Organizing Computer Facility

CS202 Lecture 16b

Content of Last Lecture

1. What is Internet of Things (IoT)?
2. History of IoT?
3. Why IoT?
4. IoT Evolution
5. **Building Blocks of IoT**
6. **IoT Verticals and Application Areas**
7. **Challenges**

Content of This Lecture

1. Organizing Computer Facility
2. Centralized Computing Facility
3. Distributed Computing Facility
4. Decentralized Computing Facility
5. Cloud Computing
6. Fog Computing
7. Edge Computing
8. Introduction to Operating Systems
9. Computer Virus and its Forms

Organizing Computer Facility

Definition: Organizing a computer facility involves structuring and managing computer hardware, software, network infrastructure, and resources to ensure optimal performance and alignment with organizational objectives.

Components:

- **Hardware:** Servers, workstations, peripherals, and storage devices
- **Software:** System software, including operating systems, and application software
- **Network:** Design and maintenance of network infrastructure for connectivity
- **Security:** Protecting systems and data with firewalls, encryption, and other security protocols

Objectives:

- Maximizing resource utilization
- Ensuring data integrity and security
- Minimizing downtime and ensuring availability
- Cost-effectiveness and scalability

Centralized Computing Facility

Definition: A computing setup where all resources (hardware, software, data, and services) are housed and managed in a single central location

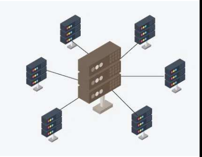
Advantages:

- **Control:** Centralized control over all resources, making it easier to enforce security policies and ensure standardization
- **Maintenance:** Simplified management and maintenance due to the concentration of resources
- **Cost Efficiency:** Reduced duplication of resources, leading to lower operational costs

Disadvantages:

- **Single Point of Failure:** If the central facility fails, all users are affected
- **Latency Issues:** Users at remote locations might experience delays

Use Cases: Suitable for organizations where strict control and security over data are paramount, such as financial institutions or government entities

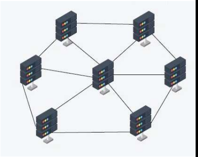


Distributed Computing Facility

Definition: Resources are distributed across multiple locations, interconnected by a network to work as a single cohesive unit

Advantages:

- **Scalability:** Easier to add new resources as needed
- **Fault Tolerance:** Failure in one part of the system doesn't bring down the entire system
- **Performance:** Tasks can be processed in parallel, improving overall performance



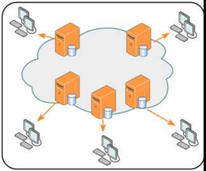
Disadvantages:

- **Complexity:** Management of a distributed environment is more complex
- **Security:** More challenging to secure compared to centralized systems

Examples: Cloud computing, and content delivery networks (CDNs)

Content Delivery Networks (CDNS)

- **Amazon CloudFront:** Helps in speeding up the delivery of static and dynamic web content, such as HTML, CSS, images, and videos, by caching content at edge locations
- **Google Cloud CDN:** Integrates with Google Cloud services and helps reduce latency by delivering content from Google's distributed servers, using Google's global infrastructure to cache and deliver content quickly
- **Microsoft Azure CDN:** Deliver high-bandwidth content by distributing it across multiple servers to reduce latency and optimize load times, especially for large media content and enterprise applications



CDNs help websites / applications deliver content to users faster and more efficiently, especially for global audiences

Decentralized Computing Facility

Definition: Each department or division in an organization manages its own computing resources independently

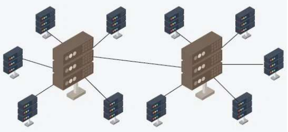
Advantages:

- **Flexibility:** Departments can choose hardware and software that best meet their needs.
- **Quick Response:** Decisions can be made locally without waiting for central approval.

Disadvantages:

- **Duplication of Resources:** Each department might purchase similar resources, leading to inefficiency.
- **Inconsistent Standards:** Different departments may implement varying standards and security protocols.

Use Cases: Ideal for large organizations with diverse operational needs, such as multinational corporations.



Cloud Computing

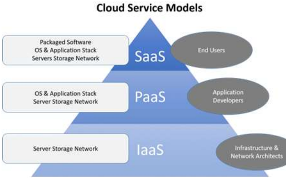
Definition: Delivery of computing services like servers, storage, databases, networking, software, and more over the internet (the cloud)

Models:

IaaS (Infrastructure as a Service): Provides virtualized computing resources like VMs (e.g., AWS EC2)

PaaS (Platform as a Service): Offers a platform allowing developers to build and deploy applications (e.g., Google App Engine)

SaaS (Software as a Service): Delivers software applications over the internet (e.g., Office 365, Salesforce)



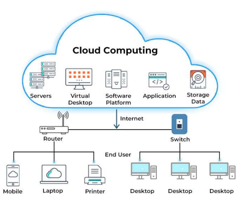
Cloud Computing

Advantages:

- **Scalability:** On-demand resource allocation
- **Cost-Efficiency:** Pay-as-you-go model
- **Accessibility:** Services available from anywhere with internet access

Challenges:

- **Security:** Cloud security concerns
- **Data Sovereignty:** Data location may be a concern for legal compliance

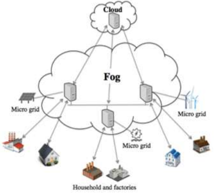


Fog Computing

Definition: An extension of cloud computing that brings computing, storage, and networking closer to devices at the edge of the network

Features:

- Processes data at the edge or near the source of data generation
- Reduces latency by performing computation closer to devices rather than sending data to centralized cloud servers



Applications: Used in Internet of Things (IoT) applications like smart cities, connected vehicles, and industrial automation

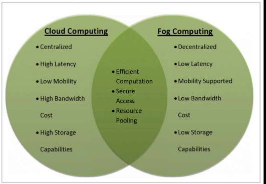
Fog Computing

Benefits:

- **Reduced Latency:** Ideal for real-time applications
- **Data Privacy:** Less data is sent to the cloud, improving security

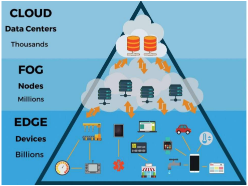
Challenge:

- **Complexity:** Managing distributed fog nodes can be complex



Edge Computing

Definition: Data processing takes place directly on devices or nearby, at the “edge” of the network, rather than being sent to a central cloud server



Advantages:

- **Lower Latency:** Faster response time by processing data locally
- **Bandwidth Efficiency:** Reduces the amount of data sent to the cloud

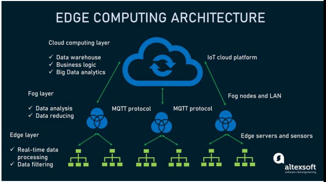
Edge Computing

Use Cases:

- **Autonomous Vehicles:** Real-time decision-making
- **Smart Homes/Devices:** Local processing in devices like smart cameras and thermostats

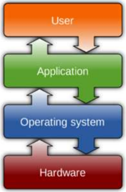
Challenges:

- **Security:** Ensuring data privacy and security at the edge



Introduction to Operating Systems

Definition: An operating system (OS) is system software that manages hardware, software, and resources in a computer system.

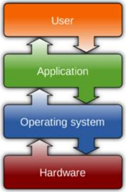


Functions:

- **Process Management:** OS manages processes and resources efficiently
- **Memory Management:** Allocates and tracks memory usage
- **File Management:** Organizes and manages files on storage devices
- **Device Management:** Manages input/output devices (e.g., printers, disk drives)

Introduction to Operating Systems

- **Today's topic: Introduction to Operating Systems (OS).**
- **We will cover:**
  - **Basics of OS**
  - **Types of OS**
  - **Key Components**
  - **Multitasking and multithreading**
  - **and more….**



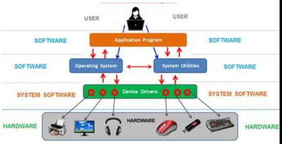
What is an Operating System?

- **OS: A bridge between hardware and software.**

- Manages hardware resources

- Provides a user-friendly environment for

  executing applications



Why is the OS Important?

- Central to every computing device
- Manages multitasking and resource allocation
- Allows us to use applications efficiently

Types of Operating Systems

- **Single-user OS (e.g., Windows)**
- **Multi-user OS (e.g., Unix)**
- **Real-time OS (e.g., RTOS for medical devices)**

Key Responsibilities of an OS

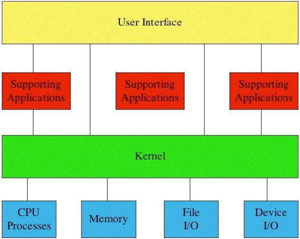
- **Process Management**
- **Memory Management**
- **File System Management**
- **Security and Access Control**

OS Components Overview

- **Kernel:** Core of the OS, manages resources
- **User Interface:** Provides ways for users to interact with the system (GUI/CLI)
- **Drivers:** Communicate between OS and hardware devices

What is a Kernel?

- Core of the OS
- Manages system resources
- Provides low-level interfaces



User Interface: CLI vs GUI

- **CLI:** Command-line interface, text-based control
- **GUI:** Graphical User Interface, visual interaction with icons and windows
- Both serve as user interfaces to interact with the OS

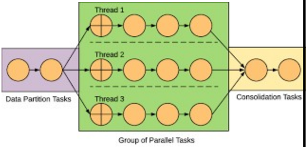
Process Management

- **What is a process?**

- **How does the OS handle multiple processes?**

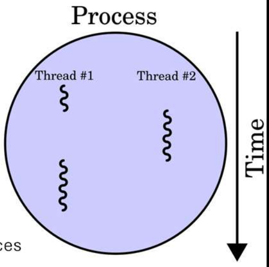
Multitasking & Multithreading Introduction

- **Multitasking**: Running several programs at the same time
- **Multithreading:** Running multiple parts (threads) of the same program in parallel
- Improves system efficiency and performance



Multithreading in Action

- **Threads:** Small tasks within a process
- **Example:** A web browser loading multiple tabs simultaneously
- Helps execute tasks faster and with fewer resources

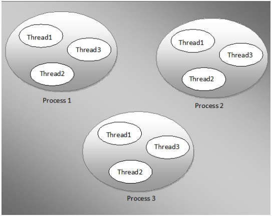


Why Multithreading?

- Faster execution
- Efficient resource utilization
- Improved performance

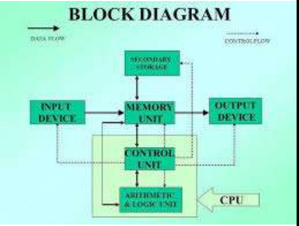
Thread vs Process

- **Thread:** Part of a process, shares resources with other threads
- **Process:** Independent program with its own memory space
- Threads allow finer control and faster execution compared to processes



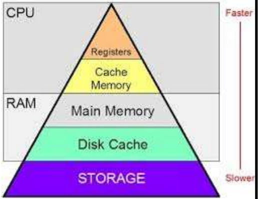
Memory Management

- Allocating and freeing memory for processes
- Ensures efficient use of system RAM
- Prevents issues like memory leaks or system crashes



Virtual Memory

- Extends physical memory by using disk space
- Allows running larger applications with limited physical RAM
- Virtual memory enables efficient multitasking

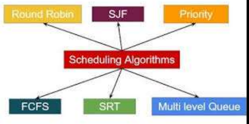


File Systems

- Hierarchical structure
- File naming conventions
- Data storage and retrieval

Disk Scheduling

- OS decides the order of reading/writing data from disk.
- Algorithms include FCFS (First Come First Serve), SSTF (Shortest Seek Time First).
- Optimizes speed and efficiency in data retrieval.

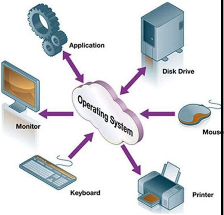


Security in Operating Systems

- User Authentication
- File Permissions
- Encryption

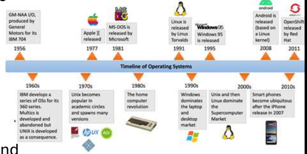
Device Drivers

- Drivers enable communication between hardware and OS
- Each device requires its own driver to function
- Without drivers, the OS cannot control hardware



Evolution of Operating Systems

- Batch processing: Early OS ran one task at a time
- Time-sharing: Multiple users shared processing power in real-time
- Modern OS: Multitasking, graphical interfaces, and network capabilities



Unix & Linux

- Multi-user, multitasking OS
- Open-source

Windows OS

- User-friendly
- Most popular desktop OS

macOS

- Unix-based
- Known for its design

Android and iOS

- Android: Open-source, used on most smartphones globally
- iOS: Closed-source, exclusive to Apple devices
- Both OS focus on mobile user experience and app ecosystems

Real-World OS Use Cases

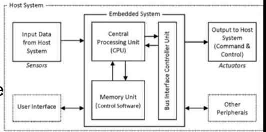
- Banking systems
- Healthcare devices
- Gaming consoles

OS in Embedded System

Embedded systems: OS designed for specific hardware

Found in appliances, cars, and medical device

Requires minimal resources and efficient real-time responses



Challenges in OS Design

- Security vulnerabilities
- Scalability

Open-Source OS vs Proprietary OS

Examples:

Linux (open-source)

Windows (proprietary)

Modern Trends in OS Development

- Cloud-based OS
- AI-enhanced OS

The Future of Operating Systems

- Quantum computing
- Autonomous systems

OS Fun Facts

- The first OS was developed in the 1950s
- The average person interacts with an OS more than 100 times a day

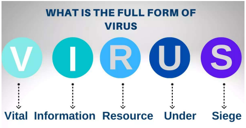
Review: What Have We Learned?

- What is an OS?
- Key OS components
- Multithreading and process management

Computer Virus and its Forms

- A type of malicious software (malware) designed to replicate itself and spread to other devices, often causing harm to the host computer

- It attaches itself to legitimate programs or files, and when the program is executed or the file is opened, the virus activates and can perform a range of damaging activities such as corrupting data, stealing information, or disabling system functions.



Computer Virus and its Forms

Forms of Computer Viruses:

1. File Infector Virus:

- Attaches itself to executable files (.exe, .com) or programs
- When the infected file is opened, the virus spreads.
- **Examples:** Cascade, Nimda

2. Boot Sector Virus:

- Infects the master boot record (MBR) of a hard drive
- The virus is activated when the computer starts up from an infected disk
- **Examples:** Michelangelo, Stone

Computer Virus and its Forms

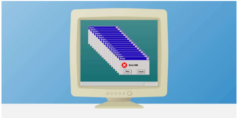
Forms of Computer Viruses:

3. Macro Virus:

- Written in macro language (like VBA) and targets applications such as Microsoft Word or Excel
- Spreads when infected documents are opened
- **Examples:** Melissa, Concept

4. Polymorphic Virus:

- Changes its code or structure with each infection, making it harder for antivirus software to detect
- **Example:** Storm Worm

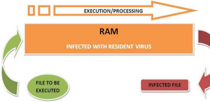


Computer Virus and its Forms

Forms of Computer Viruses:

5. Resident Virus:

- Installs itself in the system memory and can infect files as they are accessed
- Does not need a host program to execute
- **Example:** CMOS, Randex



6. Stealth Virus:

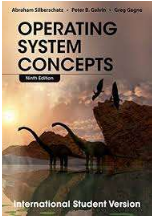
- Hides its presence by redirecting system functions or concealing its files
- Can bypass antivirus detection by temporarily removing itself
- **Example:** Rootkit, Frodo

Summary

- **Organizing Computer Facility**: Approaches to structure and manage computing resources.
- **Centralized Computing Facility**: All computing resources are located in a single, centralized location.
- **Distributed Computing Facility**: Resources are spread across multiple locations but work together.
- **Decentralized Computing Facility**: Independent computing facilities in various locations with minimal coordination.
- **Cloud Computing**: Remote servers store and process data, offering scalability and accessibility.
- **Fog Computing**: Extension of cloud computing closer to the network edge, reducing latency.
- **Edge Computing**: Data is processed at or near the source of data generation for faster responses.
- **Introduction to Operating Systems**: Software that manages computer hardware and software resources, providing services to programs.
- **Computer Virus and its Forms**: Malicious software that can replicate and spread, with various forms like file infector, macro, and boot sector viruses.

References

- Books: "Operating System Concepts" by Silberschatz et al.
- Online Resources: Linux.org, Microsoft Developer Network
- Articles: "Understanding Multithreading in OS" by TechCrunch



Thank You!