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

3.4.2 Process states and state transitions

3.4.3 Process Control Block (PCB)

3.4.4 Context Switching

3.4.5 Process Management By Operating Systems

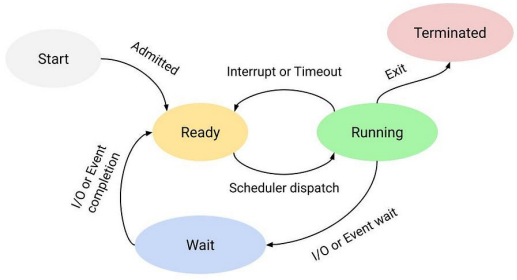

Figure 1: Ini adalah gambar contoh dari multithreading.

Manajemen proses oleh sistem operasi adalah salah satu tugas utama dari sistem operasi yang melibatkan pengelolaan dan pengendalian berbagai proses yang berjalan di komputer. Sistem operasi bertanggung jawab untuk memastikan bahwa proses-proses ini dieksekusi secara efisien dan tidak saling mengganggu. Berikut adalah aspek utama dari manajemen proses oleh sistem operasi:

- **Pembuatan dan Penghancuran Proses**: Sistem operasi membuat proses baru ketika suatu program dieksekusi atau ketika suatu proses meminta sistem untuk membuat subproses (misalnya dengan menggunakan perintah *fork()* pada sistem UNIX). Saat proses dibuat, *Process Control Block (PCB)* dialokasikan, dan informasi proses disimpan, termasuk ID proses, status proses, dan informasi lainnya. Sistem operasi juga bertanggung jawab untuk menghentikan dan menghapus proses yang telah selesai. Setelah proses selesai, sumber daya yang digunakan oleh proses dibebaskan agar bisa digunakan oleh proses lain.

- **Penjadwalan Proses (Process Scheduling)**: Sistem operasi harus menentukan proses mana yang akan dijalankan terlebih dahulu, terutama dalam lingkungan multitasking di mana banyak proses berjalan secara bersamaan. Penjadwal Proses *(Scheduler)* menentukan urutan eksekusi proses berdasarkan prioritas dan status proses. Beberapa jenis penjadwalan proses yang umum meliputi:
  - *First Come First Serve* (FCFS): Proses yang datang terlebih dahulu akan dijalankan terlebih dahulu.
  - *Round Robin*: Setiap proses diberikan waktu eksekusi yang terbatas, kemudian berpindah ke proses lain.
  - *Shortest Job Next* (SJN): Proses dengan waktu eksekusi tersingkat dijalankan lebih dulu.
  - *Priority Scheduling*: Proses dengan prioritas lebih tinggi mendapatkan akses CPU lebih cepat.
- **Context Switching**: *Context switching* adalah proses di mana CPU berpindah dari satu proses ke proses lain. Sistem operasi harus menyimpan konteks proses yang sedang berjalan (informasi register, *program counter*, dll.) dan memuat konteks proses baru. Ini memungkinkan multitasking, di mana beberapa proses berjalan secara bersamaan (secara sekuensial tetapi sangat cepat sehingga tampak bersamaan).
- **Komunikasi Antar-Proses (*Inter-Process Communication* - IPC)**: Sistem operasi menyediakan mekanisme bagi proses untuk berkomunikasi satu sama lain, terutama jika mereka perlu berbagi data atau bekerja sama. Beberapa metode umum IPC termasuk:
  - *Shared Memory*: Dua atau lebih proses berbagi bagian dari memori.
  - *Message Passing*: Proses saling mengirim pesan untuk bertukar informasi.
- **Sinkronisasi Proses**: Sinkronisasi diperlukan untuk menghindari *race conditions*, di mana dua proses mencoba mengakses atau memodifikasi sumber daya yang sama secara bersamaan. Mekanisme sinkronisasi yang disediakan oleh sistem operasi termasuk *semaphores, mutexes,* dan *monitors.*

- **Manajemen Deadlock**: Deadlock terjadi ketika dua atau lebih proses saling menunggu untuk sumber daya yang tidak bisa mereka peroleh, yang mengakibatkan proses tersebut tidak bisa melanjutkan eksekusi. Sistem operasi mengelola deadlock melalui berbagai strategi, seperti deteksi *deadlock,* pencegahan *deadlock*, atau penghindaran *deadlock* (seperti algoritma Banker's).

- **Manajemen Sumber Daya Proses**: Sistem operasi bertanggung jawab untuk mengalokasikan dan mengelola sumber daya yang dibutuhkan oleh proses, seperti CPU, memori, perangkat I/O, dan file. Ini termasuk memantau penggunaan sumber daya untuk mencegah konflik dan memastikan distribusi yang adil antara proses.

- **Keadaan Proses (Process States)**: Setiap proses dalam sistem memiliki status tertentu (misalnya *New, Ready, Running, Waiting, Terminated*) yang mencerminkan posisinya dalam siklus hidupnya. Sistem operasi bertanggung jawab untuk memindahkan proses antar status ini dan melakukan transisi secara efisien, seperti memindahkan proses dari *ready* ke *running* atau dari *running* ke *waiting.*

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

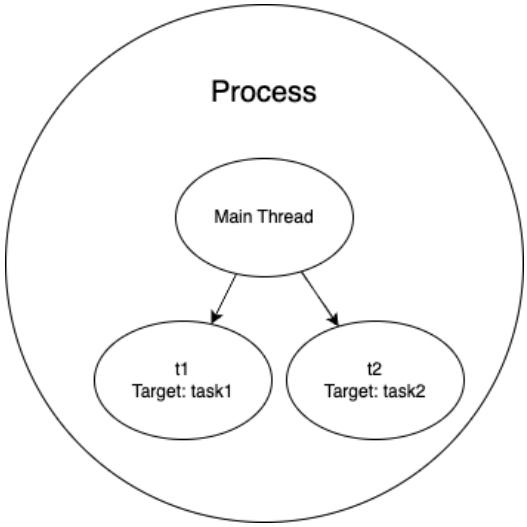

Figure 2: Ini adalah gambar contoh dari multithreading.

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`