

# Operating System Course Report - First Half of the Semester

A class

October 10, 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
3.11.1	Fungsi User Interface . . . . .	6
3.11.2	Komponen-komponen User Interface . . . . .	7
3.12	Virtualization in Operating Systems . . . . .	11
<b>4</b>	<b>Assignments and Practical Work</b>	<b>12</b>
4.1	Assignment 1: Process Scheduling . . . . .	12
4.1.1	Group 1 . . . . .	12
4.2	Assignment 2: Deadlock Handling . . . . .	12
4.3	Assignment 3: Multithreading and Amdahl's Law . . . . .	12
4.4	Assignment 4: Simple Command-Line Interface (CLI) for User Interface Management . . . . .	13
4.5	Assignment 5: File System Access . . . . .	13
<b>5</b>	<b>Conclusion</b>	<b>13</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

Figure 1: Ini adalah gambar contoh dari multithreading.

Seperti yang terlihat pada Gambar 1, 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)

#### 3.11.1 Fungsi User Interface

##### 1. Meningkatkan Pengalaman Pengguna

*User interface* (UI) yang dirancang dengan baik berfokus pada kemudahan penggunaan dan kenyamanan pengguna dalam berinteraksi dengan sistem. UI yang intuitif, tata letak yang rapi, dan penempatan elemen yang jelas membantu pengguna untuk memahami fungsi aplikasi tanpa perlu belajar berulang kali. Hal ini menciptakan pengalaman yang lebih memuaskan.

##### 2. Menciptakan Kesan Pertama yang Positif

Kesan pertama merupakan hal yang penting ketika pengguna mulai menggunakan suatu aplikasi atau situs web. UI yang menarik secara visual, responsif, dan terorganisir akan memberikan impresi awal yang positif dan dapat mempengaruhi keputusan pengguna untuk terus menggunakan layanan tersebut.

### 3. Meningkatkan Aksesibilitas

Desain UI yang baik memperhatikan beragam kebutuhan pengguna, termasuk pengguna dengan keterbatasan fisik atau sensorik. Fitur seperti peningkatan ukuran teks, kontras warna yang sesuai, dan kemampuan untuk menggunakan perangkat bantu memastikan bahwa aplikasi atau sistem dapat diakses oleh lebih banyak orang.

### 4. Memfasilitasi Navigasi yang Efisien

UI yang baik membantu pengguna untuk menavigasi aplikasi atau situs web dengan mudah melalui struktur yang konsisten dan logis. Elemen navigasi seperti menu, tombol, dan ikon ditempatkan pada posisi yang strategis, sehingga pengguna tidak perlu membuang banyak waktu mencari fitur atau informasi.

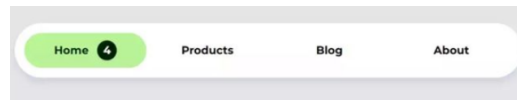
### 5. Memudahkan Interaksi Pengguna dengan Produk

UI menjadi perantara antara pengguna dan produk atau layanan. Elemen-elemen visual seperti tombol, ikon, gambar, dan teks menjadi sarana utama yang digunakan pengguna untuk berinteraksi dengan sistem. Desain UI yang baik memastikan interaksi tersebut mudah dan menyenangkan.

## 3.11.2 Komponen-komponen User Interface

Komponen-komponen utama dalam *user interface* (UI) dirancang untuk memfasilitasi interaksi pengguna dengan sistem secara efektif dan intuitif. Berikut adalah komponen-komponen umum dalam UI:

- *Navigation Bar*



Navbar adalah komponen navigasi yang biasanya terletak di bagian atas halaman dan digunakan untuk mengarahkan pengguna ke halaman-halaman penting dalam aplikasi atau situs web. Navbar membantu pengguna menavigasi konten dengan mudah dan cepat.

Steve Krug (2005) menjelaskan bahwa navigation bar adalah elemen yang dirancang untuk membantu pengguna berpindah dari satu halaman ke halaman lainnya dengan cara yang paling efisien. Navigation bar yang baik harus mudah ditemukan dan digunakan tanpa perlu berpikir keras.

- *Buttom*



Tombol (*Button*) adalah elemen interaktif yang dapat diklik oleh pengguna untuk melakukan suatu tindakan, seperti mengirimkan data, membuka halaman lain, atau menjalankan fungsi tertentu. Tombol memiliki bentuk dan gaya yang mudah dikenali dan diidentifikasi.

Ben Shneiderman (2010) menjelaskan bahwa tombol (*button*) adalah elemen dasar dalam antarmuka yang digunakan untuk memicu aksi atau perintah tertentu. Tombol harus dirancang dengan visual yang jelas dan mudah dikenali oleh pengguna, serta memberikan umpan balik yang sesuai setelah diklik.

- *Form Input*

*Form input* adalah elemen di mana pengguna dapat memasukkan informasi ke dalam sistem. *Form input* biasanya digunakan untuk mengumpulkan data dari pengguna, seperti teks, angka, pilihan dari daftar, atau file.

Ben Shneiderman (2010) menjelaskan bahwa form input adalah elemen UI yang digunakan untuk memungkinkan pengguna memasukkan data atau informasi ke dalam sistem. Input form harus dirancang agar mudah digunakan, dapat diakses, dan memberikan umpan balik yang jelas kepada pengguna setelah data dimasukkan.

- *Card*



The image shows a 'Sign Up' form card. At the top, the text 'Sign Up' is displayed in a large, blue, sans-serif font. Below this, the label 'Email' is positioned above a white rectangular input field with a thin blue border. Further down, the label 'Password' is positioned above another identical white rectangular input field. At the bottom of the form, there is a blue rectangular button with rounded corners and the text 'Sign Up' in white, sans-serif font. The entire form is centered on a white background.

*Card* adalah komponen UI yang biasanya berbentuk kotak persegi atau persegi panjang, yang digunakan untuk menampilkan informasi singkat, media, atau konten dalam bentuk yang rapi dan terstruktur. *Card* memudahkan pengguna untuk melihat beberapa elemen atau konten yang terkait dalam satu wadah.

Luke Wroblewski (2011) menjelaskan bahwa *card* adalah elemen UI yang efektif untuk menyajikan informasi dalam potongan kecil dan mudah dicerna, terutama pada perangkat seluler. Dengan menggunakan *card*, konten dapat disajikan secara modular, sehingga pengguna dapat dengan mudah memahami setiap elemen informasi tanpa merasa kewalahan.

- Modal (*Popup Dialog*)

Modal adalah kotak dialog yang muncul di atas halaman utama untuk menampilkan informasi penting atau meminta konfirmasi dari pengguna. Modal digunakan untuk interaksi yang tidak mengharuskan pengguna berpindah halaman, sehingga mengurangi gangguan pada alur pengguna.

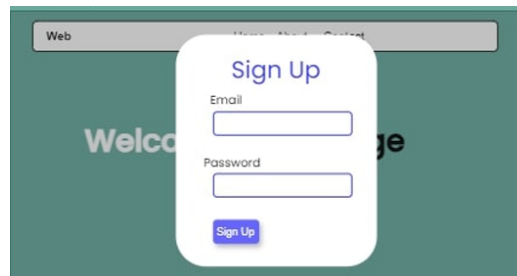
Alan Cooper (2014) menjelaskan bahwa modal adalah jendela dialog yang memblokir interaksi dengan halaman utama hingga pengguna menyelesaikan aksi pada modal tersebut. Modal sering digunakan untuk meminta pengguna mengambil keputusan penting atau



## Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna

Action

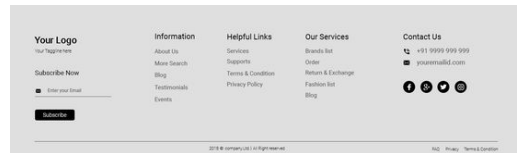


memberikan konfirmasi atas tindakan tertentu.

- *Footer*

*Footer* adalah area yang terletak di bagian bawah halaman web dan biasanya digunakan untuk menampilkan informasi umum, tautan penting, atau hak cipta. *Footer* berfungsi sebagai penutup halaman dan sering kali menyertakan elemen navigasi tambahan yang tidak terlalu esensial dibandingkan navbar.

Steve Krug (2005) menjelaskan bahwa *footer* adalah bagian penting dari halaman web yang memberikan informasi tambahan kepada pengguna, seperti tautan ke halaman kebijakan privasi, syarat dan ketentuan, atau informasi kontak. *Footer* membantu pengguna menemukan



informasi penting tanpa harus mencari di seluruh halaman.

## References

- [1] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (6th ed.). Pearson.
- [2] Tullis, T., & Albert, B. (2021). *Measuring the user experience: Collecting, analyzing, and presenting usability metrics* (3rd ed.). Morgan Kaufmann.
- [3] Krug, S. (2005). *Don't Make Me Think: A Common Sense Approach to Web Usability*. 2nd Edition. New Riders Publishing.
- [4] Shneiderman, B. (2010). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5th Edition. Addison-Wesley.
- [5] Luke Wroblewski, *Mobile First*. A Book Apart, 2011.
- [6] Alan Cooper, *About Face: The Essentials of Interaction Design*. Wiley, 2014.

### 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.