Operating System Course Report - First Half of the Semester

B class

September 30, 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

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

- Process states and state transitions

- Process control block (PCB)

  Process Control Block (PCB) adalah struktur data yang paling penting dalam sebuah sistem operasi. PCB memegang peranan penting dalam mengatur proses, memungkinkan sistem operasi melanjutkan eksekusi dari titik terakhir saat proses kembali dijalankan setelah context switching. PCB digunakan oleh sistem operasi untuk menyimpan semua informasi yang diperlukan untuk mengelola proses, dan setiap proses yang berjalan di sistem operasi memiliki PCB yang terkait dengannya. PCB berisi berbagai informasi tentang proses tersebut, termasuk:

  - Process ID (PID): Merupakan nomor unik yang digunakan untuk mengidentifikasi setiap proses di dalam sistem.
  - State of Process: Menyimpan status atau kondisi proses saat ini, seperti ‘running‘, ‘waiting‘, atau ‘terminated‘.
  - Program Counter (PC): Menyimpan alamat dari instruksi berikutnya yang akan dieksekusi oleh CPU untuk proses ini.
  - CPU Registers: Menyimpan konten register CPU ketika terjadi context switch, sehingga proses dapat dilanjutkan dari titik terakhirnya saat beralih kembali.
  - Memory Management Information: Menyimpan informasi tentang alokasi memori untuk proses ini, termasuk informasi tentang batas memori, tabel halaman (page table), dan segmen memori.
  - I/O Status Information: Menyimpan informasi tentang perangkat input/output yang digunakan oleh proses, seperti file yang terbuka, dan perangkat yang dialokasikan.
  - Accounting Information: Menyimpan informasi tentang jumlah waktu CPU yang telah digunakan oleh proses, prioritas proses, dan informasi penggunaan sumber daya lainnya.

  PCB memungkinkan sistem operasi untuk melacak status setiap proses dan memastikan bahwa CPU dapat melanjutkan proses dengan benar setelah context switch terjadi. Modifikasi PCB dilakukan oleh modul-modul dalam sistem operasi seperti penjadwalan, alokasi sumber daya, dan penanganan interrupt. PCB juga berperan dalam pengawasan dan analisis kinerja sistem.

- 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:

- 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.

4.2 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.