Activity1$_S is_O p$

Kevin Pérez Román

August 2024

1 Introduction

In this document we realized the activity 1 from the firts assignament by the courses Oprative systems.

2 Enumerate the milestones in the evolution of computer systems.

- **First Generation (1940s-1950s):** The computers in this era were characterized by vacuum tubes and plugboards, with no operating system. All tasks were programmed manually, and they were executed one at a time without any automation or task management.
- **Second Generation (1950s-1960s):** In this era, computers upgraded to transistors and batch systems, where jobs were grouped together to be processed in sequence. Simple operating systems began to emerge, introducing concepts that are still relevant today, such as executing tasks in a specific order and with defined priorities.
- **Third Generation (1960s-1970s):** Integrated circuits appeared, enabling the development of multiprogramming and time-sharing systems, which allowed multiple users to interact with the computer simultaneously. These advances significantly improved efficiency and resource access.
- **Fourth Generation (1970s-present):** The advent of microprocessors led to the creation of personal computers and graphical user interfaces (GUIs). This generation also saw the rise of distributed systems and networks, transforming how computing resources were used and shared.
- **Fifth Generation (Present and Future):** Characterized by parallel processing, real-time systems, cloud computing, virtualization, and advancements in mobile and embedded operating systems. This generation also explores emerging technologies such as artificial intelligence and machine learning, which are redefining the capabilities and functions of operating systems.

3 What are the four components of a computer system? Describe each one.

- **Hardware:** The physical components of a computer system, including the CPU, memory, I/O devices, and storage devices. It provides the basic computing resources for the system.
- **Operating System:** The software layer that manages the hardware resources, provides a user interface, and serves as an intermediary between applications and hardware. It ensures efficient and fair resource allocation.
- **Application Programs:** Software that performs specific tasks for users, such as word processors, web browsers, and database systems. These programs rely on the OS to access hardware resources.
- **Users:** Individuals or entities that interact with the computer system. Users can be humans, other computers, or automated processes.

4 What is the difference between a monolithic kernel and a microkernel?

- **Monolithic Kernel:** In a monolithic kernel, all OS services (such as file system management, process management, and device drivers) are integrated into a single large block of code that runs in kernel mode. This approach can be efficient but can lead to stability and security issues since a bug in one component can crash the entire system.
- **Microkernel:** A microkernel has a minimalistic design, where only essential services (like basic inter-process communication and scheduling) run in kernel mode. Other services, like file systems and device drivers, run in user mode as separate processes. This approach enhances modularity, security, and stability but can introduce performance overhead due to the need for communication between user-mode services.

5 Define an Operating System from two different perspectives.

- **User Perspective:** From the user's point of view, an operating system is an interface that provides a convenient way to interact with the computer hardware. It abstracts the complexity of hardware management and provides services such as file management, program execution, and I/O operations.
- **System Perspective:** From the system's point of view, an operating system is a resource manager that allocates and controls the computer's

resources, such as CPU time, memory, and I/O devices, ensuring that different applications and users can operate efficiently and securely.

6 What is the purpose of system calls?

System calls are the interface through which user-level applications request services from the operating system. They provide a controlled entry point into the kernel, allowing programs to perform operations such as process control, file manipulation, and communication with hardware, without requiring direct access to the hardware.

7 What is a multiprogrammed operating system?

A multiprogrammed operating system is one that can execute multiple programs simultaneously by managing the CPU's time among them. This is achieved by keeping several jobs in memory at once and switching between them, allowing the CPU to be utilized more efficiently by reducing idle time.

8 What is a process?

A process is an instance of a program in execution. It consists of the program code (text), data, stack, and a Process Control Block (PCB) that contains information about the process's current state and resources. A process is the fundamental unit of work in an operating system.

9 What are the states of a process?

- **New:** The process is being created.
- **Running:** Instructions are being executed by the CPU.
- **Waiting (Blocked):** The process is waiting for some event (such as I/O completion) to occur.
- **Ready:** The process is ready to run but is waiting for CPU time.
- **Terminated:** The process has finished execution and is being removed from memory.

10 What information is stored in the Process Control Block (PCB) associated with a process?

- **Process State:** The current state of the process (e.g., running, waiting).
- **Program Counter:** The address of the next instruction to be executed.
- **CPU Registers:** The contents of all processor registers.
- **Memory Management Information:** Information such as page tables, segment tables, and base and limit registers.
- **Accounting Information:** CPU usage, process priority, and other accounting data.
- **I/O Status Information:** List of I/O devices allocated to the process, list of open files, etc.

11 What are the main activities of an operating system in relation to process management?

- **Process Creation and Deletion:** Managing the creation and termination of processes.
- **Process Scheduling:** Deciding which process should be executed by the CPU next.
- **Process Synchronization:** Ensuring that processes do not interfere with each other while sharing resources.
- **Process Communication:** Facilitating communication between processes, whether they are on the same machine or across a network.
- **Deadlock Handling:** Ensuring that processes do not get stuck in a state where they cannot proceed because each is waiting for a resource held by another.