

Operating Systems & Programming Basics

- **Operating Systems** execute commands
 - Example: When you open **Command Prompt** in Windows and type dir, the operating system processes that command and lists the contents of the current directory.
- **A program application** is software designed to perform a specific task or function for users
 - Example: **Microsoft Word** is an application program designed to create, edit, and format text documents.
- **A workstation operating system** is the type of operating system that provides end users with the ability to access information and run applications locally while managing the hardware and software resources of the system.
 - Example: **Windows 10 Pro** is a workstation operating system used by professionals to run productivity software, connect to networks, and manage system hardware.
- **The Extended Support** is the operating system life cycle stage in which critical security updates will be supported
 - Example: **Windows 10** entered Extended Support on October 14, 2020, and will remain in that stage until October 14, 2025. During this period, Microsoft continues to provide critical security updates but no new features or design changes.
- **Breakpoint** is a debugging tool that allows the stopping of program execution
 - Example: In **Visual Studio**, you can set a breakpoint on a line of code. When the program reaches that line during execution, it pauses so you can inspect variable values and program flow.
- **Hadoop** is processing and analyzing large datasets

Programming & Scripting Tools

- **Visual Basic Script (VBScript)** is scripting or programming tool is based on COM
- **Windows and Linux** common tasks:
 - Create and delete files
 - Create and terminate a program
- **React** is preferred for web apps across platforms and used for front-end solutions.
- **Swift** is a programming language used for Apple iOS app development.
- **Java** is a programming language for Android OS development.
- **Python** is an example of an interpreted language and is suited for backend web interactions

Web Development & Frontend/Backend Tools

- **CSS** is used for formatting a website's UI.
- **Tailwind** is a utility-first CSS framework designed for styling web applications.
- **Node.js** is a JavaScript runtime environment for executing code outside a browser.
- A **variable** is a storage location in programming that holds a value that can be changed or retrieved during program execution
- **Gnome** is an example of a graphical user interface shell that is provided in Linux operating systems
- **CreateProcess()** is a system call in the Windows API used to create and manage processes, allowing new applications to run in the system.
- **"Something you know"** provides authentication method provides multiple different answers

Programming-Language Support & iOS GUI

- **Programming-Language Support** is a category of a system service that provides compilers, debuggers, and interpreters for common programming languages
- **SpringBoard** is the graphical user interface and application launcher for Apple's iOS devices, including iPads.

Memory Types

- **ROM (read-only memory)**
 - **Non-volatile:** Retains data even when the power is off.
 - **Read-only:** Typically, data stored in ROM cannot be modified (or it is difficult to modify).
 - Can be etched onto a silicon chip
- **RAM (random access memory)**
 - It is a **volatile memory**, meaning it loses all data when power is off.
 - Used by a computer to store data temporarily while it's working.
- **SODIMM** is a memory module that is used in laptops and other small devices
- **Memory Data Register (MDR)** is what stores the data that will be used by the **Memory Address Register (MAR)**
- **Memory Address Register (MAR)** is the control unit register responsible for holding the address of a memory location

- **Cache memory** is a small, high-speed type of memory located close to the CPU
- **Data movement** is the process of transferring data from one place to another.

Storage Devices

- **SSD (Solid State Drive)**
 - o No moving parts; faster and more durable.
 - o Energy efficient
 - o Durability Requirements: For systems exposed to physical movement (e.g., laptops, portable systems).
 - o Quick Boot and Load Times: Systems that need faster startup and application loading.
 - o When fast read/write speeds are needed
- **HDD (Hard Disk Drive)**
 - o Uses spinning disks and moving parts, slower but with larger capacity.
 - o More affordable (cost-effective)
 - o Ideal for storing massive amounts of data
 - o Situations where slower performance is acceptable
- **Flash Storage**
 - o Portability: When you need small, portable, and easy-to-use storage for transferring data between devices.
 - o Temporary or Lightweight Data: For quick, short-term backups or file sharing.
 - o Specialized Devices: Used in cameras, smartphones, and IoT devices due to compact size and compatibility.
 - o **Examples**
 - USB flash drives for sharing documents or presentations.
 - SD cards for cameras, drones, or other portable devices.
 - Embedded systems where small form factors are necessary (e.g., IoT devices).

Motherboards and chips

Motherboards

- **ATX** motherboard supports a full-size expansion slot
- **Mini-ITX** motherboard has a GPU mounted at a right angle next to the expansion slot
- **Riser card-based** motherboard has components right angled configuration for better cooling
- The **bus** is the motherboard component that acts as a signal pathway

- The **Northbridge** chip is the motherboard component primarily responsible for managing high-speed peripheral communication
- The **Southbridge** chip focuses on managing slower peripherals like USB ports and audio.

RAID Storage

RAID (Redundant Array of Independent Disks)

- Definition: A data storage virtualization technology that combines multiple drives.
- The primary purpose is to improve reliability
- Common RAID Levels:
 - **RAID 0**: Striping, no redundancy; improves performance.
 - **RAID 1**: Mirroring; duplicates data for fault tolerance.
 - **RAID 5**: Striping with parity; balance of performance and redundancy.
 - **RAID 10**: Provides high read and write performance by striping data across mirrored pairs of drives

Virtual Machines

Virtual Machines

- It is a software-based emulation of a physical computer, providing an isolated environment for running an operating system (OS) and applications. It operates using the resources of a physical host system but behaves as if it were a standalone system
- A **virtual sandbox** is an isolated environment specifically designed to test and run software, including legacy applications
- A **sandbox** is isolated and safe for testing changes without affecting other systems,
- A **test environment** simulates the real-world production environment.
- Components of a virtual machine
 - A **hypervisor** allows a virtual machine (VM) to let multiple users share an operating system.
 - **Type 1 Hypervisor** is both software and hardware that are needed to create a virtual environment and run directly on the host hardware.
 - **Type 2 Hypervisor** runs on top of a host OS.
- **Example: Parallels Desktop (for Mac)**
Used to run Windows or Linux VMs on macOS.
Example: Running Windows 10 on a MacBook for software testing.

Containers

- are lightweight, portable, and self-contained units that package an application along with its dependencies (libraries, binaries, configuration files) to ensure it runs consistently across different computing environments
- Share the host system's OS kernel, reducing overhead compared to virtual machines.
- Designed to work consistently across various environments
- Smaller than virtual machines as they only include the application and its dependencies (no guest OS).
- Faster startup times compared to VMs due to their minimal overhead.
- Can be spun up or removed quickly, making them ideal for scaling applications based on demand.
- Example: **Docker**
 - The most widely used container platform.
 - A Docker container running a Node.js app with MongoDB as a sidecar container.

Non-Functional Requirements (NFR)

- specifications that define how a system performs a function, rather than what the system does. They describe the quality attributes, constraints, or performance standards the system must meet.
- Analogy
 - "The car must be able to accelerate from 0 to 60 mph within 6 seconds."
 - This isn't about what the car *does* (like driving or turning), but *how well* it performs — its acceleration speed is a quality attribute, just like an NFR defines system performance or behavior.

Networking Devices I

- A **router** connects different networks and directs data between them.
- **Switch** connects devices within the same network and forwards data based on MAC addresses.
- **Hub** is a basic device that broadcasts data to all devices in a network segment (less common today).
- **Modem** modulates and demodulates signals for internet access over telephone lines or cable.
- **Access Point (AP)** provides wireless network access to devices.
- A **firewall** monitors and controls incoming and outgoing network traffic based on security rules.

- **The gateway** connects different networks that use different protocols.

Networking Devices II

- A **repeater** extends the range of a network by amplifying signals.
- A **bridge** connects two LAN segments to work as a single network.
- A **Network Interface Card (NIC)** is hardware that connects a computer to a network.
- **Category 4 cables** were designed for telephone and early networking applications, supporting data rates **up to 16 Mbps**
- **Protocols** define ground rules used by channel signals and messages within a network
- **Full-Duplex:** Signals can travel simultaneously in both directions

Networking Models & Security Devices

- In a **client-server** model, a central application provides resources and services to multiple client computers
- **Horizontal scaling** involves adding multiple servers to handle increased demand.
- **Intrusion Prevention System (IPS)** is a device actively interacts with or affects its environment when detecting attacks
- **Unified Threat Management (UTM)** provides multipoint protection through various security features combined into a single solution.
- **Virtual Private Network (VPN)** has a single firewall and provides protection. It also allows secure remote connections to a corporate network

Cloud Computing Models

Cloud computing service availability is measured in **nines**

- **Software as a Service (SaaS) - most common**
 - o SaaS delivers fully functional software applications over the internet, requiring no local installation.
 - o Accessible from any device via a web browser.
 - o Subscription-based pricing.
 - o Providers handle updates and security.
- **Examples of SaaS:**
 - o Google Workspace (Docs, Sheets, Gmail)
 - o Web based Email
 - o Microsoft 365 (Word, Excel, Outlook)
 - o Dropbox, Salesforce, Zoom
- **Use Cases of SaaS:**

- o cloud computing model suits a company needing IT help desks and other services
- **Infrastructure as a Service (IaaS)**
 - o provides virtualized computing resources over the internet, such as servers, storage, networking, and virtualization.
 - o On-demand computing infrastructure.
 - o Scalable resources (CPU, RAM, storage).
 - o Pay-as-you-go pricing model.
- **Examples of IaaS:**
 - o Amazon Web Services (AWS) EC2
 - o Microsoft Azure Virtual Machines
 - o Google Compute Engine (GCE)
- **Use Cases of IaaS:**
 - o Hosting websites and applications.
- **Desktop as a Service (DaaS)**
 - o DaaS delivers cloud-hosted virtual desktops to users, allowing access from any device.
 - o Secure remote desktop access.
 - o Centralized management of virtual desktops.
 - o Supports multiple operating systems.
 - o Pay-per-user model.
- **Examples of DaaS:**
 - o Amazon WorkSpaces
 - o Microsoft Azure Virtual Desktop
 - o Citrix Virtual Apps and Desktops
- **Use Cases of DaaS:**
 - o Remote work and telecommuting.
- **Platform as a Service (PaaS)**
 - o PaaS provides a managed environment for developers to build, test, and deploy applications without worrying about the underlying infrastructure.
 - o Includes development tools, databases, and runtime environments.
 - o Automates infrastructure management.
 - o Supports various programming languages.
 - o Developers focus on coding while the provider handles scaling and maintenance.
- **Examples of PaaS:**
 - o Google App Engine
 - o Microsoft Azure App Service

- o AWS Elastic Beanstalk
- **Use Cases of PaaS:**
 - o A category of internet and cloud-based services provides software development tools

Tech Stacks: MERN, MEVN, MEAN, LAMP

- **MERN (MongoDB, Express.js, React, Node.js)**
 - o It is a JavaScript-based stack used for building full-stack web applications. It consists of:
 - MongoDB – NoSQL database for data storage.
 - Express.js – Web framework for handling backend logic.
 - React – Frontend JavaScript library for building user interfaces.
 - Node.js – Runtime environment for executing JavaScript on the server side.
 - **When to Use MERN:**
 - o When building single-page applications (SPAs) with dynamic UI updates.
 - **Example Applications:**
 - o Facebook
 - o Netflix
 - o Airbnb
 - o Instagram
- **MEVN Stack (MongoDB, Express.js, Vue.js, Node.js)**
 - o It is like MERN but **replaces React with Vue.js**, another JavaScript framework for the frontend. It consists of:
 - MongoDB – NoSQL database.
 - Express.js – Backend web framework.
 - Vue.js – JavaScript framework for frontend UI.
 - Node.js – JavaScript runtime for server-side execution.
 - **When to Use MEVN:**
 - o When you need an easy-to-learn framework for frontend development.
 - **Example Applications:**
 - o Alibaba
 - o GibLab
 - o Grammarly Editor UI

- **MEAN Stack (MongoDB, Express.js, Angular, Node.js)**
 - It is a full JavaScript stack for building dynamic web applications. It includes:
 - MongoDB – NoSQL database.
 - Express.js – Backend web framework.
 - Angular – TypeScript-based frontend framework.
 - Node.js – JavaScript runtime for server-side execution.
- **When to Use MEAN:**
 - When developing large-scale, enterprise-grade web apps with complex frontend logic.
- **Example Applications:**
 - Google
 - Weather.com
 - Paypal
- **LAMP Stack (Linux, Apache, MySQL, PHP)**
 - It is a traditional open-source web stack used for backend-heavy applications. It consists of:
 - Linux – Operating system.
 - Apache – Web server.
 - MySQL – Relational database management system.
 - PHP – Programming language for backend logic.
- **When to Use LAMP:**
 - This should be recommended for configuring Apache servers
- **Example Applications:**
 - Blogging platforms
 - ♣ WordPress
 - Government and corporate portals
 - ♣ USDA Agricultural Marketing Service

Malware Types

Viruses

- A **computer worm** is a type of virus that does not need a host
 - **Example:** ILOVEYOU Worm (2000)
 - Spread via email without needing to attach to a host program
 - Caused billions in damage by overwriting files and spreading rapidly through contact lists
- **Keylogger** is a type of malicious software that records every keystroke made on a computer to capture sensitive information, such as passwords and credit card details
 - **Example:** Spyrix Keylogger
 - Records keystrokes, screenshots, and clipboard data
 - Often used to steal passwords, credit card numbers, and private messages
- **Boot Sector Virus** is a type of virus that attaches itself to the master boot record of a hard drive, making it difficult to detect and remove
 - **Example:** Michelangelo Virus
 - Infects the Master Boot Record (MBR) of disks
 - Activates on March 6th and can overwrite critical data
- **Ransomware** is a type of malware that encrypts a victim's files and demands payment for the decryption key
 - **Example:** WannaCry (2017)
 - Encrypted users' files and demanded Bitcoin for decryption
 - Affected over 200,000 systems in more than 150 countries
- **Trojan Horse** is a type of malware that disguises itself as legitimate software but performs malicious actions when executed
 - **Example:** Zeus Trojan
 - Masqueraded as a legitimate software or email attachment
 - Stole banking information and login credentials
- **Rootkit** is a type of malware that is designed to conceal its presence on a system, often granting unauthorized access to attackers
 - **Example:** Sony BMG Rootkit (2005)

- Installed with some Sony music CDs to prevent copying
- Hid its presence, making it difficult to detect and remove, and opened systems to other threats

Practice test

- [D386 Practice Questions](#)