

Operating System Course Report - First Half of the Semester

A class

October 10, 2024

Contents

1	Introduction	3
2	Course Overview	3
2.1	Objectives	3
2.2	Course Structure	3
3	Topics Covered	4
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
3.3.1	Fungsi Arsitektur Sistem Komputer	4
3.3.2	Komponen Utama Sistem Arsitektur Komputer	4
3.3.3	Tipe atau Jenis dari Sistem Arsitektur Komputer	4
3.4	Process Description and Control	11
3.5	Scheduling Algorithms	11
3.6	Process Creation and Termination	11
3.7	Introduction to Threads	12
3.8	File Systems	12
3.9	Input and Output Management	12
3.10	Deadlock Introduction and Prevention	12
3.11	User Interface Management	13
3.12	Virtualization in Operating Systems	13
4	Assignments and Practical Work	13
4.1	Assignment 1: Process Scheduling	13
4.1.1	Group 1	13
4.2	Assignment 2: Deadlock Handling	14
4.3	Assignment 3: Multithreading and Amdahl's Law	14
4.4	Assignment 4: Simple Command-Line Interface (CLI) for User Interface Management	14
4.5	Assignment 5: File System Access	14
5	Conclusion	15

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.

3.3.1 Fungsi Arsitektur Sistem Komputer

3.3.2 Komponen Utama Sistem Arsitektur Komputer

3.3.3 Tipe atau Jenis dari Sistem Arsitektur Komputer

1. Arsitektur *Von Neumann*

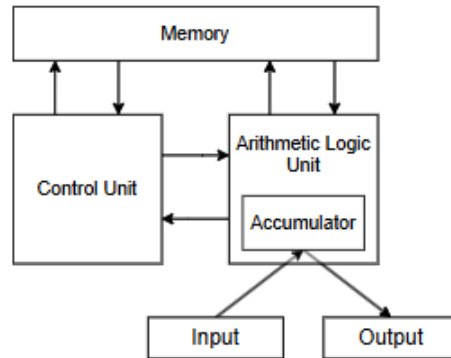


Figure 1: Diagram Arsitektur *Von Neumann*

2. Arsitektur Harvard

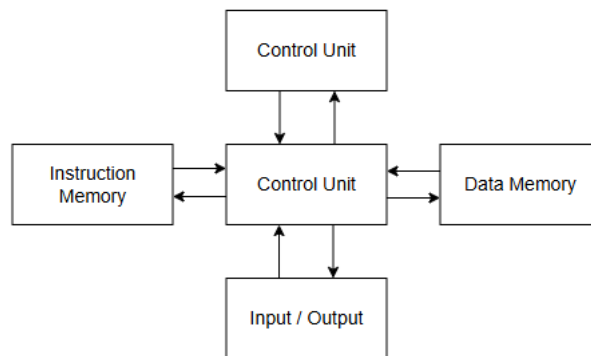


Figure 2: Diagram Arsitektur Harvard

3. Arsitektur ISA (*Instruction Set Architecture*)

ini bertujuan untuk menggabungkan berbagai elemen sistem menjadi satu kesatuan yang berfungsi secara efektif dan efisien, termasuk perencanaan elemen seperti CPU, memori, I/O (*Input/Output*), dan *bus*.

6. Arsitektur SISD (*Single Instruction, Single Data*)

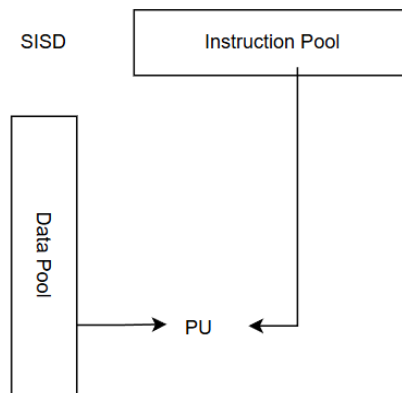


Figure 5: Diagram Arsitektur SISD

Sistem *Single Instruction Single Data* (SISD) adalah arsitektur komputasi di mana satu instruksi diproses pada satu aliran data secara berurutan dalam sebuah mesin uniprosesor. CPU hanya menangani satu aliran instruksi dan satu aliran data pada satu waktu, menjadikannya ideal untuk komputasi sekuensial dalam sistem yang sederhana. Dalam sistem ini, instruksi dan data disimpan di memori utama, tetapi kecepatan pemrosesannya dibatasi oleh kecepatan transfer data internal. Model ini kurang cocok untuk aplikasi yang membutuhkan pemrosesan paralel atau kinerja komputasi tinggi.

- Keuntungan SISD adalah kesederhanaannya dan mudah diimplementasikan, cocok untuk komputasi sekuensial
- kekurangannya adalah tidak mendukung pemrosesan paralel, sehingga kinerjanya terbatas untuk tugas komputasi intensif atau modern yang membutuhkan pemrosesan simultan.

7. Arsitektur SIMD (*Single Instruction, Multiple Data*)

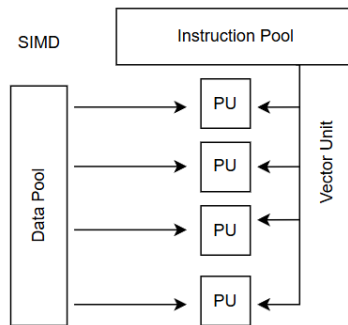


Figure 6: Diagram Arsitektur SIMD

Menurut Kahfi Alfarisi Firdaus (2015), kelas komputer paralel dalam taksonomi Flynn. Ini menggambarkan komputer dengan beberapa elemen pemrosesan yang melakukan operasi yang sama pada beberapa titik data secara bersamaan. Dengan demikian, mesin tersebut memanfaatkan data tingkat paralelisme. SIMD ini terutama berlaku untuk tugas umum seperti menyesuaikan kontras dalam citra digital atau menyesuaikan volume audio digital. Paling modern CPU desain termasuk instruksi SIMD dalam rangka meningkatkan kinerja multimedia digunakan.

- Keuntungan utama sistem SIMD adalah kemampuannya untuk memproses banyak data sekaligus dengan satu instruksi. Misalnya, jika SIMD memuat delapan data sekaligus, operasi seperti add dapat diterapkan ke seluruh data dalam satu waktu, meningkatkan paralelisme dan efisiensi dibandingkan prosesor super-skalar
- kekurangannya adalah tidak semua algoritma, terutama yang berbasis aliran kontrol, dapat dioptimalkan dengan SIMD. Selain itu, kebutuhan register besar meningkatkan konsumsi daya, dan sebagian besar kompiler tidak secara otomatis menghasilkan instruksi SIMD, sehingga memerlukan tenaga ahli untuk implementasinya.

8. Arsitektur MISD (*Multiple Instruction, Single Data*)

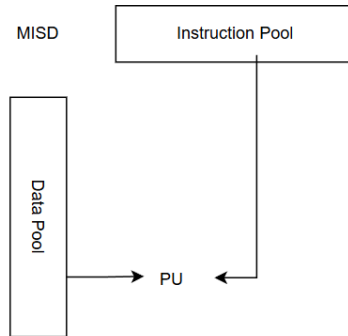


Figure 7: Diagram Arsitektur MISD

MISD (*Multiple Instruction, Single Data*) menurut Imam (2014), adalah jenis arsitektur komputasi paralel di mana beberapa prosesor menjalankan operasi berbeda pada data yang sama secara bersamaan. Arsitektur ini digunakan untuk mendeteksi dan menutupi kesalahan, seperti pada sistem toleransi kesalahan yang menggunakan replikasi tugas untuk memastikan keandalan. Meskipun termasuk dalam jenis komputasi paralel, MISD kurang umum dibandingkan dengan arsitektur SIMD dan MIMD, yang lebih cocok untuk pemrosesan data paralel pada umumnya. Contoh penerapannya jarang, karena efisiensinya yang lebih rendah dibanding arsitektur lainnya.

- keuntungan MISD dalam deteksi kesalahan dan peningkatan keamanan karena beberapa prosesor menjalankan instruksi berbeda pada data yang sama, memungkinkan validasi hasil secara efektif. Sistem ini ideal untuk aplikasi yang memerlukan keandalan tinggi, seperti toleransi kesalahan.
- kekurangannya adalah penggunaan yang sangat terbatas dalam komputasi umum, karena tidak efisien untuk tugas-tugas yang memerlukan paralelisme standar. kekurangannya adalah penggunaan yang sangat terbatas dalam komputasi umum, karena tidak efisien untuk tugas-tugas yang memerlukan paralelisme standar.

9. Arsitektur MIMD (*Multiple Instruction, Multiple Data*)

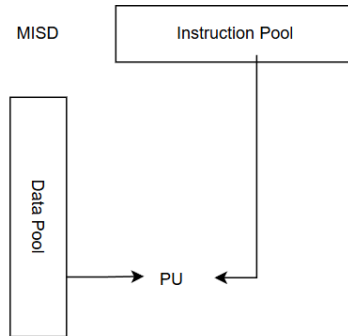


Figure 8: Diagram Arsitektur MIMD

MIMD (*Multiple Instruction Multiple Data*) adalah arsitektur di mana beberapa instruksi dapat dieksekusi secara paralel pada banyak unit data. Dengan kata lain, MIMD menerapkan berbagai instruksi pada beberapa data secara bersamaan. Arsitektur ini menggunakan mekanisme sinkronisasi lokal yang eksplisit, yang membuatnya fleksibel, tetapi juga menyebabkan perangkat lunak menjadi lebih kompleks karena programmer atau kompiler paralelisasi harus menulis secara langsung perintah untuk komunikasi data antara prosesor yang berbeda. Namun, menurut Hennessy dan Patterson (2017), kompleksitas dalam desain perangkat keras dan pemrograman menjadi tantangan utama dalam penerapannya.

- keuntungan arsitektur MIMD seperti fleksibilitas tinggi, skalabilitas, dan kecocokan untuk berbagai aplikasi karena mampu menjalankan instruksi yang berbeda secara paralel.
- kekurangan arsitektur MIMD adalah kompleksitas dalam desain perangkat keras dan pemrograman, serta efisiensi yang sangat bergantung pada pengaturan beban kerja, yang bisa membuatnya kurang optimal jika tidak dikelola dengan baik.

References

- [1] Laudon, K., dan Laudon, J. (2020). Management Information Systems: Managing the Digital Firm. Pearson.

- [2] Kahfie, M. (2024). Tugas 4 Arsitektur SIMD dan SISD. Retrieved October 3, 2024, <https://kahfie.com/tugas-4-arsitektur-simd-dan-sisd>
- [3] Iman (2014). Arsitektur MISD dan Arsitektur SIMD. Retrieved July 21, 2014, from <https://imandisini.blogspot.com/2014/07/arsitektur-misd-arsitektur-simd.html>

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

Figure 9: Ini adalah gambar contoh dari multithreading.

Seperti yang terlihat pada Gambar 9, inilah cara menambahkan gambar dengan keterangan.

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:

- Concept of virtualization
- Hypervisors and their types
- Benefits of virtualization in modern computing

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.

4.1.1 Group 1

```
class Process:
def __init__(self, pid, arrival_time, burst_time):
    self.pid = pid
    self.arrival_time = arrival_time
    self.burst_time = burst_time
    self.completion_time = 0
    self.turnaround_time = 0
    self.waiting_time = 0
```

Header 1	Header 2	Header 3
Row 1, Column 1	Row 1, Column 2	Row 1, Column 3
Row 2, Column 1	Row 2, Column 2	Row 2, Column 3

Table 1: Your table caption

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.