Operating System Course Report - First Half of the Semester

B class

October 8, 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

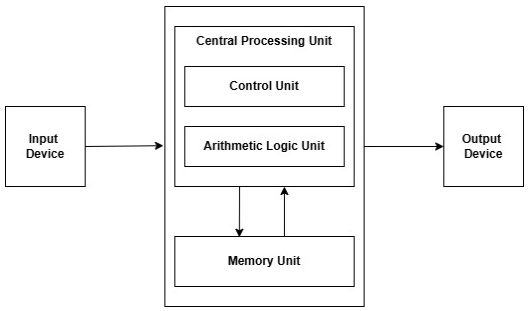

Figure 1: Arsitektur Von Neumann

3.3.1 Arsitektur Komputer

Arsitektur komputer merujuk pada struktur dasar dari sistem komputer yang melibatkan perangkat keras atau *hardware* dan perangkat lunak atau *software* yang bekerja sama untuk menjalankan instruksi dan memproses data. Arsitektur komputer lebih menekankan fokus pada konsep dan prinsip dasar yang mengatur operasi komputer secara umum. Berikut ini merupakan komponen utama dari arsitektur komputer.

Control Unit

Control Unit merupakan pusat dari otak komputer. *Control Unit* bertugas untuk mengawasi berbagai siklus instruksi sehingga dapat melakukan *control* pada sinyal-sinyal yang relevan pada saat yang tepat, supaya operasi *micro* dapat dikerjakan pada CPU dan unit eksternal yang terhubung pada CPU seperti *memory* dan *input output devices* atau *contoller*. *Control unit* dirancang pada suatu data spesifik yang berada pada organisasi *datapath* ALU (unit logika aritmatika), register serta sebagainya yang berada pada CPU.

Arithmetic Logic Unit

Semua komponen CPU yang lainnya dan seluruh komponen penyusun komputer membawa data ke ALU untuk diproses dan mengambil kembali hasil setelah diproses ALU. ALU membentuk fungsi-fungsi pengolahan data pada komputer, di antaranya operasi aritmatika dan logika terhadap data. Adapun tugas ALU yaitu melakukan perintah yang berhubungan dengan perhitungan aritmatika serta mengambil keputusan dari suatu operasi sesuai dengan instruksi dari program yang disebut operasi logika. Karena itu, ALU berguna untuk melakukan proses data dalam bentuk angka dan logika.

Memori

Memori adalah tempat penyimpanan informasi yang dapat diakses oleh CPU. Sistem komputer modern menggunakan beberapa jenis memori, masing-masing dengan tujuan dan karakteristiknya sendiri.

- RAM (*Random Access Memory*) adalah jenis memori yang digunakan oleh komputer untuk menyimpan data sementara yang dibutuhkan CPU saat menjalankan program atau tugas tertentu. Keuntungan

RAM adalah kecepatan akses tinggi, tetapi data yang disimpan di dalamnya hilang ketika daya listrik dimatikan

- ROM (*Read Only Memory*) adalah jenis memori yang menyimpan data yang tidak berubah, bahkan ketika daya listrik dimatikan.

Input/Output

Input/Output (I/O) merupakan bagian dari sebuah sistem mikroprosesor yang berfungsi untuk dapat berinteraksi dengan berbagai *peripheral* perangkat luar. Unit input merupakan unit luar yang berfungsi sebagai unit yang dapat memberikan sebuah instruksi atau memasukan sebuah data dari perangkat eksternal ke dalam mikroprosesor, sebagai Contoh dari unit input adalah sebuah data yang bersumber dari perangkat input berupa *keyboard* atau *mouse*. Sedangkan unit output adalah sebuah unit yang berfungsi untuk dapat menampilkan sebuah data yang dikirimkan dari mikroprosesor, sebagai Contoh dari unit output adalah sebuah hasil pengolahan data yang dapat diterjemahkan oleh pengguna baik berupa gambar, suara, visual atau hasil cetak yaitu berupa monitor, *printer* atau *sound audio*.

References

[1] Cahyaningrum, Y. Y. (2023). *ARISTEKTUR DAN ORGANISASI KOMPUTER*. Yogyakarta: PT Penamuda Media.

[2] Amrival, V. A. (2013). *ARSITEKTUR KOMPUTER: Teori dan Perkembangannya*. Jakarta: Halaman Moeka Publishing.

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 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.

Group 3

Soal:

Buatlah program CLI sederhana yang memungkinkan pengguna untuk mengelola tampilan antarmuka pengguna (UI). Program ini harus memberikan opsi untuk menampilkan menu, menambahkan item ke UI, dan menghapus item dari UI. Program harus memiliki opsi sebagai berikut:

- Tampilkan item dalam UI
- Tambah item ke UI

- Hapus item dari UI
- Keluar

Implementasikan program dalam Python menggunakan konsep CLI.

```
def display_menu():
print("=== UI Management ===")
print("1. Tampilkan item dalam UI")
print("2. Tambah item ke UI")
print("3. Hapus item dari UI")
print("4. Keluar")

def display_ui(ui_items):
if ui_items:
print("\nItem UI saat ini:")
for idx, item in enumerate(ui_items, start=1)
    :
print(f"{idx}. {item}")
else:
print("\nUI masih kosong.")
print()

def add_ui_item(ui_items):
item = input("Masukkan nama item yang ingin
    ditambahkan ke UI: ")
ui_items.append(item)
print(f"'{item}' berhasil ditambahkan ke UI.\
    n")

def remove_ui_item(ui_items):
if not ui_items:
print("\nUI kosong, tidak ada item yang bisa
    dihapus.\n")
return
display_ui(ui_items)
try:
idx = int(input("Masukkan nomor item yang
    ingin dihapus: "))
if 1 <= idx <= len(ui_items):
removed_item = ui_items.pop(idx - 1)
print(f"'{removed_item}' berhasil dihapus
    dari UI.\n")
else:
print("Nomor item tidak valid.\n")
except ValueError:
```

```
                print("Input tidak valid, masukkan angka.\n")

                def cli_ui_management():
                ui_items = []
                while True:
                display_menu()
                try:
                choice = int(input("Pilih opsi: "))
                if choice == 1:
                display_ui(ui_items)
                elif choice == 2:
                add_ui_item(ui_items)
                elif choice == 3:
                remove_ui_item(ui_items)
                elif choice == 4:
                print("Keluar dari program.")
                break
                else:
                print("Pilihan tidak valid, coba lagi.\n")
                except ValueError:
                print("Input tidak valid, masukkan angka.\n")


                if __name__ == "__main__":
                cli_ui_management()
```

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

Group 3

Soal:

Buatlah program Python yang melakukan operasi dasar pada sistem file. Program ini harus:

- Membaca isi dari file tertentu
- Menambahkan teks ke dalam file
- Menampilkan daftar file dalam direktori

Implementasikan program dalam Python menggunakan konsep CLI.

```
import os

def list_files_in_directory(directory):
try:
files = os.listdir(directory)
print(f"\nDaftar file dalam direktori '{
    directory}':")
for file in files:
print(file)
print()
except FileNotFoundError:
print(f"Direktori '{directory}' tidak
    ditemukan.\n")

def read_file(file_path):
try:
with open(file_path, 'r') as file:
content = file.read()
print(f"\nIsi file '{file_path}':")
print(content)
print()
except FileNotFoundError:
print(f"File '{file_path}' tidak ditemukan.\n
    ")
except IOError:
print(f"Tidak dapat membaca file '{file_path
    }'.\n")

def write_to_file(file_path, text):
try:
with open(file_path, 'a') as file:
file.write(text + "\n")
print(f"Teks berhasil ditambahkan ke file '{
    file_path}'.\n")
except IOError:
print(f"Tidak dapat menulis ke file '{
    file_path}'.\n")
```

```
def file_system_access():
while True:
print("=== File System Access ===")
print("1. Tampilkan daftar file dalam
   direktori")
print("2. Baca file")
print("3. Tambah teks ke file")
print("4. Keluar")

try:
choice = int(input("Pilih opsi: "))
if choice == 1:
directory = input("Masukkan direktori: ")
list_files_in_directory(directory)
elif choice == 2:
file_path = input("Masukkan path file: ")
read_file(file_path)
elif choice == 3:
file_path = input("Masukkan path file: ")
text = input("Masukkan teks yang ingin
   ditambahkan: ")
write_to_file(file_path, text)
elif choice == 4:
print("Keluar dari program.")
break
else:
print("Pilihan tidak valid, coba lagi.\n")
except ValueError:
print("Input tidak valid, masukkan angka.\n")

if __name__ == "__main__":
file_system_access()
```

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.