Operating System - Week 9

Khaula Molapo

1. A distributed operating system is used to connect and manage many computers that work together like one large system. This design is often used in cloud platforms such as Google Cloud, AWS, and Microsoft Azure. Each computer, also called a node, runs part of the system and shares resources such as storage, memory, and processing power. The distributed OS helps users access data or applications without needing to know where they are stored physically. It controls communication, load balancing, and data synchronization between nodes, ensuring that all systems work smoothly together. If one machine fails, others take over automatically, which improves reliability and uptime. The operating system also helps in dividing tasks evenly, so no single machine becomes overloaded. This design is very useful for businesses and research centers that need to handle large amounts of data or heavy workloads. Overall, a distributed OS provides efficiency, flexibility, and fault tolerance in IT environments.

2.

Distributed process management is how a distributed operating system controls and coordinates multiple processes running across different machines. Each process can be on a separate computer but still communicate and work together to complete a shared goal. This helps the system use resources efficiently and balance workloads between computers. For example, if one computer is too busy, the OS can move a process to another computer that has free space. It also helps monitor, start, stop, and recover processes in case one fails. Communication between processes happens through message passing, ensuring they stay connected and share information safely. Distributed process management is important for maintaining performance and reliability in large systems such as cloud servers or data centers. It helps make the whole system work as one powerful unit, even though it is made up of many smaller parts.

3.

Setting up a distributed system using Docker helps to understand how multiple containers can work together like separate computers. In this setup, I created several Docker containers that each run a small service, such as a web server or a database. By linking them through a network, they can communicate and share data. For example, one container can process data while another stores it. This setup shows how distributed systems handle tasks through cooperation and connection. Docker makes it easy to simulate this because each container acts as its own mini operating system. When I tested it, I saw how tasks could run in parallel without interfering with each other. It also showed how updates and scaling can be done easily by adding or removing containers. Overall, Docker is a powerful tool to practice and understand distributed system concepts in a simple and practical way.

4.

The designed distributed system architecture includes multiple computers or servers connected through a fast network, each handling specific roles. For example, one server can manage user requests, another can process data, and another can store it securely. These servers communicate using a distributed operating system, which manages resources and keeps everything synchronized. The purpose of this design is to improve speed, reliability, and scalability. If one server fails, others can continue running without affecting users. This fault tolerance makes distributed systems more stable than single systems. It also allows the system to grow easily by adding more servers as demand increases. Businesses use this type of system to manage websites, databases, and cloud services that handle millions of users every day. In short, distributed system architecture helps organizations achieve high performance, better load distribution, and flexible scalability for future needs.