

# Operating System Course Report - First Half of the Semester

B class

October 1, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Course Overview</b>	<b>3</b>
2.1	Objectives . . . . .	3
2.2	Course Structure . . . . .	3
<b>3</b>	<b>Topics Covered</b>	<b>4</b>
3.1	Basic Concepts and Components of Computer Systems . . . . .	4
3.2	System Performance and Metrics . . . . .	4
3.3	System Architecture of Computer Systems . . . . .	4
3.4	Process Description and Control . . . . .	4
3.5	Scheduling Algorithms . . . . .	5
3.6	Process Creation and Termination . . . . .	5
3.7	Introduction to Threads . . . . .	5
3.8	File Systems . . . . .	5
3.9	Input and Output Management . . . . .	6
3.10	Deadlock Introduction and Prevention . . . . .	6
3.11	User Interface Management . . . . .	6
3.12	Virtualization in Operating Systems . . . . .	6
3.12.1	Pengertian Virtualisasi Sistem Operasi . . . . .	6
3.12.2	Jenis Jenis Virtualisasi pada Sistem Operasi . . . . .	8
3.12.3	Komponen Virtualisasi Sistem Operasi . . . . .	8
3.12.4	Hypervisor dan Virtual Machines . . . . .	8
3.12.5	Contoh Penerapan Virtualisasi pada Sistem Operasi . . . . .	8
3.12.6	Keuntungan dan Kerugian Virtualisasi . . . . .	8
3.12.7	Kesimpulan . . . . .	10
<b>4</b>	<b>Assignments and Practical Work</b>	<b>11</b>
4.1	Assignment 1: Process Scheduling . . . . .	11
4.2	Assignment 2: Deadlock Handling . . . . .	11
4.3	Assignment 3: Multithreading and Amdahl's Law . . . . .	11
4.4	Assignment 4: Simple Command-Line Interface (CLI) for User Interface Management . . . . .	11
4.5	Assignment 5: File System Access . . . . .	12
<b>5</b>	<b>Conclusion</b>	<b>12</b>

# 1 Introduction

This report summarizes the topics covered during the first half of the Operating System course. It includes theoretical concepts, practical implementations, and assignments. The course focuses on the fundamentals of operating systems, including system architecture, process management, CPU scheduling, and deadlock handling.

## 2 Course Overview

### 2.1 Objectives

The main objectives of this course are:

- To understand the basic components and architecture of a computer system.
- To learn process management, scheduling, and inter-process communication.
- To explore file systems, input/output management, and virtualization.
- To study the prevention and handling of deadlocks in operating systems.

### 2.2 Course Structure

The course is divided into two halves. This report focuses on the first half, which covers:

- Basic Concepts and Components of Computer Systems
- System Performance and Metrics
- System Architecture of Computer Systems
- Process Description and Control
- Scheduling Algorithms
- Process Creation and Termination

- Introduction to Threads
- File Systems
- Input and Output Management
- Deadlock Introduction and Prevention
- User Interface Management
- Virtualization in Operating Systems

## **3 Topics Covered**

### **3.1 Basic Concepts and Components of Computer Systems**

This section explains the fundamental components that make up a computer system, including the CPU, memory, storage, and input/output devices.

### **3.2 System Performance and Metrics**

This section introduces various system performance metrics used to measure the efficiency of a computer system, including throughput, response time, and utilization.

### **3.3 System Architecture of Computer Systems**

Describes the architecture of modern computer systems, focusing on the interaction between hardware and the operating system.

### **3.4 Process Description and Control**

Processes are a central concept in operating systems. This section covers:

- Process states and state transitions
- Process control block (PCB)
- Context switching

### **3.5 Scheduling Algorithms**

This section covers:

- First-Come, First-Served (FCFS)
- Shortest Job Next (SJN)
- Round Robin (RR)

It explains how these algorithms are used to allocate CPU time to processes.

### **3.6 Process Creation and Termination**

Details how processes are created and terminated by the operating system, including:

- Process spawning
- Process termination conditions

### **3.7 Introduction to Threads**

This section introduces the concept of threads and their relation to processes, covering:

- Single-threaded vs. multi-threaded processes
- Benefits of multithreading

### **3.8 File Systems**

File systems provide a way for the operating system to store, retrieve, and manage data. This section explains:

- File system structure
- File access methods
- Directory management

### **3.9 Input and Output Management**

Input and output management is key for handling the interaction between the system and external devices. This section includes:

- Device drivers
- I/O scheduling

### **3.10 Deadlock Introduction and Prevention**

Explores the concept of deadlocks and methods for preventing them:

- Deadlock conditions
- Deadlock prevention techniques

### **3.11 User Interface Management**

This section discusses the role of the operating system in managing the user interface. Topics covered include:

- Graphical User Interface (GUI)
- Command-Line Interface (CLI)
- Interaction between the user and the operating system

### **3.12 Virtualization in Operating Systems**

Virtualization allows multiple operating systems to run concurrently on a single physical machine. This section explores:

#### **3.12.1 Pengertian Virtualisasi Sistem Operasi**

Virtualisasi adalah teknologi yang memungkinkan beberapa sistem operasi (OS) berjalan secara bersamaan pada satu mesin fisik. Ini dicapai dengan membuat lingkungan virtual yang memisahkan perangkat keras dari perangkat lunak. Setiap OS yang berjalan di atas mesin virtual (VM) beroperasi seolah-olah memiliki sumber daya perangkat keras yang berdedikasi,

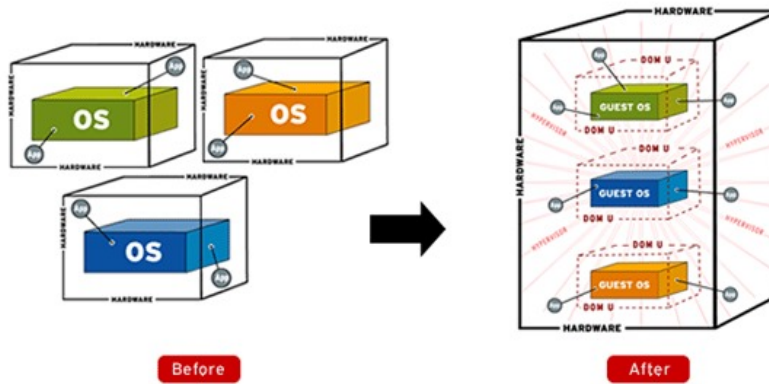


Figure 1: Cara Kerja Virtualisasi

meskipun pada kenyataannya perangkat keras tersebut dibagi oleh beberapa VM.

Pada ilustrasi gambar, terlihat dua kondisi:

- Sebelum Virtualisasi (Before): Di sisi kiri gambar, kita melihat bahwa setiap sistem operasi (OS) berjalan langsung di atas perangkat keras yang terpisah. Pendekatan ini kurang efisien karena setiap OS membutuhkan perangkat kerasnya sendiri, yang menyebabkan pemborosan sumber daya fisik ketika perangkat keras tidak dimanfaatkan secara maksimal.
- Setelah Virtualisasi (After): Di sisi kanan gambar, terlihat bahwa setelah virtualisasi diterapkan, satu perangkat keras fisik dapat menjalankan beberapa OS sekaligus melalui penggunaan hypervisor. Hypervisor ini menciptakan beberapa mesin virtual (VM) yang memungkinkan berbagai sistem operasi berjalan di atas satu perangkat keras secara bersamaan. Hal ini meningkatkan efisiensi penggunaan sumber daya dan fleksibilitas dalam pengelolaan sistem.

Dengan cara ini, virtualisasi memungkinkan pemanfaatan sumber daya yang lebih baik, meningkatkan isolasi antar sistem, dan mempermudah proses pemeliharaan serta pengelolaan infrastruktur komputasi.

### 3.12.2 Jenis Jenis Virtualisasi pada Sistem Operasi

### 3.12.3 Komponen Virtualisasi Sistem Operasi

### 3.12.4 Hypervisor dan Virtual Machines

### 3.12.5 Contoh Penerapan Virtualisasi pada Sistem Operasi

### 3.12.6 Keuntungan dan Kerugian Virtualisasi

#### 1. Keuntungan dari Virtualisasi

Virtualisasi menawarkan banyak keuntungan baik secara operasional maupun finansial, yang membuatnya menjadi kunci dalam komputasi enterprise dan pengembangan perangkat lunak. Berikut ini adalah beberapa keuntungan utama dari virtualisasi:

- **Menggunakan Perangkat Keras yang Ada dengan Lebih Baik:** Prosesor modern berkembang dengan cepat, dari 8-bit hingga 64-bit, serta jumlah inti prosesor yang terus bertambah. Dengan menjalankan beberapa mesin virtual pada satu server fisik, pengguna dapat mengoptimalkan penggunaan perangkat keras yang ada. Sistem multi-core dapat menjalankan beberapa mesin virtual di setiap inti prosesor yang berbeda, memaksimalkan efisiensi sumber daya.
- **Mengurangi Harga Perangkat Keras:** Virtualisasi memungkinkan pengguna untuk menambah server atau layanan tanpa harus membeli perangkat keras baru, yang pada akhirnya mengurangi biaya secara signifikan.
- **Mengurangi Infrastruktur IT:** Server fisik membutuhkan listrik, ruang, dan pendinginan. Dengan virtualisasi, kebutuhan akan listrik, ruang, dan pendinginan bisa dikurangi karena satu server fisik dapat menjalankan banyak mesin virtual.
- **Menyederhanakan Administrasi Sistem:** Dengan menjalankan beberapa mesin virtual pada satu perangkat keras fisik, kesehatan sistem bisa lebih mudah dipantau, dan administrasi sistem dapat disederhanakan, terutama dalam hal migrasi atau cloning ketika terjadi kesalahan perangkat keras.
- **Meningkatkan Uptime dan Mempercepat Failure Recovery:** Portabilitas mesin virtual memudahkan pemindahan dari



satu server fisik ke server lain jika ada masalah pada perangkat keras, tanpa mengganggu operasi yang sedang berjalan.

- **Menyederhanakan Ekspansi Kapasitas:** Mesin virtual dapat dipindahkan dengan mudah ke perangkat keras baru dengan spesifikasi yang lebih tinggi, seperti prosesor yang lebih cepat, memori yang lebih besar, atau kapasitas jaringan yang lebih tinggi.
- **Dukungan Perangkat Lunak Asli yang Lebih Mudah:** Dengan virtualisasi, pengguna dapat menjalankan sistem operasi lama di partisi logis, sehingga memungkinkan perangkat lunak lama tetap berjalan meskipun sistem operasi sudah diperbarui.
- **Menyederhanakan Pengembangan dan Testing Aplikasi:** Pengembangan dan pengujian aplikasi pada berbagai sistem operasi menjadi lebih mudah dengan virtualisasi, karena mesin virtual dapat menjalankan berbagai sistem operasi tanpa perlu perangkat keras khusus.

## 2. Kerugian Virtualisasi

Walaupun virtualisasi menawarkan banyak keuntungan, ada beberapa kerugian yang perlu diperhatikan, di antaranya:

- **Satu Titik Kesalahan:** Jika satu perangkat keras fisik digunakan untuk menjalankan banyak server virtual, kerusakan pada perangkat keras tersebut bisa memengaruhi semua server virtual yang berjalan di atasnya. Oleh karena itu, diperlukan strategi cadangan dan monitoring untuk meminimalkan risiko ini.
- **Kepadatan Saluran Jaringan:** Jika satu server fisik menjalankan banyak mesin virtual yang membutuhkan akses jaringan intensif, ini bisa menyebabkan kepadatan saluran jaringan dan menurunkan kinerja jaringan. Solusi untuk masalah ini adalah dengan menggunakan beberapa kartu jaringan pada host fisik.
- **Meningkatkan Kompleksitas Jaringan dan Debugging:** Virtualisasi jaringan yang kompleks dapat menambah kesulitan dalam hal desain dan pengelolaan infrastruktur, serta memakan lebih banyak waktu dalam proses debugging.
- **Meningkatkan Kompleksitas Administrasi:** Administrasi bisa menjadi lebih rumit jika sistem manajemen terdistribusi yang di-

gunakan tidak mendukung mesin virtual. Oleh karena itu, diperlukan perencanaan dan pemilihan alat yang tepat untuk mendukung virtualisasi secara efisien.

### **3.12.7 Kesimpulan**

Virtualisasi merupakan teknologi penting dalam komputasi modern, yang memungkinkan pemanfaatan sumber daya secara lebih efisien, meningkatkan fleksibilitas, dan menyederhanakan pengelolaan infrastruktur. Meskipun memiliki beberapa kelemahan seperti potensi satu titik kesalahan dan peningkatan kompleksitas jaringan, manfaat yang diperoleh jauh lebih besar, terutama dalam hal penghematan biaya dan peningkatan uptime sistem.

## References

- [1] Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson.
- [2] Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. John Wiley & Sons.
- [3] Singh, G. (2019). *Virtualization Essentials*. John Wiley & Sons.
- [4] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., & Warfield, A. (2003). *Xen and the art of virtualization*. In *Proceedings of the nineteenth ACM symposium on Operating systems principles* (pp. 164-177).

## 4 Assignments and Practical Work

### 4.1 Assignment 1: Process Scheduling

Students were tasked with implementing various process scheduling algorithms (e.g., FCFS, SJN, and RR) and comparing their performance under different conditions.

### 4.2 Assignment 2: Deadlock Handling

In this assignment, students were asked to simulate different deadlock scenarios and explore various prevention methods.

### 4.3 Assignment 3: Multithreading and Amdahl's Law

This assignment involved designing a multithreading scenario to solve a computationally intensive problem. Students then applied \*Amdahl's Law\* to calculate the theoretical speedup of the program as the number of threads increased.

### 4.4 Assignment 4: Simple Command-Line Interface (CLI) for User Interface Management

Students were tasked with creating a simple \*CLI\* for user interface management. The CLI should support basic commands such as file manipulation

(creating, listing, and deleting files), process management, and system status reporting.

## 4.5 Assignment 5: File System Access

In this assignment, students implemented file system access routines, including:

- File creation and deletion
- Reading from and writing to files
- Navigating directories and managing file permissions

## 5 Conclusion

The first half of the course introduced core operating system concepts, including process management, scheduling, multithreading, and file system access. These topics provided a foundation for more advanced topics to be covered in the second half of the course.