The University of Faisalabad

NAME: Ayesha Imran

2023-BS-AI-057

SUBMITTED TO: MISS IRSHA QURESHI

CLASS TASK

Operating System

1. Introduction

An **Operating System (OS)** is software that manages computer hardware and software resources and provides services for computer programs. It acts as an interface between the user and the computer hardware. The OS is essential for executing programs efficiently, managing files, allocating memory, and scheduling processes.

In this assignment, we will cover the basics of operating systems, process management, memory management, file systems, and scheduling algorithms to understand how an OS functions.

2. Operating System Basics

2.1 Types of Operating Systems

1. **Batch Operating System** – Executes a batch of jobs without user interaction.
2. **Time-Sharing Operating System** – Multiple users share the CPU simultaneously (e.g., UNIX).
3. **Distributed Operating System** – Manages multiple computers connected via a network.
4. **Real-Time Operating System (RTOS)** – Provides immediate response to inputs, used in embedded systems.
5. **Multi-User and Multi-Tasking OS** – Supports multiple users and tasks at the same time (e.g., Windows, Linux).
6. **Mobile Operating System** – Designed for smartphones and tablets (e.g., Android, iOS).

2.2 Functions of an Operating System

1. **Process Management** – Controls execution of processes and scheduling.
2. **Memory Management** – Allocates and deallocates memory for processes.
3. **File Management** – Organizes, stores, and retrieves data efficiently.
4. **Device Management** – Manages input/output devices and drivers.
5. **Security and Protection** – Provides authentication, access control, and encryption.

3. Process Management

3.1 What is a Process?

A **process** is an executing program. It includes program code, data, registers, and other necessary information.

3.2 Process States

1. **New** – Process is created.
2. **Ready** – Process is waiting for CPU allocation.
3. **Running** – Process is being executed by the CPU.
4. **Waiting** – Process is waiting for an event (e.g., I/O operation).
5. **Terminated** – Process execution is completed.

3.3 Process Control Block (PCB)

The **Process Control Block (PCB)** contains essential information about a process, such as:

- Process ID
- Program Counter (PC)
- CPU registers
- Memory management information
- I/O status information

3.4 Inter-Process Communication (IPC)

Processes need to communicate with each other for better coordination. IPC mechanisms include:

- **Message Passing** – Processes send and receive messages.
- **Shared Memory** – Processes access shared memory space for communication.

4. Memory Management

4.1 Overview

Memory management ensures efficient use of RAM and virtual memory. It keeps track of memory allocation and deallocation for processes.

4.2 Virtual Memory and Paging

- **Virtual Memory** – Allows execution of processes larger than physical RAM by using a portion of disk storage.
- **Paging** – Divides memory into fixed-size blocks (**pages**) and loads them into frames in RAM.

4.3 Segmentation vs. Paging

Feature	Paging	Segmentation
Division	Fixed-size pages	Variable-size segments
Fragmentation	Internal fragmentation possible	External fragmentation possible
Addressing	Page number + Offset	Segment number + Offset

4.4 Memory Allocation Techniques

1. **Contiguous Allocation** – Allocates a continuous block of memory.
2. **Non-Contiguous Allocation** – Divides memory into different locations.
3. **Paging** – Divides memory into equal-sized pages.
4. **Segmentation** – Divides memory into logical units.

5. File Systems

5.1 What is a File System?

A file system is a method used by an OS to organize, store, and retrieve data on a disk.

5.2 Types of File Systems

1. **FAT (File Allocation Table)** – Used in older Windows systems and USB drives.
2. **NTFS (New Technology File System)** – Used in modern Windows systems, supports large files and security features.
3. **ext3/ext4 (Extended File System)** – Used in Linux-based systems.
4. **HFS+ and APFS** – Used in macOS.

5.3 File Operations

- **Create** – Make a new file.
- **Read** – Access file content.
- **Write** – Modify file content.
- **Delete** – Remove file from the system.

5.4 Directory Structures

1. **Single-Level Directory** – All files stored in one directory.
2. **Two-Level Directory** – Separate directories for each user.
3. **Tree-Structured Directory** – Hierarchical organization of files and directories.

6. Scheduling Algorithms

6.1 What is CPU Scheduling?

CPU scheduling determines which process runs when multiple processes are waiting for execution.

6.2 Types of Scheduling Algorithms

1. First Come First Serve (FCFS)

- Processes are executed in the order they arrive.
- Simple but causes the **convoy effect** (long processes delay short ones).

2. Shortest Job Next (SJN) / Shortest Job First (SJF)

- Process with the shortest burst time is executed first.
- Can lead to **starvation** (longer processes may never execute).

3. Round Robin (RR)

- Each process gets a fixed **time quantum** for execution.
- Prevents starvation and ensures fairness.

4. Priority Scheduling

- Each process has a priority level. Higher priority processes execute first.
- **Problem:** Low-priority processes may starve.
- **Solution:** Aging (gradually increasing priority over time).

5. Multilevel Queue Scheduling

- Processes are divided into different priority queues.
- Each queue has its own scheduling algorithm.

6.3 Comparison of Scheduling Algorithms

Algorithm	**Efficiency**	**Fairness**	**Starvation**
FCFS	Low	Fair	No
SJN	High	Unfair	Yes
RR	Medium	Fair	No
Priority	Medium	Unfair	Yes
Multilevel Queue	High	Depends on queue policy	Yes

7. Conclusion

Operating systems play a crucial role in managing computer resources efficiently. We explored key concepts like **process management, memory management, file systems, and CPU scheduling**. Understanding these fundamentals is essential for software development and system administration.

8. References

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts (10th ed.)*.
2. Stallings, W. (2018). *Operating Systems: Internals and Design Principles (9th ed.)*.
3. Tanenbaum, A. S., & Bos, H. (2014). *Modern Operating Systems (4th ed.)*.