Question 1)

An**Operating System** is a System software that manages all the resources of the computing device.

* Acts as an interface between the software and different parts of the computer or the computer hardware.
* Manages the overall resources and operations of the computer.
* Controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer.
* Examples of Operating Systems are Windows, Linux, macOS, Android, iOS, etc.

Features:

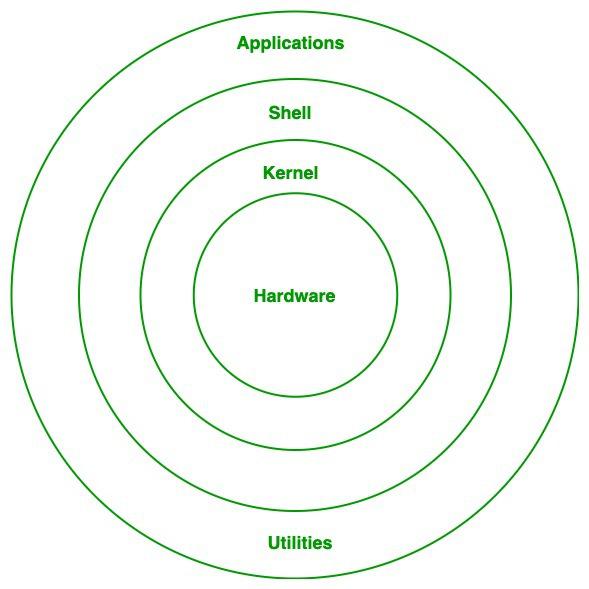
* **Memory Management:**  
  The operating system manages the main memory by tracking allocated and unallocated memory addresses. It decides how long a process stays in memory and in what order processes access it, ensuring efficient memory use. When a process requests memory, the OS allocates it, and deallocates it when the process ends.
* **Processor Management:**  
  The OS manages the allocation of the CPU, deciding which process gets access and for how long, especially in a multiprogramming environment. It tracks process statuses using a traffic controller program, allocates processing time, and deallocates it when a process is finished.
* **Device Management:**  
  The OS oversees communication between devices and their respective drivers. It tracks all connected devices and manages access, ensuring that devices are allocated to processes efficiently. Devices are deallocated when no longer in use, ensuring resource optimization.
* **File Management:**  
  The OS organizes and tracks files within a computer’s file system, which is usually structured in directories. It manages access and ensures that files are stored and retrieved efficiently.
* **I/O Management:**  
  The OS manages input/output devices, such as the mouse, keyboard, and printer. It handles I/O operations synchronously (CPU waits for I/O) or asynchronously (I/O occurs while the CPU continues processing). The OS supports communication with I/O devices using special instructions or direct memory access.
* **Security Management:**  
  The OS ensures the security of confidential data and prevents unauthorized access to the system. It helps protect against malware and uses security features like firewalls, operating in supervisor mode to maintain system integrity and safety.

Question 2)

The Linux Operating System is based on the Linux Kernel, which acts as the core of the system, managing hardware and resource interactions to ensure smooth operation. However, the Linux Kernel alone is insufficient for a complete operating system. To create a fully functional system, the kernel is combined with various software packages and utilities, forming what are known as Linux distributions. These distributions make the operating system user-friendly, secure, and ready to support a range of applications and tasks. Linux distributions are available in different Flavors, each designed to meet specific user needs, preferences, and use cases, offering flexibility for both general users and specialized environments.

**Components of Linux:**

Like any operating system, Linux consists of software, computer programs, documentation, and hardware. **The main components of Linux operating system are:** Application, Shell, Kernel, Hardware, Utilities



[Kernel](https://www.geeksforgeeks.org/introduction-to-linux-operating-system/) is the main core component if Linux, it controls the activity of other hardware components. It visualizes the common hardware resources and provide each process with necessary virtual resources. It makes the process to wait in the ready queue and execute in consequently to avoid any kind of conflict.

[Shell](https://www.geeksforgeeks.org/introduction-linux-shell-shell-scripting/)can be determined as the interface to the kernel, which hides the internal execution of functions of kernel from the user. Users can just enter the commend and using the kernel’s function that specific task is performed accordingly.

System libraries are some predefined functions by using which any application programs or system utilities can access kernel’s features. These libraries are the foundation upon which any software can be built.

Hardware layer of Linux is the lowest level of operating system track. It is playing a vital role in managing all the hardware components. It includes device drivers, kernel functions, memory management, CPU control, and I/O operations. This layer generalizes hard complexity, by providing an interface for software by assuring proper functionality of all the components.

System utilities are the commend line tools that preforms various tasks provided by user to make system management and administration better. These utilities enable user to perform different tasks, such as file management, system monitoring, network configuration, user management etc.

Question 3)

**Case Study: Installation and Exploration of Various Operating Systems on a Virtual Machine**

**Objective:**

To explore the installation and functionality of different types of operating systems (OS) on a virtual machine. This study focuses on **Windows 10, Ubuntu Linux, and Arch Linux** to analyze their installation processes, system requirements, and key features.

**2. Setting Up the Virtual Environment**

**2.1 Choosing a Virtualization Tool**

We use **Oracle VirtualBox** as it supports multiple OS types and provides flexibility.

**2.2 System Requirements**

| **Component** | **Minimum Requirements** |
| --- | --- |
| Processor | AMD Ryzen 5 or higher |
| RAM | 8GB (16GB recommended) |
| Storage | 100GB free space |
| Software | VirtualBox, OS ISO files |

**3. Installation and Exploration of Operating Systems**

**3.1 Windows 10 Installation**

1. **Download Windows 10 ISO** from Microsoft.
2. **Create a New VM** in VirtualBox:
   * Allocate **4GB RAM, 50GB Storage**
   * Select **Windows 10 (64-bit)** as OS type.
3. **Boot the ISO and Install Windows 10**:
   * Choose language and edition.
   * Format disk and install.
4. **Post-Installation Configuration**:
   * Install VirtualBox Guest Additions.
   * Update drivers and security patches.
5. **Exploration**:
   * User-friendly interface, Start Menu, Cortana.
   * Microsoft Store, multitasking, system settings.

**3.2 Ubuntu Linux Installation**

1. **Download Ubuntu ISO** from the official site.
2. **Create a New VM**:
   * Allocate **2GB RAM, 25GB Storage**
   * Select **Linux → Ubuntu (64-bit)**.
3. **Boot and Install Ubuntu**:
   * Choose “Install Ubuntu.”
   * Select keyboard layout and partition disk.
4. **Post-Installation**:
   * Install updates and additional software (VLC, LibreOffice).
5. **Exploration**:
   * GNOME desktop, Terminal access.
   * Package management with APT.
   * Strong security and open-source flexibility.

**3.3 Kali Linux Installation**

Kali Linux is a Debian-based distribution specifically built for **penetration testing, digital forensics, and security research**.

**Installation Process:**

1. **Download the Kali Linux ISO** from the official website.
2. **Create a New VM** in VirtualBox:
   * Allocate **4GB RAM (8GB recommended), 50GB Storage**.
   * Select **Linux → Debian (64-bit)**.
3. **Boot the Kali Linux ISO and Start Installation**:
   * Select **"Graphical Install"**.
   * Choose language, keyboard layout, and location.
   * Create a user account and set a strong password.
4. **Disk Partitioning**:
   * Choose **"Guided - use entire disk"** (for beginners).
   * Select the partition scheme.
5. **Install the Base System and GRUB Bootloader**:
   * Confirm GRUB installation on the main disk.
   * Wait for the installation to complete and reboot the system.

**4. Comparison of OS Features**

| **Feature** | **Windows 10** | **Ubuntu Linux** | **Kali Linux** |
| --- | --- | --- | --- |
| **User-Friendliness** | High | Medium | Low |
| **Security** | Medium | High | Very High |
| **Software Availability** | High | Medium | Medium |
| **Customization** | Low | High | Very High |
| **Performance** | Medium | High | High |
| **Package Management** | Windows Store | APT | APT |
| **Target Users** | General users, gamers | Developers, IT pros | Cybersecurity experts, hackers |

Question 4)

**1. MS-DOS**

**Features:**

* Single-user OS.
* Lightweight and allows direct BIOS access.
* Loads data from external sources to internal memory.
* Supports basic file management (create, edit, delete).
* Controls external devices like printers, keyboards.

**Drawbacks:**

* No multi-user support.
* No GUI, requires command-line input.
* No multiprogramming support.
* Lacks memory protection and stability.
* Cannot address more than 640 MB RAM.

**2. Windows OS**

**Features:**

* Designed for x86 Intel and AMD processors.
* GUI-based, eliminating command-line dependency.
* Supports multi-core processors.
* Large software compatibility and backward compatibility.
* Automatic hardware detection.

**Drawbacks:**

* Expensive (paid license and applications).
* High system requirements.
* Performance slows with heavy program loads.
* Vulnerable to viruses.

**3. Linux OS**

**Features:**

* Open-source and free to use.
* Portable across various devices.
* Multi-user, multitasking support.
* Secure with authentication and threat detection.
* Provides a hierarchical file structure.

**Drawbacks:**

* No standard edition, leading to confusion.
* Limited application support.
* Some hardware may not be compatible.
* Steeper learning curve for beginners.

**4. Solaris OS**

**Features:**

* Scalable with high performance.
* Reliable, rarely crashes.
* Built for network computing.
* Advanced security features.
* Supports software interoperability.

**Drawbacks:**

* Expensive enterprise OS.
* Poor GUI support.
* Limited hardware compatibility.
* May become unstable under high load.

**5. Symbian OS**

**Features:**

* Microkernel-based with efficient memory management.
* Supports third-party software.
* User-friendly interface.
* Applications written in C++, Python, Java, etc.
* Efficient and stable.

**Drawbacks:**

* Slower responsiveness.
* Highly vulnerable to viruses.
* No virtual memory support.

**6. Android OS**

**Features:**

* Open-source and customizable.
* Supports 2D/3D graphics, GSM connectivity.
* Large app ecosystem (Google Play Store).
* Optimized for ARM architecture.

**Drawbacks:**

* Java dependency complicates UI development.
* Background apps drain battery.
* Performance issues with multiple processes.
* More susceptible to security threats.

**7. iOS**

**Features:**

* Closed-source OS optimized for Apple devices.
* Fast and responsive user interface.
* High-performance and security.
* App Store offers high-quality apps.
* Regular security updates.

**Drawbacks:**

* Closed-source, limiting beta testing.
* Large memory footprint for apps.
* Less customization than other OS.
* No third-party app installations.

**8. FreeBSD**

**Features:**

* Robust and stable, ideal for servers.
* High performance with advanced memory management.
* Strong security measures.
* Flexible and customizable.
* Large software repository.

**Drawbacks:**

* Limited hardware support.
* More complex to use.
* Smaller community than Linux.

**9. Chrome OS**

**Features:**

* Fast and simple user interface.
* Web-based applications.
* Seamless integration with Google services.
* Automatic security updates.

**Drawbacks:**

* Limited offline functionality.
* Fewer software options.
* Best suited for Chromebooks only.
* Heavy reliance on Google services.

**10. macOS**

**Features:**

* User-friendly interface.
* Integrates seamlessly with Apple devices.
* Preloaded with high-quality default apps.
* Reliable and stable performance.

**Drawbacks:**

* Expensive hardware.
* Limited to Apple devices.
* Restricted third-party software access.
* Limited customization options.

Question 5)

**1. Memory Management**

The OS keeps track of memory allocation, ensuring that each process gets the required memory space while preventing conflicts between processes. It efficiently manages memory in a multiprogramming environment by deciding how and when memory is allocated or deallocated. This helps in optimizing system performance and avoiding memory leaks or fragmentation.

**2. Processor Management**

In a multi-programming environment, the OS is responsible for scheduling tasks, deciding which process gets CPU time and for how long. It ensures that all processes run smoothly without any process monopolizing the CPU. Additionally, it keeps track of process statuses using a process scheduler and traffic controller to maintain system efficiency.

**3. Device Management**

The OS manages input and output devices by communicating with their respective drivers to ensure smooth operation. It keeps track of all connected devices, decides which process gets access to a device, and for how long. Moreover, it efficiently allocates and deallocates devices to processes, preventing conflicts and optimizing resource utilization.

**4. File Management**

The OS maintains a structured file system that organizes data into directories for easy access and retrieval. It keeps track of file storage locations, user access permissions, and file modifications. It also ensures data integrity and prevents unauthorized access by enforcing security policies on files and directories.

**5. I/O Management**

Input and output operations are handled by the OS to ensure smooth communication between the system and external devices like keyboards, printers, and monitors. It processes requests from these devices, performs necessary tasks, and returns results to the requesting application. Efficient I/O management helps in improving system responsiveness and device performance.

**6. User Interface**

The OS provides an interface for users to interact with the system, either through a command-line interface (CLI) or a graphical user interface (GUI). The GUI makes it easier for users to navigate and operate the system using visual elements, while CLI provides advanced users with direct command execution. This interface enables users to control applications, files, and hardware seamlessly.

**7. Booting the Computer**

Booting is the process of starting or restarting a computer by loading the OS into memory. A cold boot occurs when the computer is powered on from a completely off state, while a warm boot happens when the system is restarted without turning off the power. The OS initializes hardware components and prepares the system for user interaction.

**8. Security**

Operating systems provide various security measures to protect user data and system resources. It enforces authentication methods like passwords and biometric scans to prevent unauthorized access. Additionally, it includes firewalls and encryption techniques to guard against cyber threats and maintain data confidentiality.

**9. System Performance Control**

The OS optimizes system performance by efficiently managing CPU, memory, and I/O resources. It implements process scheduling techniques to ensure smooth multitasking and prevent system slowdowns. Monitoring tools provided by the OS help users track system performance and take corrective actions if needed.

**10. Job Accounting**

The OS maintains records of resource usage by different users and processes, helping administrators track system activity. It logs CPU time, memory usage, and disk operations for each task, allowing for performance analysis and resource optimization. This information can also be used for billing in environments where computing resources are shared among multiple users.

**11. Error Detection**

The OS constantly monitors the system to identify and prevent errors that could cause crashes or data loss. It detects hardware failures, software bugs, and security breaches, alerting users about potential risks. Error handling mechanisms are implemented to recover from failures and maintain system stability.

**12. Coordination Between Software and Users**

An OS acts as an intermediary between different software programs and users, ensuring they function smoothly together. It coordinates resource sharing among multiple applications and prevents conflicts between programs. This allows users to run multiple applications simultaneously without disruptions.

**13. Basic Computer Task Management**

The OS handles the management of peripheral devices like keyboards, printers, and storage drives. Most modern operating systems support plug-and-play functionality, automatically recognizing and configuring newly connected devices. This reduces the need for manual driver installations and enhances user convenience.

**14. Network Management**

The OS enables network communication by managing data transmission between computers and internet services. It ensures secure and efficient data transfer using network protocols and monitoring tools. Additionally, it helps users set up Wi-Fi and Ethernet connections, optimizing internet speed and security.

**15. OS Services**

Operating systems provide essential services such as user interface management, program execution, and file handling. They facilitate input/output operations, resource allocation, and communication between applications. These services ensure a seamless computing experience by simplifying complex system functions for users.