Operating System Unit 2 Notes.

Introduction to Operating System

What is an Operating System?

An operating system (OS) is software that acts as an intermediary between computer hardware and users. It manages hardware resources, provides a user interface, and facilitates the execution of application software.

History of Operating Systems

- **1950s-1960s:** Early systems were batch processing systems where jobs were processed in batches without user interaction.
- **1960s-1970s:** Introduction of time-sharing systems, allowing multiple users to interact with a computer simultaneously.
- **1980s:** Development of personal computer operating systems (e.g., MS-DOS, early versions of Windows).
- **1990s-2000s:** Emergence of graphical user interfaces (GUIs) and multi-tasking capabilities (e.g., Windows 95, Mac OS).
- **2000s-Present:** Rise of mobile operating systems (iOS, Android) and cloud-based systems, alongside continuous evolution in desktop OSs.

Computer Hardware & Software

- **Hardware:** Physical components of a computer (CPU, memory, storage, peripherals).
- **Software:** Programs and applications that run on hardware, including system software (OS, drivers) and application software (productivity tools, games).

Different Operating Systems

- **Desktop OS:** Windows, macOS, Linux distributions (Ubuntu, Fedora).
- **Mobile OS:** Android, iOS.
- **Server OS:** Windows Server, Linux Server, UNIX.
- **Embedded OS:** Real-time operating systems used in devices like routers and IoT devices (e.g., RTOS).

Various System Software Associated with Operating Systems

- **Device Drivers:** Enable communication between the OS and hardware devices.
- **Utility Programs:** Perform maintenance tasks (disk management, antivirus).
- **Firmware:** Low-level software embedded in hardware for basic functionality.
- **Virtual Machine Monitors:** Enable virtualization, allowing multiple OS instances on a single hardware platform.

Shell and Kernel

- **Kernel:** Core component of the OS that manages system resources and hardware communication. It operates in a privileged mode and is responsible for process management, memory management, and device management.
- **Shell:** User interface for accessing the services of the OS. It can be command-line (CLI) or graphical (GUI). The shell interprets user commands and communicates with the kernel.

System Calls and Their Types

- **System Calls:** APIs provided by the OS to allow user applications to request services (e.g., file operations, process control).

Types of System Calls:

1. **Process Control:** Create, terminate, and manage processes (e.g., fork, exec).
2. **File Management:** Operations on files (e.g., open, read, write, close).
3. **Device Management:** Interact with device drivers (e.g., read/write to devices).
4. **Information Maintenance:** Manage system information (e.g., get time, get process status).
5. **Communication:** Handle inter-process communication (e.g., message passing, sockets).

Implementation: System calls typically involve switching from user mode to kernel mode, ensuring safe and controlled access to hardware resources.

Operating System Unit 1 Notes.

System Software vs. Application Software

System Software:

- This is the software that manages hardware and provides services for application software.
- Examples include operating systems (like Windows and Linux), device drivers, and utility programs.

Application Software:

- This is designed to help users perform specific tasks.
- Examples include word processors, spreadsheets, and presentation software.

Concepts of Files and Folders

- **Files:** These are units of storage that hold data. Each file has a name and an extension (like .docx for Word documents or .xlsx for Excel spreadsheets).
- **Folders:** These are containers used to organize files. Folders can hold files and other folders, helping users keep their data structured.

Basic Features of GUI Operating Systems: Windows & Linux

Windows:

- **Start Menu:** Central access point for applications, settings, and files.
- **Taskbar:** Displays open applications and allows quick access to frequently used programs.
- **File Explorer:** Manages files and folders through a visual interface.
- **Control Panel/Settings:** For adjusting system settings and configurations.

Linux:

- **Desktop Environments:** Various GUIs like GNOME, KDE, etc., offer different user experiences.
- **Terminal Access:** Users can interact with the system via command line, providing powerful control.
- **File Manager:** Graphical interface for managing files and folders, similar to Windows Explorer.
- **Package Managers:** Tools for installing and managing software packages.

Programming Languages, Compiler, and Interpreter

- **Programming Languages:** These are formal languages used to write software. Examples include Python, Java, C++, and JavaScript.
- **Compiler:** A program that translates source code written in a high-level language into machine code (binary) that a computer can execute.
- **Interpreter:** A program that directly executes instructions written in a programming language, translating them line by line at runtime.

Databases

- A database is an organized collection of data that can be easily accessed, managed, and updated.
- Common types include relational databases (like MySQL, PostgreSQL) and NoSQL databases (like MongoDB).

Application Software: Generic Features

Word Processors (e.g., Microsoft Word):

- Text editing and formatting tools.
- Spell check and grammar check.
- Insertion of images, tables, and hyperlinks.
- Page layout options and printing capabilities.

Spreadsheets (e.g., Microsoft Excel):

- Cell-based layout for organizing data.
- Functions and formulas for calculations.
- Data visualization tools like charts and graphs.
- Sorting and filtering capabilities.

Presentation Software (e.g., Microsoft PowerPoint):

- Slide creation and design tools.
- Insertion of multimedia (images, videos, audio).
- Transition and animation effects.
- Presentation mode for displaying slides to an audience.

Short Notes on Computer Concepts

System Software vs. Application Software

- **System Software:** Manages hardware, includes operating systems (Windows, Linux), device drivers, and utilities.
- **Application Software:** Helps users perform specific tasks (e.g., word processors, spreadsheets).

Concepts of Files and Folders

- **Files:** Units of storage with names and extensions (e.g., .docx, .xlsx).
- **Folders:** Containers for organizing files and subfolders, aiding data management.

Basic Features of GUI Operating Systems

Windows:

- **Start Menu:** Access to applications and settings.
- **Taskbar:** Displays open apps; quick access to frequently used programs.
- **File Explorer:** Visual file and folder management.
- **Control Panel/Settings:** System configuration options.

Linux:

- **Desktop Environments:** Various GUIs (e.g., GNOME, KDE) for user experience.
- **Terminal Access:** Command-line interaction for advanced control.
- **File Manager:** Visual tool for file management.
- **Package Managers:** For software installation and management.

Programming Languages, Compiler, and Interpreter

- **Programming Languages:** Formal languages for software development (e.g., Python, Java).
- **Compiler:** Translates high-level code into machine code.
- **Interpreter:** Executes high-level code line by line at runtime.

Databases

- Organized collections of data; easily accessible and manageable.
- Types: Relational (e.g., MySQL) and NoSQL (e.g., MongoDB).

Application Software: Generic Features

Word Processors (e.g., MS Word):

- Text editing and formatting.
- Spell and grammar check.
- Insert images, tables, hyperlinks.
- Page layout and printing options.

Spreadsheets (e.g., MS Excel):

- Cell-based layout for data organization.
- Functions and formulas for calculations.

- Data visualization (charts/graphs).
- Sorting and filtering capabilities.

Presentation Software (e.g., MS PowerPoint):

- Slide creation and design tools.
- Multimedia insertion (images, videos).
- Transitions and animations.
- Presentation mode for audience display.