Operating System Course Report - First Half of the Semester

A class

October 9, 2024

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 introduction
- Conditions for deadlock
- Deadlock detection and recovery
- Deadlock prevention techniques

3.10.1 Deadlock in Everyday Life

Deadlock is not just a concept in computer systems; it can also occur in daily situations where multiple parties are waiting for each other to release resources, leading to a standstill. Here are a few examples of deadlock in real life:

- **Traffic Jam at an Intersection:** Imagine a four-way intersection with cars coming from all directions. If each car enters the intersection and blocks the others, no one can move forward. This is a classic example of deadlock, where each car is waiting for the others to move.
- **Diners Waiting for Cutlery:** In a restaurant, if several diners are sharing a set of utensils, deadlock can occur if each diner holds onto one item (e.g., one diner holds a knife and waits for a fork, while the other holds a fork and waits for the knife). Neither can proceed without the other.
- **Locked Doors and Keys:** Two people are trying to get into separate rooms, but each room's key is in the other room. If person A is waiting for person B to open their room to get the key, and person B is waiting for person A to do the same, they will both be stuck indefinitely.
- **Shared Resources at the Workplace:** In a collaborative work environment, if two teams are waiting for shared resources like a printer or project approval from another team, neither can proceed until the other has finished, leading to a deadlock in the workflow.

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: Akses File Sistem

Dalam assignment ini, siswa diminta untuk melakukan akses file sistem menggunakan Python. Soal ini mencakup pembuatan file, membaca isi file, dan menampilkan daftar file dalam direktori. Berikut adalah soal yang harus dikerjakan:

Soal: Buatlah sebuah program Python yang melakukan hal berikut: 1. Membuat sebuah file baru dengan nama ‘contoh.txt‘ dan menuliskan beberapa kalimat ke dalamnya. 2. Membaca dan menampilkan isi dari file ‘contoh.txt‘. 3. Menampilkan semua file yang ada di direktori saat ini.

Kode Python:

```
import os

# 1. Membuat file dan menuliskan isi
with open('contoh.txt', 'w') as file:
    file.write('Ini adalah contoh file.\n')
    file.write('File ini digunakan untuk demonstrasi akses file sistem.\n')

# 2. Membaca dan menampilkan isi file
with open('contoh.txt', 'r') as file:
    isi_file = file.read()
    print('Isi dari contoh.txt:')
    print(isi_file)

# 3. Menampilkan semua file di direktori saat ini
print('\nDaftar file di direktori saat ini:')
for nama_file in os.listdir('.'):
    print(nama_file)
```

Jawaban: Ketika kode di atas dijalankan, output yang diharapkan adalah sebagai berikut:

```
Isi dari contoh.txt:
Ini adalah contoh file.
File ini digunakan untuk demonstrasi akses file sistem.

Daftar file di direktori saat ini:
contoh.txt
```

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.