Operating System Course Report - First Half of the Semester

B class

October 1, 2024

Contents

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

3.4.1 Definition of Process

Proses adalah program yang sedang dieksekusi, yang terdiri dari instruksi yang berjalan pada CPU, serta informasi yang diperlukan untuk menjalankan program tersebut. Setiap proses memiliki siklus hidup yang melibatkan berbagai keadaan (states) dan transisi antar keadaan tersebut.

3.4.2 Process states and state transitions

Keadaan proses merujuk pada status atau fase yang dilalui oleh suatu proses selama siklus hidupnya. Setiap proses dapat berada dalam salah satu dari beberapa keadaan yang berbeda, dan transisi antar keadaan ini diatur oleh sistem operasi untuk memastikan manajemen proses yang efisien. Berikut adalah beberapa keadaan utama yang biasanya diidentifikasi dalam sistem operasi:

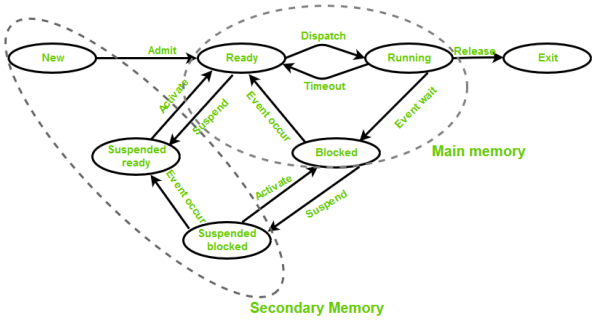

Figure 1: gambar keadaan dalam proses

- **New (Baru)**: Proses baru saja dibuat. Pada keadaan ini, proses sedang dalam tahap inisialisasi, di mana sistem operasi mengalokasikan sumber daya yang diperlukan seperti memori dan ID proses.
- **Ready (Siap)**: Proses telah siap untuk dieksekusi tetapi belum mendapatkan akses ke CPU. Proses ini menunggu dalam antrean ready hingga sistem operasi memilihnya untuk dieksekusi. Proses dalam keadaan ini dapat dipindahkan kembali ke antrean jika CPU sedang sibuk.
- **Running (Berjalan)**: Proses sedang dieksekusi oleh CPU. Pada keadaan ini, instruksi proses sedang dijalankan. CPU memberikan waktu eksekusi untuk proses dalam keadaan ini.
- **Waiting (Menunggu)**: Proses tidak dapat melanjutkan eksekusi hingga suatu event tertentu terjadi seperti menyelesaikan operasi I/O. Ketika menunggu, proses tidak dapat menggunakan CPU dan akan pindah ke antrean waiting.

- **Terminated (Selesai)**: Proses telah selesai menjalankan semua instruksinya. Semua sumber daya yang dialokasikan untuk proses tersebut dibebaskan, dan sistem operasi memperbarui status proses menjadi terminated.

Transisi antara keadaan proses ini adalah bagian penting dari manajemen proses. Berikut adalah beberapa transisi umum:

- **New Ready**: Ketika proses berhasil dibuat dan siap untuk dijadwalkan.
- **Ready Running**: Ketika CPU memilih proses dari antrean ready untuk dieksekusi.
- **Running Waiting**: Ketika proses membutuhkan I/O atau menunggu event tertentu (misalnya menyelesaikan pembacaan dari disk).
- **Running Ready**: Ketika proses di-preempt karena waktu eksekusi habis atau ada proses dengan prioritas lebih tinggi yang memerlukan CPU.
- **Waiting Ready**: Ketika event yang ditunggu oleh proses selesai, proses tersebut kembali ke antrean ready.
- **Running Terminated**: Ketika proses selesai menjalankan semua instruksinya.

3.4.3 Process Control Block (PCB)

3.4.4 Context Switching

3.4.5 Process Management By Operating Systems

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

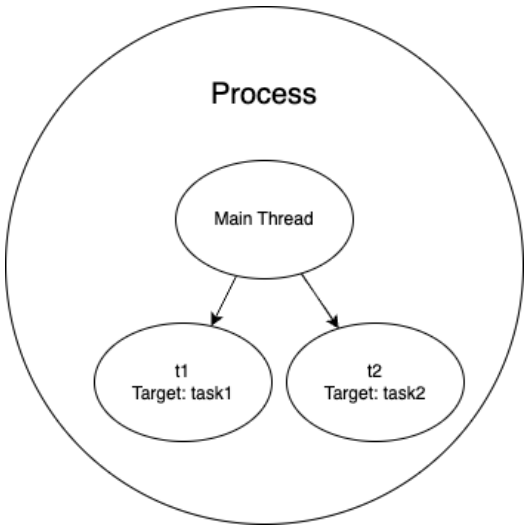

Figure 2: Ini adalah gambar contoh dari multithreading.

Seperti yang terlihat pada Gambar 2, 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.

References

[1] GeeksforGeeks. (n.d.). *Process states in operating system*. Diakses pada 30 September 2024, dari `https://www.geeksforgeeks.org/process-states-in-operating-system/`

[2] Tutorialspoint. (n.d.). *Operating system - Process control block*. Diakses pada 30 September 2024, dari `https://www.tutorialspoint.com/operating_system/os_process_control_block.htm`

[3] Wikipedia contributors. (n.d.). Context switch. Dalam *Wikipedia, The Free Encyclopedia*. Diakses pada 30 September 2024, dari `https://en.wikipedia.org/wiki/Context_switch`