UNIVERSITAS
INDONESIA

Veritas, Probitas, Iustitia
EST. 1849

BUKU RANCANGAN PENGAJARAN (BPR)

MATA KULIAH

SISTEM OPERASI

oleh

Rahmat M. Samik-Ibrahim

Fakultas Ilmu Komputer

Universitas Indonesia

Rev. 01 - 18-Sep-2020

***** DRAFT *****

Preface

I pledge allegiance to the Operating Systems of the GNU/Linux, and to the kernel for which it stands, one Bourne Again Shell under systemd, indivisible, with GNU Library and Justice For All! This is the Operating Systems (OS202) Course Guideline, a BRP wannabe.

Jolan Tru!
Pamulang, September 2020

Rahmat M. Samik-Ibrahim

1 General Information OS202

1.	Course Level	:	S1 Regular, S1 International, S2 Matriculation
2.	Course Name	:	Operating Systems
3.	Course Number	:	CSGE602055
4.	Credits	:	4 Units (SKS) or ”3?” or ”Terms and Conditions May Apply?!”
5.	ZOOM Schedule	:	Tuesday, 08:00 - 10:00 (English) Tuesday, 10:00 - 12:00 (Indonesian)
6.	Quiz Time Slot	:	Thursday, 09:00 - 11:00
7.	URLs	:	SCELE: `https://scele.cs.ui.ac.id/course/view.php?id=3020` OS: `https://os.vlsm.org/` OSP: `https://osp4diss.vlsm.org/`
8.	Lecture(s)	:	Rahmat M. Samik-Ibrahim
9.	Prerequisites	:	CSCM601252 Pengantar Organisasi Komputer or CSIM601251 Dasar Dasar Arsitektur Komputer.
10.	Teaching Method	:	The Feyerabend Method
11.	References	:	Any recent / decent Operating Systems book. Eg. (OSC10) Silberschatz et. al.: Operating System Concepts, 10^{th}ed., 2018. See also `http://codex.cs.yale.edu/avi/os-book/OS10`.
12.	Course Description	:	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. This course is student-centered where responsibility is in the hands of the students. Students are expected to be prepared for the class meeting. Students will have a thorough understanding of how GNU/Linux provides services by using a Command Line Interface.

2 Course Topics and Learning Outcome

(**Source: CUT and PASTE** from the ACM IEEE CS Curricula 2013)

2.1 Week 00: Overview I

2.1.1 Topics

- 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

2.1.2 Outcomes

- 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]

2.2 Week 01: Overview II

2.2.1 Topics

- 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.

2.2.2 Outcomes

- 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]

2.3 Week 02: Security & Protection

2.3.1 Topics

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

2.3.2 Outcomes

- 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]

2.4 Week 03: File System & FUSE

2.4.1 Topics

- 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

2.4.2 Outcomes

- 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]

2.5 Week 04: Addressing

2.5.1 Topics

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

2.5.2 Outcomes

- 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]

2.6 Week 05: Virtual Memory

2.6.1 Topics

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

2.6.2 Outcomes

- 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]

2.7 Week 06: Concurrency

2.7.1 Topics

- 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)

2.7.2 Outcomes

- 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]
- 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]

2.8 Week 07: Synchronization & Deadlock

2.8.1 Topics

- 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.

2.8.2 Outcomes

- 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]

2.9 Week 08: Scheduling

2.9.1 Topics

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

2.9.2 Outcomes

- 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]

2.10 Week 09: Storage, Firmware, Bootloader, & Systemd

2.10.1 Topics

- Storage
- Storage Arrays
- BIOS
- Loader
- Systemd

2.10.2 Outcomes

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

2.11 Week 10: I/O & Programming

2.11.1 Topics

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

2.11.2 Outcomes

- 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]

3 Course Schedule

Week	Schedule	Topic	OSC10
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	

4 Assessment

- **4 Units** (SKS) means 12 hours per week! There shall be weekly activity logs.
- **11** (weekly) assigments @ 11.11 points.
  - Assignments will vary from week to week.
- 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 your the is between 50.00 and 55.00.
- Check SIAK regularly at `https://academic.ui.ac.id/`.

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

5 Quiz Example (True/False)

T / F

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

Answer: **T (True)**

Note: this was taken from Schilbersatz et.at., 2010, Operating System Concepts Essentials, chapter 1, page 14.

6 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.
- The final grade will be reduced for the third warning.
- The final grade will be reduced to ”D” for the fourth warning.
- Five (5) or more warnings will be considered as a major academic-rules violation.

7 Disclaimer

Our apology, most of the ”Course Topics and Learning Outcome” parts are rip-offs/cut and paste from the ACM IEEE CS Curricula 2013. Some other parts of this document are based on ”Google Here and Google There” You are free to share, copy and redistribute the material of this site in any medium or format for any purpose, even commercially. Please give appropriate credit, provide a link to the page, and indicate if changes were made. You may do so in any reasonable manner. You should not apply legal terms or technological measures, that legally restrict others from doing anything that is permitted here.

Document Revision: 01 - 18-Sep-2020

Jolan Tru[2] and `May the fork() be with you!`[3].

[2] This is a Romulan Rip-Off

[3] This is a Jedi Rip-Off