Quiz#2:** 07:20-07:40			
Monday/Thursday 13:00 — 14:40 14:00 — finish	Monday/Thursday 15:00 — 16:40[1] 16:00 — finish	Monday/Thursday 13:00 — 14:40 13:00 — 14:40	Monday/Wednesday 08:00 — 09:40 09:00 — finish

Week	**Schedule & Deadline[2])**	**Topic**	**OSC10[3])**
Week 00	30 Aug - 05 Sep 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	06 Sep - 12 Sep 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	13 Sep - 19 Sep 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	20 Sep - 26 Sep 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	27 Sep - 03 Oct 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	04 Oct - 10 Oct 2021	Virtual Memory	Ch. 10.
Week 06	11 Oct - 17 Oct 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	01 Nov - 07 Nov 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	08 Nov - 14 Nov 2021	Scheduling + W06/W07	Ch. 5.
Week 09	15 Nov - 21 Nov 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	22 Nov - 28 Nov 2021	I/O & Programming	Ch. 12.

[1]) **OS B:** Week00-Week05 (RMS); Week06-Week10 (MAM).

[2]) The **DEADLINE** of Week 00 is 05 Sep 2021, whereas the **DEADLINE** of Week 01 is 12 Sep 2021, and so on...

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

[4]) 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 https://www.os-book.com/OS10/.
- **Resources**
  - **SCELE OS212** — https://scele.cs.ui.ac.id/course/view.php?id=3268. The enrollment key is **XXX**.
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/: 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/
  - **DOIT** — https://doit.vlsm.org/001.html

Agenda

Agenda (2)

How to contact the Lecturer

- **Always introduce yourself**. State your "`GitHubAccount`", "`Name`", "`Student ID`", and "`OS class`".
- Post a question/query on **SCELE OS212** — (The enrollment key is **XXX**): https://scele.cs.ui.ac.id/course/view.php?id=3268.
- For SIAK related questions, use email: (Subject:**[HELP]**) `operatingsystems(AT)vlsm.org`.
- **DO NOT** send an email for assignment-related questions.

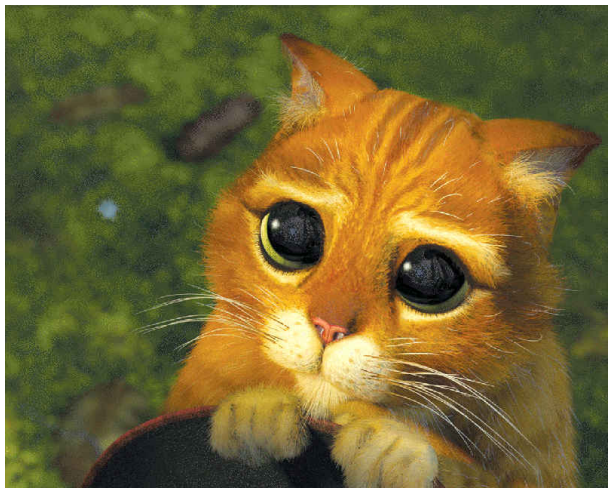


Figure: Never ever whine and pretend like this[1]!

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

Assessment

- **4 SKS** (Units) means 12 hours per week!
  - You need to log your weekly activities!
- **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 page 2.
  - See the checklist 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:** up to 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/Log/.

[1] Terms and conditions apply — void where prohibited by law.

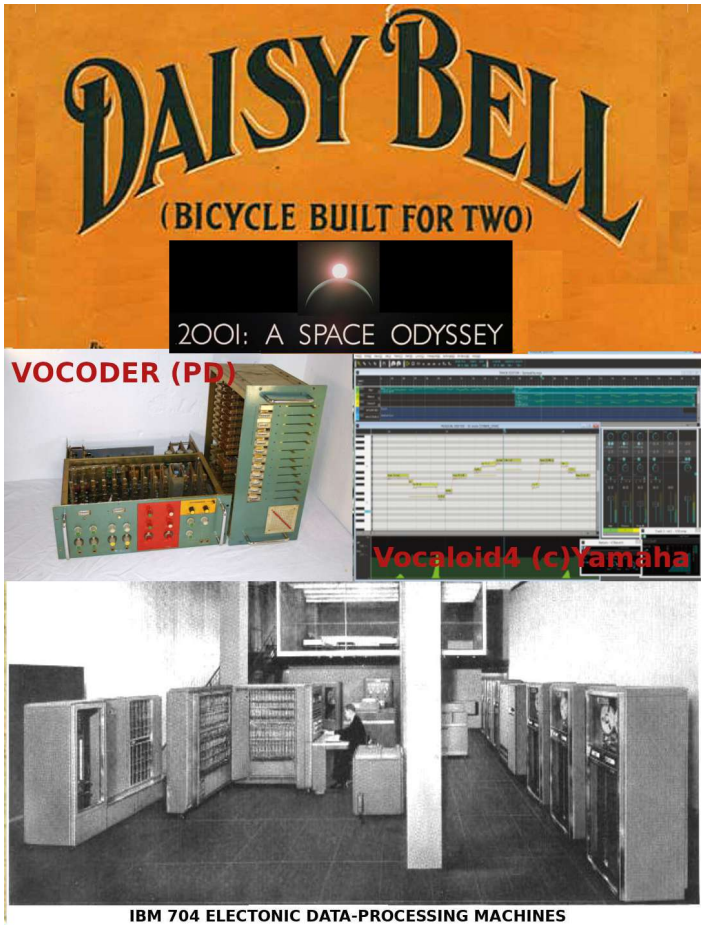
The Three-Strikes Rule

- All major academic rules violations will be handled directly by the Faculty of Computer Science, University of Indonesia.
- "Accidents" may 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!

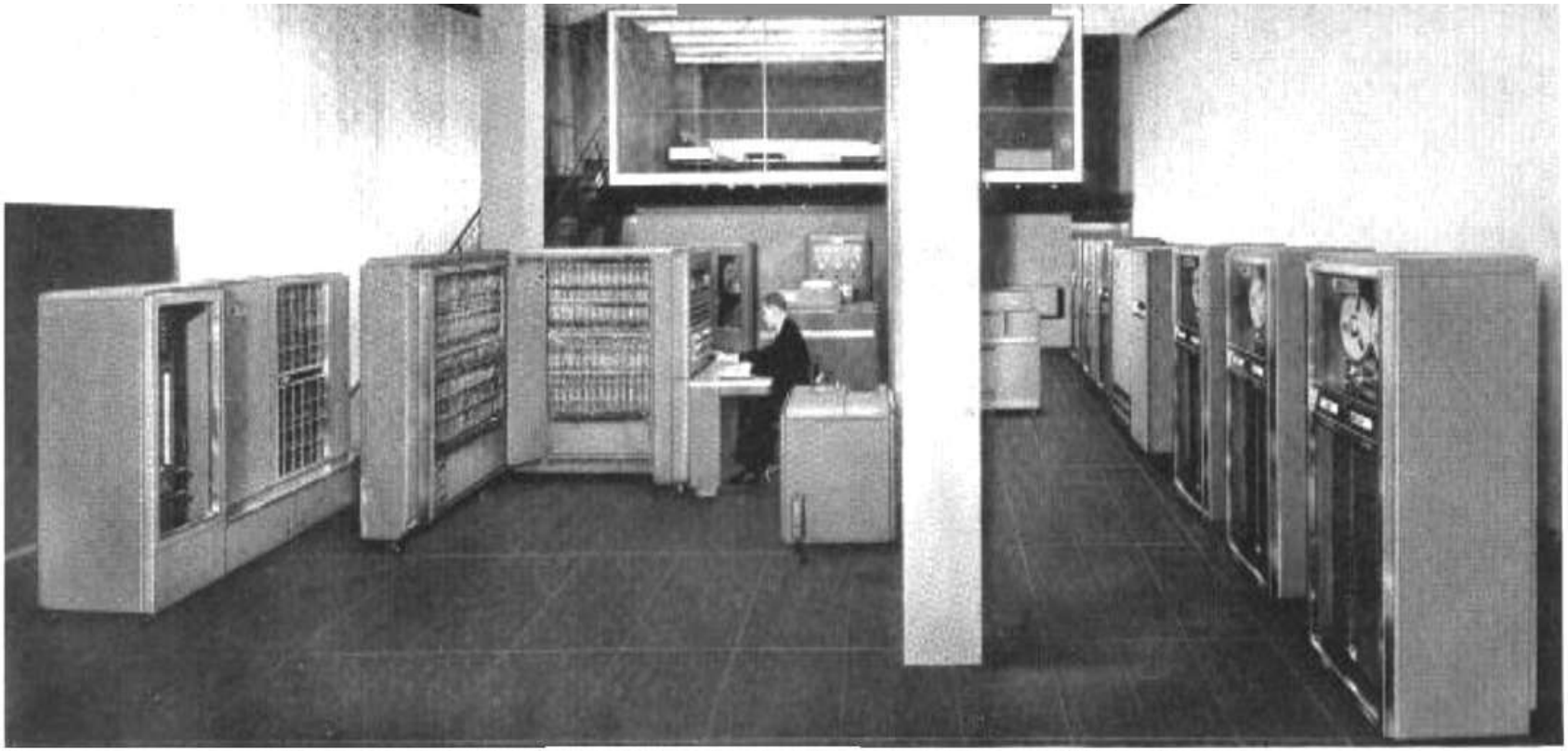
Prelude: Daisy Bell – Bicycle Built for Two



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!

A choir (emulation) of VOCODER (pre WW2), IBM704 (1950s) and Vocaloid4 (2014). See also the classical movie "**2001: A Space Odyssey**" and YouTube: 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

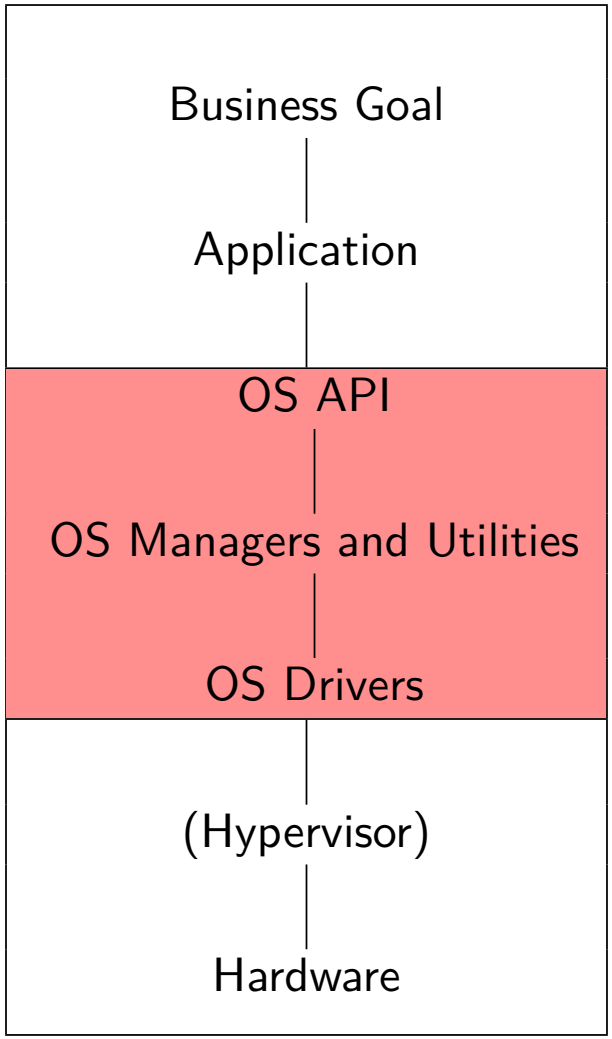
Estimate price (2020 value): Rp. 8,000,000.

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!

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: $9AB\ CDEF_{16} = 1001\ 1010\ 1011\ \ 1100\ 1101\ 1110\ 1111_2$

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

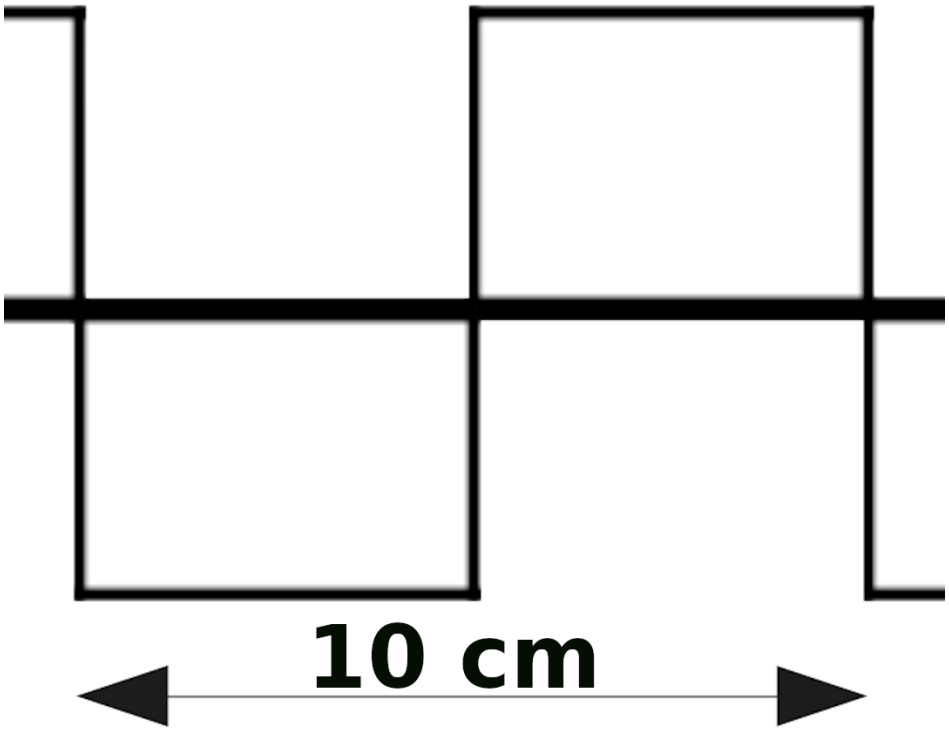


Figure: What is the length of a 3 GHz signal?

Physics 101: Safe Distance for 3 GHz

Figure: Safe Distance

Physics 101: Serial vs. Parallel Transmission

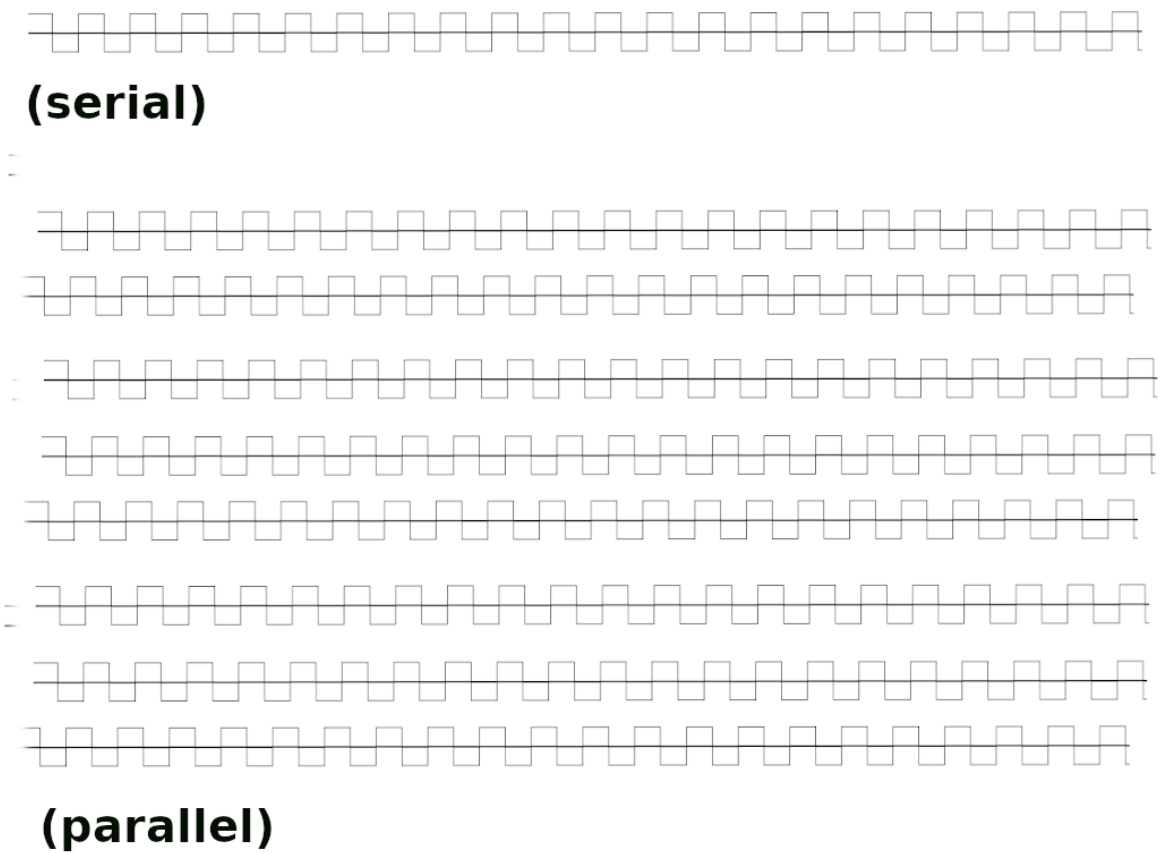
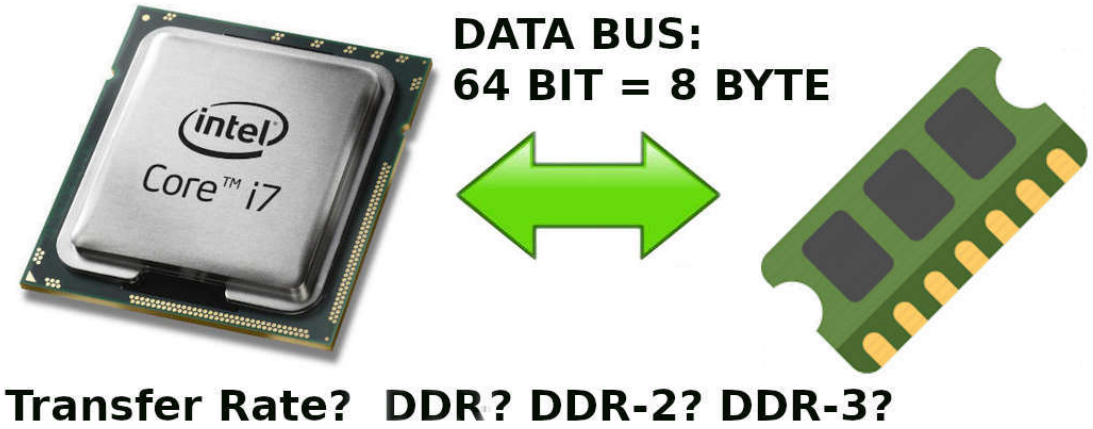


Figure: Serial vs. Parallel Transmission

- Serial: Longer Distance, easy to implement.
- Parallel: Faster, but not easy.

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-4 = DDR-3 with a better clock rate.

CPU: SuperVisor Mode

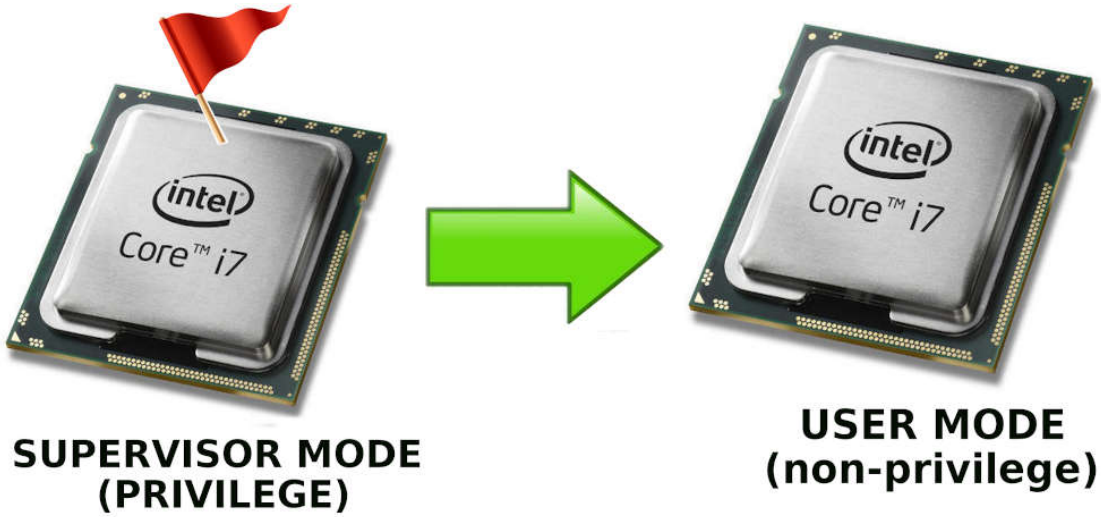


Figure: SuperVisor (Privilege) Mode to User Mode

- 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

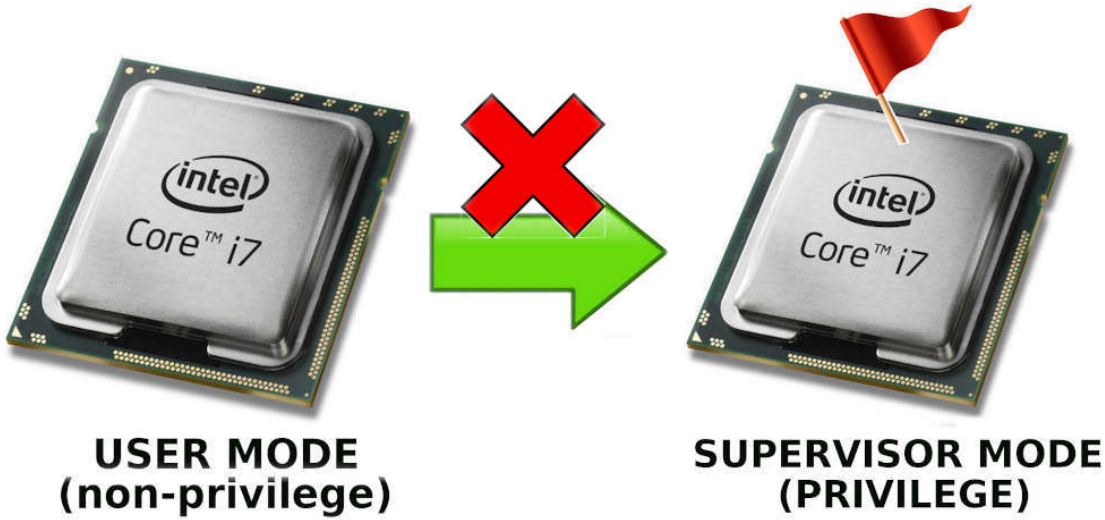


Figure: User Mode to SuperVisor (Privilege)

- 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?

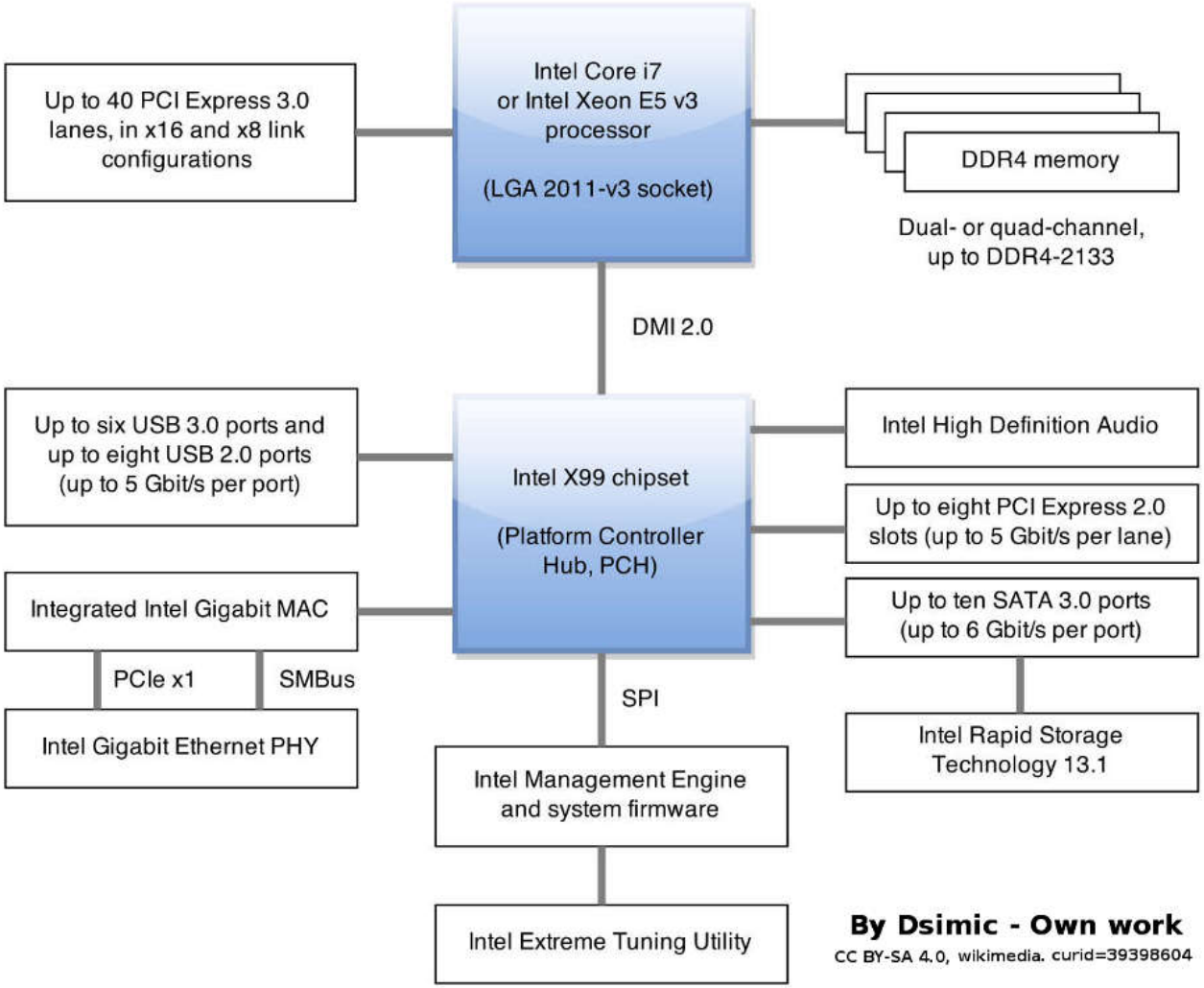


Figure: Block Diagram

What is an APIC?!

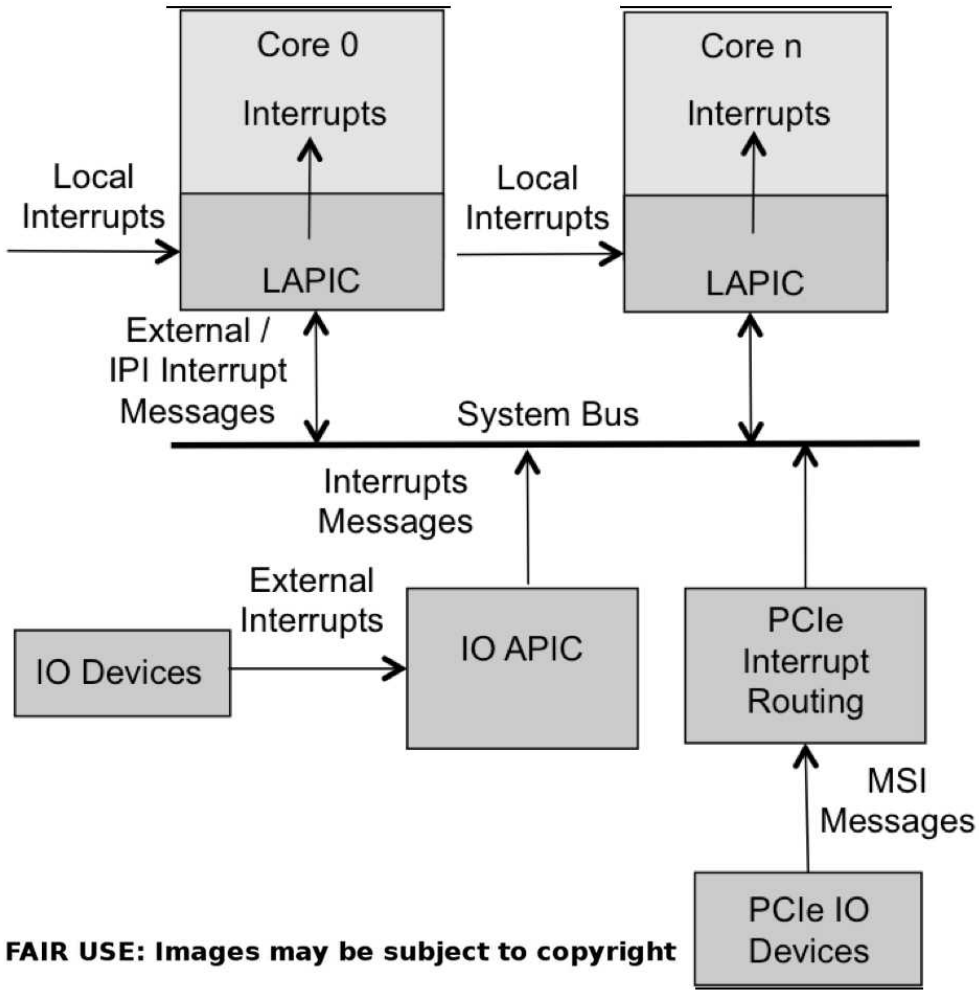


Figure: APIC (Advanced Programmable Interrupt Controller)

And, what is "Interrupt Handling"?

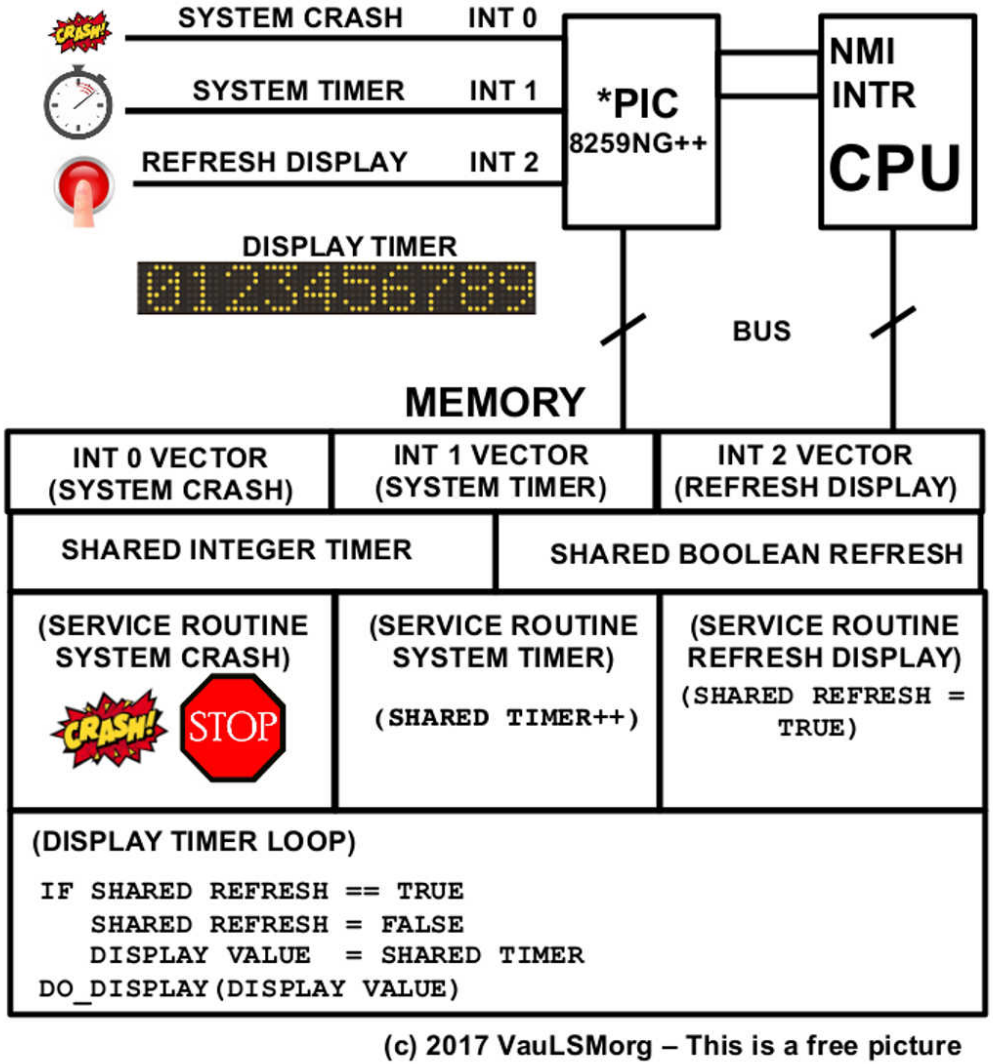


Figure: Interrupt Handling with PIC (Programmable Interrupt Controller)

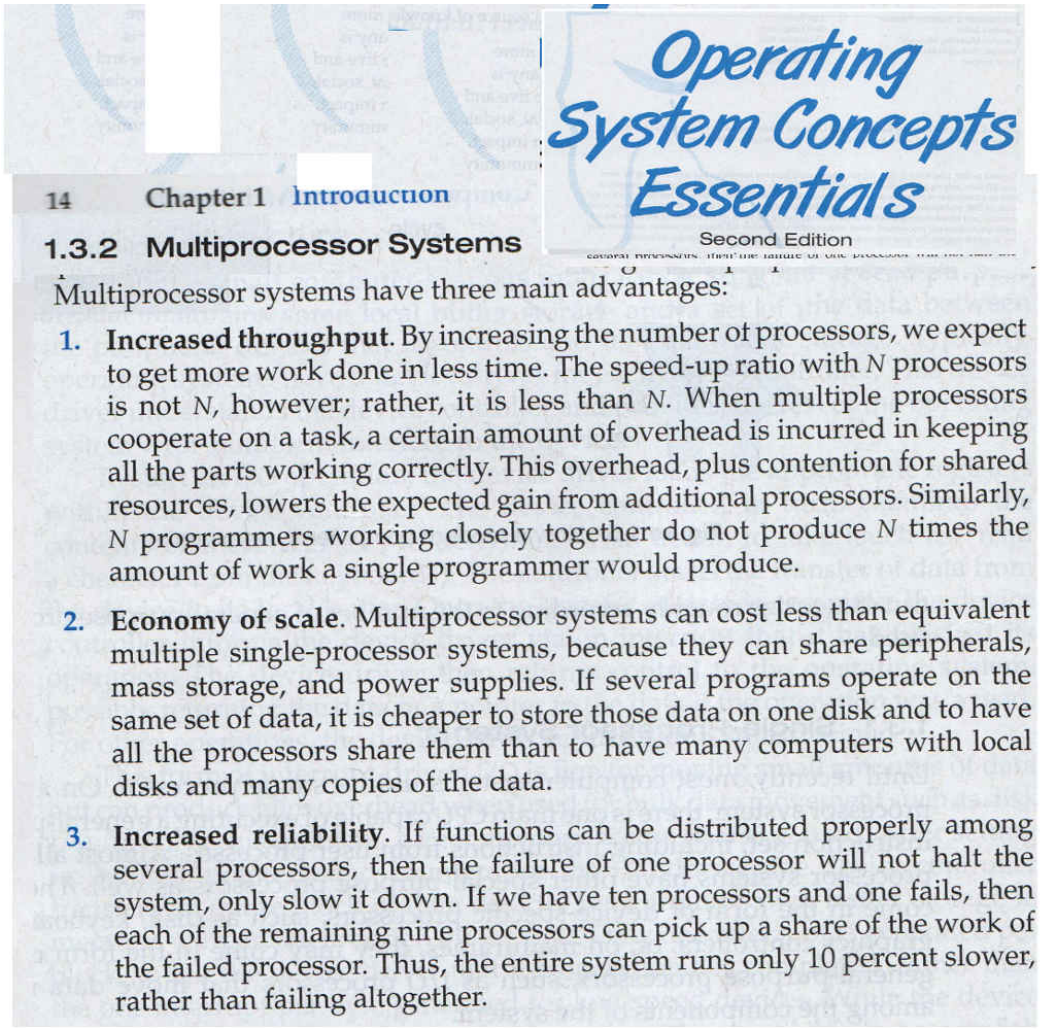
The Operating System Managers

- 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.

Figure: **T / F** The advantages of a multiprocessor system include: increased throughput, economy of scale, and increased reliability (Week 00 2016-1).

Week 00: More QUIZ Examples

- **TRUE/FALSE**
  The best way to get any help is to send an email to `operatingsystems@vlsm.org`.
- **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

- There will be 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.
- You need to run "VirtualBox" on a computer with more than 4GB RAM and up to 100 GB disk space.
- Each assignment deadline will be by the end of that "week". The weekly schedule will always be on the page **[2]**.
- 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 need to sign up for a https://github.com/ account. More information will follow.
- There will be a "checklist" at the end of this presentation.
- By popular demand, the weekly schedule will be repeated on the following page!

OS212[4]): Operating Systems 2021 - 2

OS A	OS B	OS C	OS INT
Every first day of the Week, **Quiz#1:** (07:40-07:50) and **Quiz#2:** 07:20-07:40			
Monday/Thursday 13:00 — 14:40 14:00 — finish	Monday/Thursday 15:00 — 16:40[1] 16:00 — finish	Monday/Thursday 13:00 — 14:40 13:00 — 14:40	Monday/Wednesday 08:00 — 09:40 09:00 — finish

Week	**Schedule & Deadline[2])**	**Topic**	**OSC10[3])**
Week 00	30 Aug - 05 Sep 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	06 Sep - 12 Sep 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	13 Sep - 19 Sep 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	20 Sep - 26 Sep 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	27 Sep - 03 Oct 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	04 Oct - 10 Oct 2021	Virtual Memory	Ch. 10.
Week 06	11 Oct - 17 Oct 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	01 Nov - 07 Nov 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	08 Nov - 14 Nov 2021	Scheduling + W06/W07	Ch. 5.
Week 09	15 Nov - 21 Nov 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	22 Nov - 28 Nov 2021	I/O & Programming	Ch. 12.

[1]) **OS B:** Week00-Week05 (RMS); Week06-Week10 (MAM).

[2]) The **DEADLINE** of Week 00 is 05 Sep 2021, whereas the **DEADLINE** of Week 01 is 12 Sep 2021, and so on...

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

[4]) 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 https://www.os-book.com/OS10/.
- **Resources**
  - **SCELE OS212** — https://scele.cs.ui.ac.id/course/view.php?id=3268. The enrollment key is **XXX**.
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/: 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/
  - **DOIT** — https://doit.vlsm.org/001.html

Week 00 Assignments

- Assignment #1: Public Repository
  - https://github.com/ — Account
- Assignment #2: Start Week 00 Log
  - This is a **4 UNIT(SKS) or 12 hours/week** course. Are you sure that you have only spend 5 minutes this week???
  - E.g. see cbkadal's log at https://cbkadal.github.io/os212/TXT/mylog.txt
  - For log codes, see https://osp4diss.vlsm.org/ETC/logCodes.txt
- Assignment #3: Create Your GitHub Page
  - E.g. cbkadal's page at: https://cbkadal.github.io/os212/
- Assignment #4: Course Registration
  - The Google Form link will be available at **SCELE**.
- Assignment #5: Reading/Watching Assignments
  - What defines an Operating System? https://rahmatm.samik-ibrahim.vlsm.org/2021/07/what-defines-operating-system.html

Week 00 Assignment #1: Public Repository

- This is neither programming nor a web course. However, assignments will be submitted to GitHub and will be displayed on GitHub Page.
- Visit https://github.com:
  - **SIGN UP**, if you have no account: (https://github.com/join).
    - Preferably, use all lower case characters for your GitHub account.
  - Else, **SIGN IN**: (https://github.com/login).
- Create a new repository (or repo):
  - **Repository name**, e.g:
    - "os212" for year 2021-2 (odd semester 2021/2022),
    - "os221" for year 2022-1 (even semester 2021/2022),
    - "os222" for year 2022-2 (odd semester 2022/2023),
    - "os231" for year 2023-1 (even semester 2022/2023),
    - etc.
    - **Note**: For "os", use lowercase. Do not use uppercase!
  - **Description**: (e.g.) Operating Systems 2021-2 (Odd Semester 21/22).
  - **Public**: Anyone can see this repository.
  - A simple **README.md** file.

Week 00 Assignment #1 (cont)

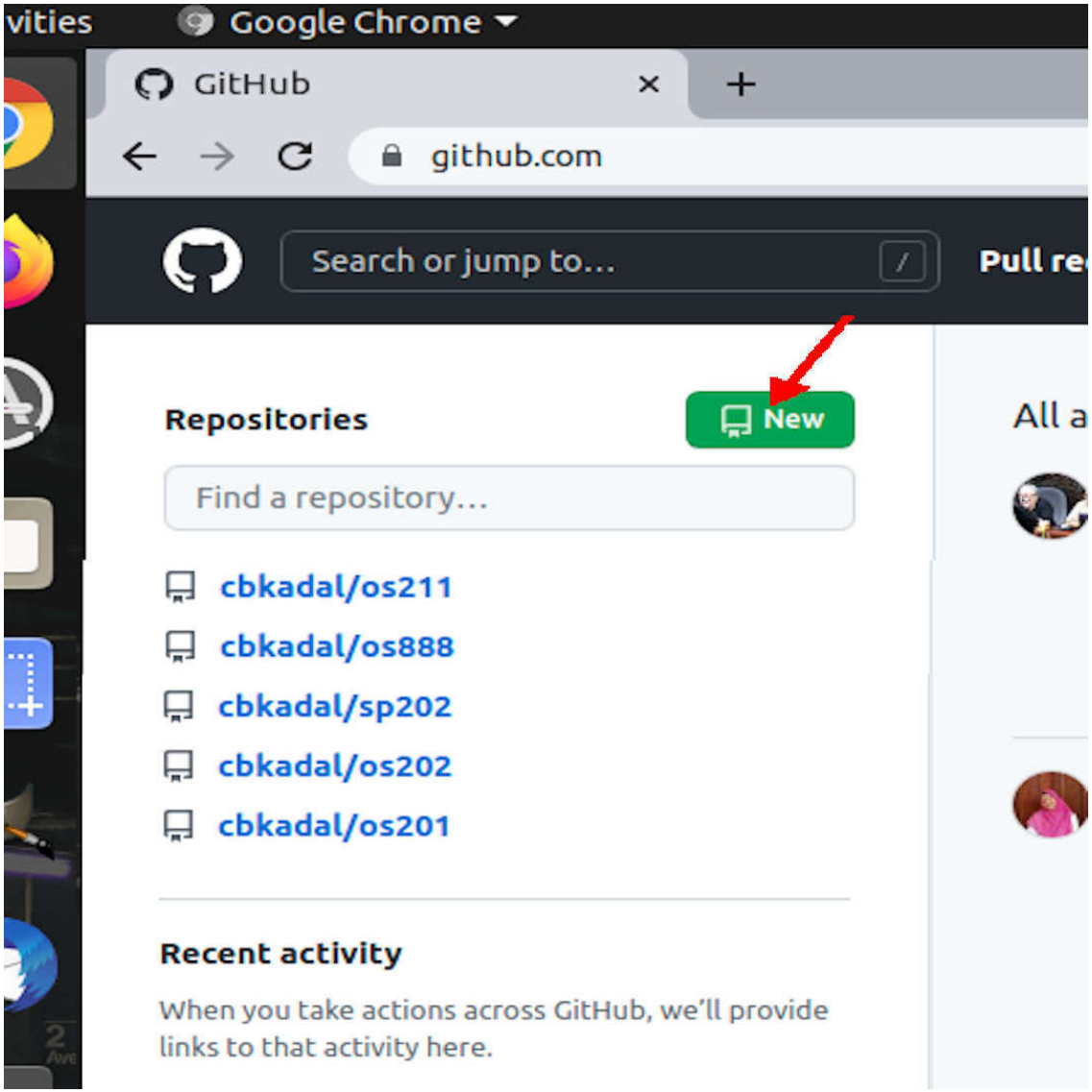


Figure: Create a new repository

Week 00 Assignment #2 (cont)

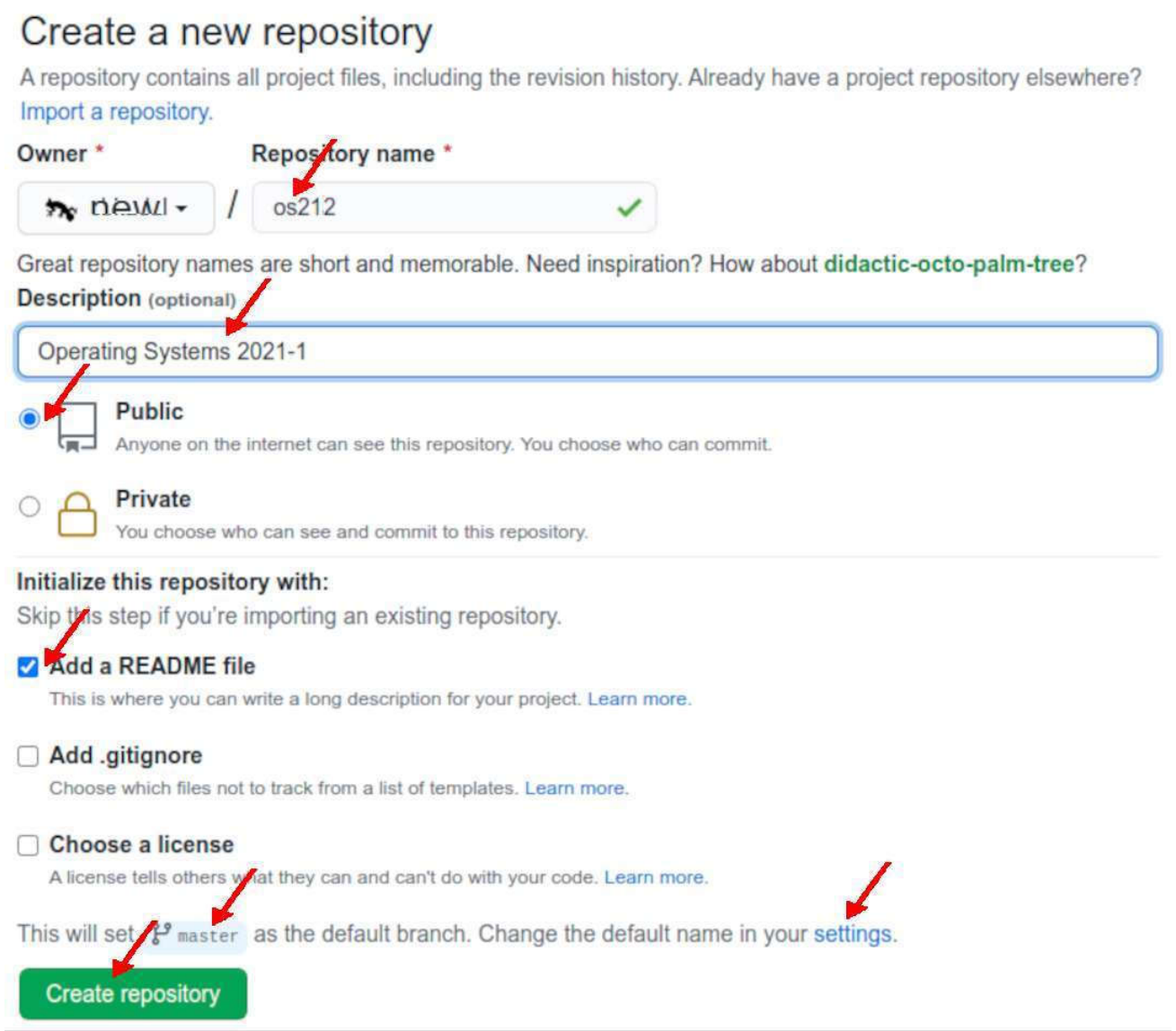


Figure: Public Repository in this example is "os212"

Week 00 Assignment #3 (cont)

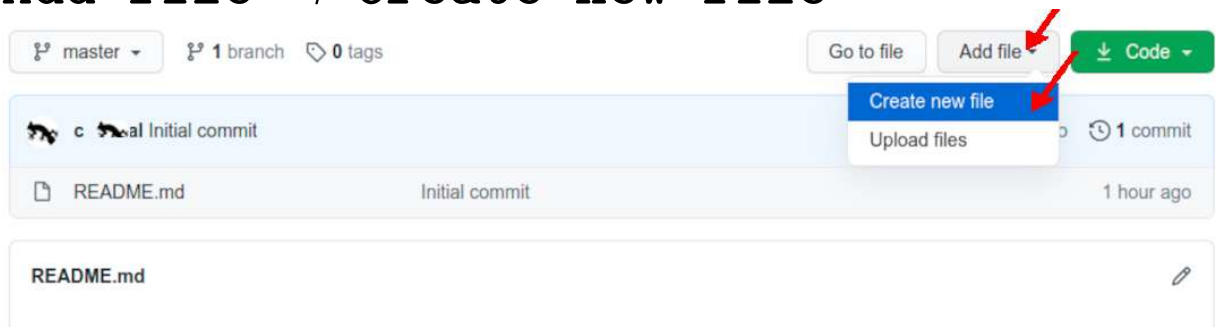
- The GitHub Default Branch Name Is Now "`main`"
  - To be "politically correct," GitHub has changed the default branch name from "`master`" to "`main`."
  - Many past examples here have been using the "`master`" branch name. Therefore — for being consistent — the "`master`" branch name will continue to be used.
  - Feel free to use either "`main`" or "`master`." However, once it has been chosen, you should not alter your branch name.
  - To change the default branch name, click "`settings`."

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

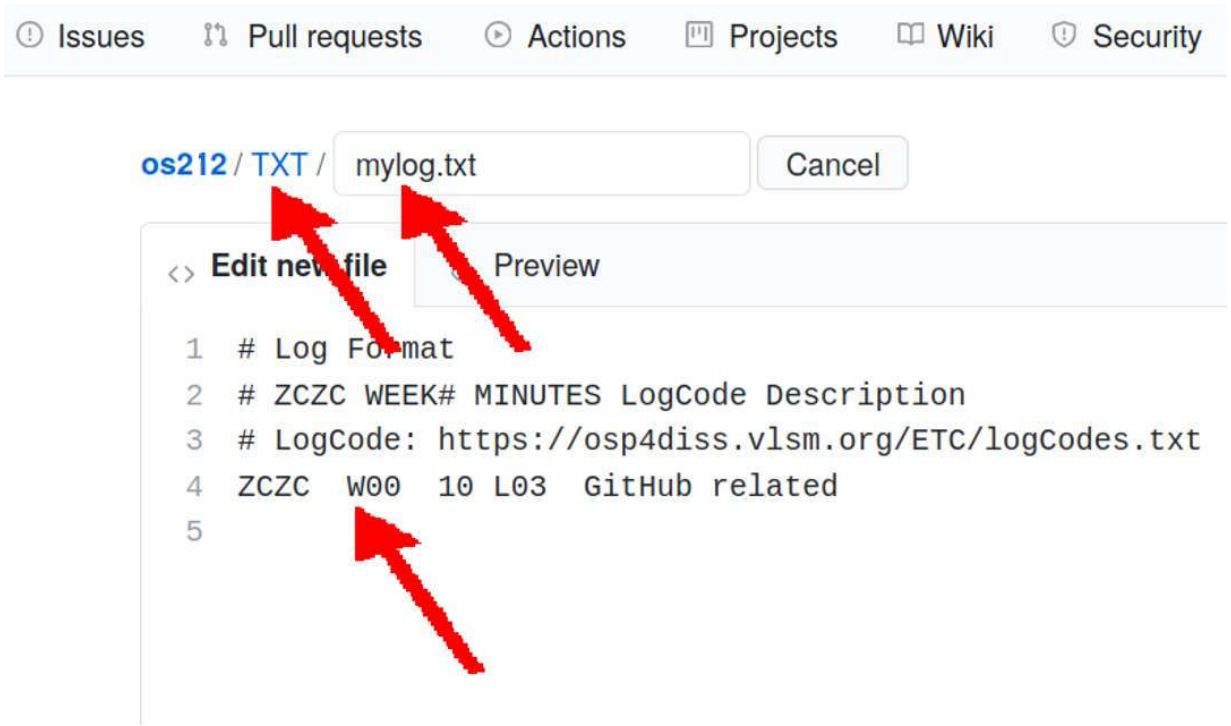
```
# REV06 Wed 07 Jul 2021 20:15:07 WIB
# https://osp4diss.vlsm.org/ETC/logCodes.txt
# ZCZC WEEK# MINUTES LogCode Description

L00 General, etc.
L01 SCELE/Discord related
L02 ZOOM meetings related
L03 GitHub related
L04 GitHub Pages related
L05 Quiz related
L06 References/Books/Documents/GSGS related
L07 Demo related
L08 AsDos: asking, etc.
L09 Assignment in General
L10 Assignment #00
L11 Assignment #01
L12 Assignment #02
L13 Assignment #03
L14 Assignment #04
L15 Assignment #05
L16 Assignment #06
L17 Assignment #07
L18 Assignment #08
L19 Assignment #09
L20 Assignment #10
L21 Trying something
L22 Quiz
L23 Linux CLI including tar, etc.
(...)
L86 House chore, including helping mom, pap, uncle, aunty, jaga warung, jualan kue, etc.
L87 Else that is not related with this Operating Systems class.
L99 Other (...)
```

Week 00 Assignment #2: (cont)

`Add file` → `Create new file`

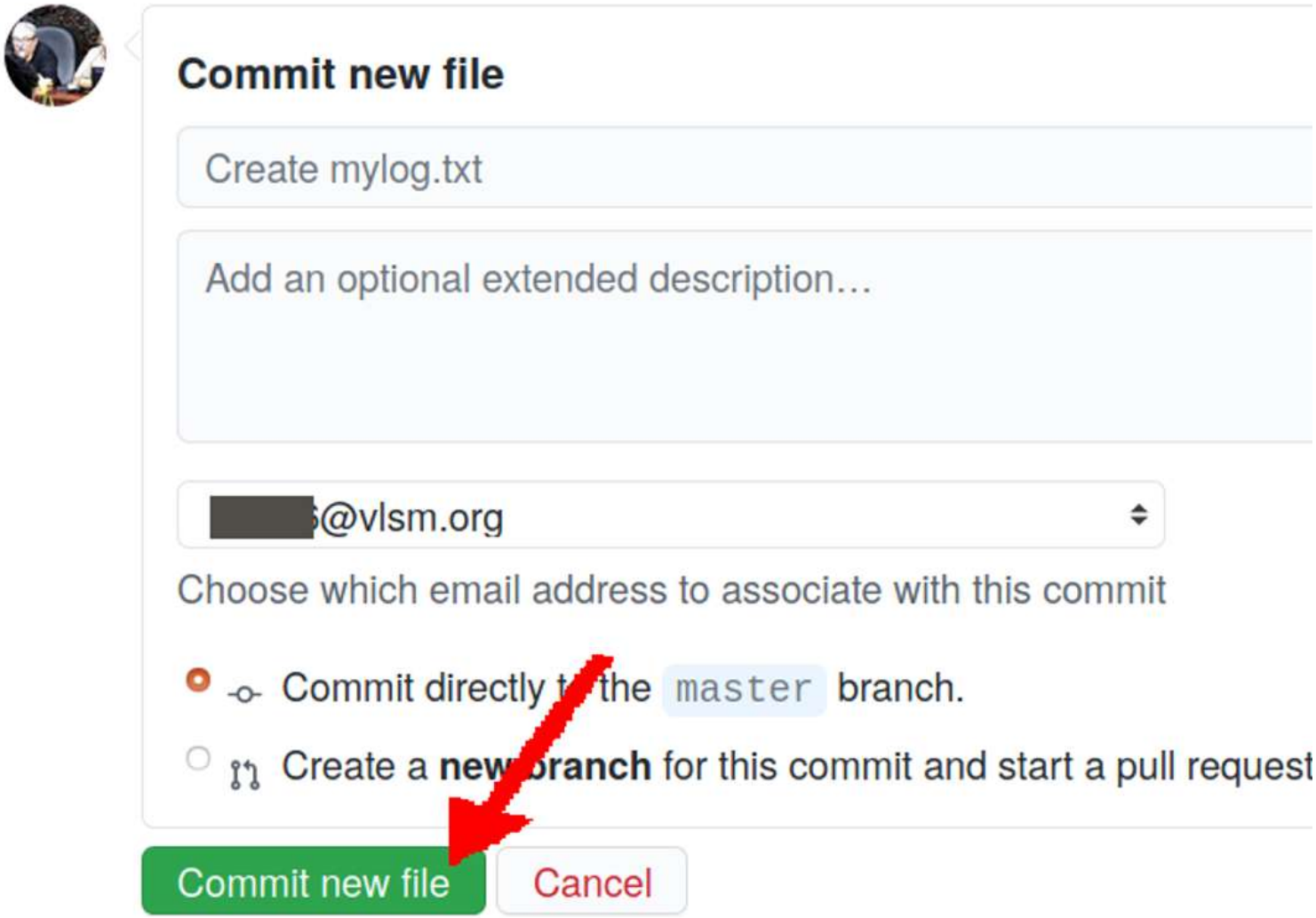


Folder/File: "`TXT/mylog.txt`" (Eg. Week-00 10 minutes doing GitHub (L03))



Week 00 Assignment #2: (cont)

Commit a new file

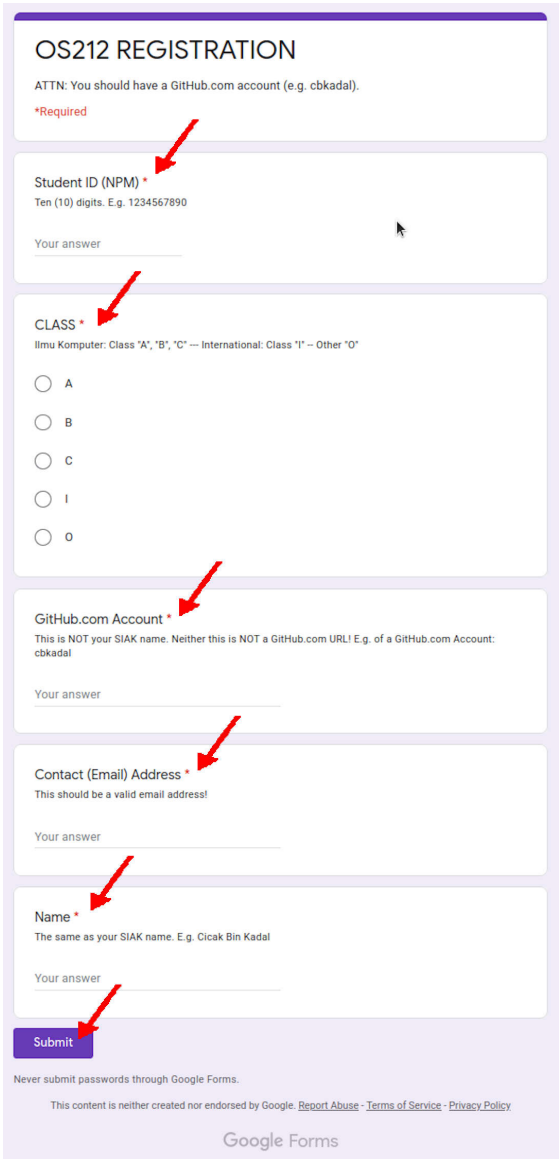


Week 00 Assignment #3: Create Your GitHub Page

- Do GSGS[1].
- Find out how to create your GitHub Page! E.g., if your GitHub account is "cbkadal" (Cicak Bin Kadal).
  - The GitHub repository will be:
    - https://github.com/cbkadal/os212/.
  - The GitHub Page will be:
    - https://cbkadal.github.io/os212/.
  - See also https://doit.vlsm.org/001.html
- Your GitHub Page should be as light as possible!
  - **The total repo size: less than 1 Mbytes**.
  - Do not apply any Jekyll theme.
  - Suggested images size: less than 50 Kbytes.
  - No external CSS and fonts.
  - Google Analytics is allowed.

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

Week 00 Assignment #4: Course Registration



- You need a Google Account to fill this Google Form[a].
- The Google Form link will be available at **SCELE**.
- Fill in with the email address that you normally use. It does not have to be Gmail.
- GitHub Account example: "`cbkadal`".
- "`github.com/cbkadal/`" **is not** a GitHub account.
- Use your SIAK name, **NOT** your NICK name.
- If you make a **mistake**, revisit the Google Form page.

[a] The form content is subject to change

Week 00 Assignment #5: Reading Assignment

Get familiar with the following documents. But, there is no need to memorize them!

- What defines an Operating System?
  https://rahmatm.samik-ibrahim.vlsm.org/2021/07/what-defines-operating-system.html
- (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10^{th} Edition, 2018, https://www.os-book.com/OS10/, chapter 1, 2, 18.
- GNU/Linux Tutorial
  https://osp4diss.vlsm.org/Welcome2GNULinux.html
- More GNU/Linux
  https://osp4diss.vlsm.org/osp-115.html
- Operating Systems: Visual Metaphor (Udacity)
  https://www.youtube.com/playlist?list=PLqoiDr4YpRdm_nzFhCDuj74P8ul5z7SdO

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]

- 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 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 the virtualized infrastructure. [Familiarity]

[1] Source: ACM IEEE CS Curricula 2013

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 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 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.
- 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).
- Prepare the weekly MEMO as thoroughly 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 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 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 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);

Week 00: Check List (Deadline: 5 Sep 2021).

- ☐ 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=3268
  - ☐ **OSP4DISS:** https://osp4diss.vlsm.org/
  - ☐ **LFS:** http://www.linuxfromscratch.org/lfs/view/stable/
- ☐ Week 00: Assignment (more details in **os00.pdf**).
  1. Create GitHub Public Repository "os212".
  2. Start Week 00 Log.
  3. Create your "os212" GitHub Page.
  4. Course Registration
     - https://scele.cs.ui.ac.id/mod/forum/discuss.php?d=30285.
  5. Reading Assignment
- ☐ Revisit/add your own Weekly Log.
- ☐ **Study Hard**! No more TABULA RASA. Qapla!
- ☐ 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.