Operating System Course Report - First Half of the Semester

A class

October 3, 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.

3.3.1 *Process Control Block*

Process Control Block (PCB) adalah struktur data penting yang digunakan oleh sistem operasi untuk melacak dan mengelola proses yang sedang berjalan. Setiap kali sebuah proses dibuat, sistem operasi menciptakan PCB yang berisi informasi lengkap tentang proses tersebut. PCB ini menyimpan berbagai elemen yang memungkinkan sistem operasi mengatur proses secara efisien, termasuk status, memori, hingga informasi tentang perangkat

input/output yang digunakan. PCB bisa dianggap sebagai identitas atau profil dari sebuah proses. Setiap proses di dalam sistem operasi memiliki satu PCB, yang berisi informasi yang digunakan saat proses tersebut dijalankan, dihentikan, atau ditukar dengan proses lain (*context switching*).
Elemen-elemen PCB:

1. *Identifier*
   Nomor unik yang disebut PID untuk membedakan setiap proses.

2. *Priority*
   Nilai yang menentukan urutan proses mana yang akan dijalankan terlebih dahulu oleh CPU.

3. *Memory Pointers*
   Menunjuk ke lokasi kode, data, dan stack proses di memori.

4. *I/O Status Information*
   Nomor unik yang disebut PID untuk membedakan setiap proses.

5. *State*
   Status proses saat ini, seperti *new, ready, running, waiting,* atau *terminated.*

6. *Program Counter*
   Menyimpan alamat instruksi berikutnya yang akan dijalankan dalam proses.

7. *Context Data*
   Isi register CPU dan informasi lain yang disimpan saat proses dihentikan sementara untuk memungkinkan *context switching.*

8. *Accounting Information*
   Data penggunaan sumber daya oleh proses, termasuk waktu CPU, waktu mulai, dan statistik penting lainnya.

3.3.2 *Process Control*

Kontrol Proses/*Process Control* adalah sekumpulan metode dan mekanisme yang digunakan oleh sistem operasi untuk mengelola siklus hidup suatu proses, mulai dari penciptaan, penjadwalan, hingga penghentian proses. Proses ini sangat penting untuk menjaga sistem tetap berjalan efisien dan memastikan setiap proses mendapatkan sumber daya CPU yang dibutuhkan tanpa mengganggu proses lain.

3.4 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.

3.5 Process Creation and Termination

Details how processes are created and terminated by the operating system, including:

- Process spawning
- Process termination conditions

3.6 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

Seperti yang terlihat pada Gambar 1, inilah cara menambahkan gambar dengan keterangan.

/Users/khawaritzmi/Unhas/os_report_mid2024/a_class/asset/example.j

Figure 1: Ini adalah gambar contoh dari multithreading.

3.7 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

3.8 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 performance

3.9 Deadlock Introduction and Prevention

Explores the concept of deadlocks and methods for preventing them:

- Deadlock conditions
- Deadlock prevention techniques

3.10 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

3.11 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 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.