Scoring Indicators

COURSE NAME :Server Administration

COURSE CODE :6131B **QID : 2102240134**

Q No	Scoring Indicators	Split score	Total score
	PART A		9
I. 1	chown	1	
I. 2	chgrp	1	
I. 3	useradd	1	
I. 4	usermod	1	
I. 5	ifconfig	1	
I. 6	vsftpd.conf	1	
I. 7	Secure Shell	1	
I. 8	Dynamic Host Configuration Protocol	1	
I. 9	Network File System.	1	
	PART B		24
II. 1	The command sudo apt-get install [package_name]. This command will download and install the specified package along with any necessary dependencies.	Apt-get – 1Mark Explanation – 2 Marks	3
II. 2	for disk mounting, the mount command can be used. For instance, to implement the mounting of a partition on /dev/sdb1 to a directory /mnt/data, one would execute the command sudo mount /dev/sdb1 /mnt/data.	Command – 1 Mark Example – 1 Mark Explanation – 1 Mark	3
II. 3	Apache offers some of the following benefits and advantages: ■ It is stable. ■ It is used, backed, and supported by several major sites and organizations. ■ The entire program and related components are open source. ■ It works on a large number of platforms (all popular variants of Linux/UNIX, some of the not-so-popular variants of UNIX, and even Microsoft Windows). ■ It is extremely flexible. ■ It has proved to be secure.	Any 3 points – 3 Marks each	3
II. 4	Edit the sshd configuration file sshd_config and change the value of the key port. sudo nano /etc/ssh/sshd_config, and use the line Port 2222 to change the default SSH port to 2222.	File name – 1 mark Explanation 2 Marks	3

II. 5	The uttilitieslike **ifconfig** or **ip** are used to view current network interface settings, including IP addresses, subnet masks, and MAC addresses To use ifconfig to display all the network interfaces available on the system, type [root@server ~]# ifconfig -a To use ip to display all the network interfaces available on the system, type [root@server ~]# ip link show	Commands – 2 Marks Explanation – 1 Mark	3
II. 6	To apply route configuration, one can use the route add command. For example, applying sudo route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.1 would add a route to the 10.0.0.0/24 network.	Command – 1 Mark Example – 2 Mark	3
II. 7	The ***dump*** tool works by making a copy of an entire file system. The ***restore*** tool can then pull any and all files from this copy. The dump utility stores all the information about its dumps in the /etc/dumpdates file. This file lists each backed-up file system, when it was backed up ***tar*** was originally meant to create archives of files onto tape (tar = tape archive). ***rsync*** utility, which is used for synchronizing files, directories, or entire file systems from one location to another. The location could be from a local system to another networked system, or it could be within the local file system.	Tool name(Any one) – 1 Mark Explanation – 2 Marks	3
II. 8	**lpr** The lpr command is used to print documents. Most PostScript and text documents can be printed by directly using the lpr command. **lpr test-page.txt** This will print/send the document test-page.txt to the default printer, which is usually the first printer that was installed. **lpq** After you have submitted the job, you can view what is on the print spooler by using the lpq command. If you've just printed a job and notice that it doesn't come out of the printer, use the lpq command to display the current list of jobs that are spooled on the printer. **lprm** When you suddenly realize that you didn't mean to print the document you just printed, you might have a chance to delete it before it gets printed. To do this, use the lprm command. This will unspool the print job from the printer. **lprm 2**	Command – 1 Marks each Explanation – 3 marks Example – 1 Mark	3
II.9	Dynamic Host Configuration Protocol -The client machine is configured to obtain its IP address from the network. When the DHCP client software is started, it broadcasts a request onto the network for an IP address. If all goes well, a DHCP server on the network will respond, issuing an address and other necessary information to complete the client's network configuration.	Full form – 1 Mark Explanation – 2 Marks	3

II.10	In summary, Samba is a software suite that facilitates interoperability between Unix/Linux servers and Windows clients, allowing seamless file and print sharing across different operating systems.	Explanation – 3 Marks	3
	PART C		42
III	Building the Linux kernel involves compiling the kernel source code to create a custom kernel image tailored to the specific requirements of a system. Below are the steps involved in building the Linux kernel: 1. **Download Kernel Source:** First, you'll need to download the kernel source code. You can obtain it from the official Linux kernel website (https://www.kernel.org/) or by using Git to clone the source repository. 2. **Extract the Source Code:** If you downloaded a tarball, extract it to a directory of your choice: ```bash tar xvf linux-x.y.z.tar.gz cd linux-x.y.z ``` Replace `x.y.z` with the version number of the kernel you downloaded. 3. **Preparing to Configure the Kernel:** You can configure the kernel using one of the following methods: - Use the default configuration: `make defconfig` - Use an existing configuration: `cp /boot/config-$(uname -r) .config` - Use a graphical configuration tool: `make menuconfig`, `make xconfig`, or `make nconfig` Choose the appropriate configuration method and customize the kernel options if necessary. 4. **Compile the Kernel:** Once you've configured the kernel, you can start the compilation process: ```bash make ``` This command will compile the kernel 5. **Install the Kernel Modules:** After compilation, you can install the kernel modules: ```bash sudo make modules_install ``` 6. **Install the Kernel Image:** Copy the kernel image to the `/boot` directory and update the bootloader configuration: ```bash # cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.20-csc262-lab # cp -v System.map /boot/System.map-2.6.20-csc262-lab # new-kernel-pkg --mkinitrd --dracut --depmod --install <kernel-version> ```	Steps – 4 Marks Explanation – 3 Marks	

	7. **Reboot:** Finally, reboot your system to boot into the newly compiled kernel: ```bash sudo reboot		
IV	Applying a patch to the Linux kernel involves integrating changes from a patch file into the kernel source code. Here are the general steps for applying a patch to the Linux kernel: 1. Copy the compressed patch file that you downloaded into a directory one level above the root of your target kernel source tree. Assuming, for example, that the kernel you want to patch has been decompressed and untarred into the /usr/src /linux-4.2/ directory, you would copy the patch file into the /usr/src/ directory. 2. Change your current working directory to the top level of the kernel source tree. This directory in our example is /usr/src/linux-4.2/. Type the following: [root@server ~]# cd /usr/src/linux-4.2/ 3. It is a good idea to do a test run of the patching process to make sure there are no errors and that the new patch will indeed apply cleanly: [root@server linux-4.*]# xz -dc../patch-4.2.5.xz | patch -p1 --dry-run 4. Assuming the preceding command ran successfully without any errors, you're now ready to apply the patch. Run this command to decompress the patch and apply it to your kernel: [root@server linux-4.*]# xz -dc ../patch-4.2.5.xz | patch -p1 Here, ../patch-4.2.5.xz is the name and path to the patch file. A stream of filenames is printed out to your screen. Each of those files has been updated by the patch file. If any problems occurred with the upgrade, you will see them reported here.	Steps – 4 Marks Explanation – 3 Marks	
V	The role of package managers in Linux distributions is pivotal to simplifying software installation, management, and maintenance. Package managers handle the installation, upgrading, configuration, and removal of software packages on a Linux system. Here's an explanation of their role and how they streamline these processes: 1. **Centralized Package Repositories:** - Package managers operate with centralized repositories, which are online databases containing a vast collection of software packages. These	Any 7 points- 1 Mark each	7

repositories are maintained by the Linux distribution's maintainers, ensuring that the software is tested, compatible, and secure.

2. **Dependency Resolution:**
   - One of the primary functions of package managers is to manage dependencies. Software often relies on other libraries or packages to function correctly. Package managers automatically identify and install these dependencies, ensuring that all required components are present for the software to run.

3. **Installation Process:**
   - Users can request the installation of a specific software package through the package manager. The package manager retrieves the package from the repository and installs it on the system. This process is simplified, eliminating the need for manual downloading, compilation, and resolving dependencies by hand.

4. **Upgrading Software:**
   - Package managers make it easy to keep installed software up to date. Users can simply instruct the package manager to check for updates, and it will compare the installed versions with those available in the repository. If updates are available, the package manager can download and install them automatically.

5. **Removal of Software:**
   - Uninstalling software is straightforward with package managers. Users can instruct the package manager to remove a specific package, and it takes care of uninstalling the software, cleaning up dependencies, and freeing up disk space.

6. **Configuration Management:**
   - Package managers often handle the configuration of software packages. Configuration files can be distributed with the software, and the package manager may prompt users during installation or update to merge or keep their existing configurations.

7. **Rollback and Version Control:**
   - Some advanced package managers support rollback functionality, allowing users to revert to previous software versions if an update causes issues. This provides a safety net in case an update introduces unforeseen problems.

8. **Security:**
   - Package managers play a crucial role in enhancing system security. They provide a secure and controlled method for software distribution, reducing the risk of downloading compromised or malicious software from untrusted sources.

9. **Consistency Across Systems:**
   - Package managers contribute to system consistency by enforcing uniformity in software installation and management across multiple instances of the same Linux distribution. This ensures that software behaves predictably and consistently across different systems.

In summary, package managers in Linux distributions simplify software management by providing centralized repositories, handling dependencies,

	automating installation and upgrades, managing configurations, and contributing to system security. Their role streamlines the process of software installation and maintenance, making it efficient and user-friendly for Linux users.		
VI	1. **Process Management:** - The kernel manages processes, including creating and terminating processes, scheduling them for execution, and allocating system resources such as CPU time and memory to ensure efficient multitasking. 2. **Memory Management:** - The kernel oversees the allocation and deallocation of memory to processes. It manages virtual memory, paging, and swapping to optimize the use of physical RAM and provide a consistent memory space for applications. 3. **Device Driver Management:** - The kernel provides a layer between hardware devices and user applications through device drivers. It loads and manages these drivers, facilitating communication between software and hardware components. 4. **File System Management:** - The kernel handles file systems, providing an interface for file creation, deletion, reading, and writing. It manages file permissions, directories, and storage devices, ensuring data integrity and security. 5. **System Calls:** - The kernel exposes system calls, which are interfaces for applications to request services from the kernel. These services include I/O operations, process management, and communication between processes. 6. **Interrupt Handling:** - The kernel manages hardware and software interrupts, responding to events such as I/O completion or hardware errors. It ensures that the appropriate interrupt service routine is executed to handle these events. 7. **Security and Access Control:** - The kernel enforces security policies, controlling access to system resources based on user permissions and privileges. It protects critical system components and prevents unauthorized access or malicious activities. 8. **Networking:** - The kernel manages network protocols and communication, handling tasks such as routing, packet forwarding, and socket creation. It facilitates network connections and data transfer between applications. 9. **System Initialization and Booting:** - The kernel initializes the system during boot, configuring hardware, loading necessary modules, and transitioning the system to a usable state. It manages the startup process, including initializing user-space processes. 10. **Interprocess Communication (IPC):** - The kernel provides mechanisms for processes to communicate with each other, including shared memory, message passing, and synchronization primitives. This enables collaboration between different parts of a system.	Any 4 points- 4 Mark Explanation – 3 Marks	7

	11. **Error Handling:** - The kernel is responsible for detecting and handling errors to maintain system stability. It logs errors, provides error codes, and takes appropriate actions to prevent system crashes or data corruption. 12. **Power Management:** - The kernel manages power-related functions, including handling sleep and wake events, managing CPU frequency scaling, and coordinating power-saving features to optimize energy efficiency. Understanding these tasks demonstrates the central role of the kernel in coordinating and managing various aspects of system operation, ensuring the reliable and efficient functioning of the entire computing environment.		
VII	1. **Identify the Runaway Process:** - Use the `top` command to identify the process consuming excessive resources. Look for processes with high CPU or memory utilization. Note the process ID (PID) of the runaway process. ```Command: top``` Identify the process with abnormal resource usage. 2. **Gather Detailed Process Information:** - Use the `ps` command to gather detailed information about the identified process. This includes the command, arguments, and additional details. ```Command: ps aux | grep <PID>``` Replace `<PID>` with the actual process ID obtained from the `top` command. 3. **Terminate the Runaway Process:** - If the runaway process is causing issues and needs to be immediately stopped, use the `kill` command to terminate the process. ```Command: kill <PID> ``` Replace `<PID>` with the actual process ID. 4. **Monitor Resource Usage:** - After terminating the process, monitor resource usage using `top` again to ensure that the system resources are stabilizing. ```Command: top``` Check if CPU and memory usage return to normal levels.	Steps – 5 Marks Explanation – 2 Marks	7
VIII	To mount a disk on a Linux system, you'll need to follow these steps: 1. **Identify the Disk:** 2. **Create a Mount Point:** Choose or create a directory where you want to mount the disk. This directory is known as the mount point. For example, create a directory named `/mnt/mydisk`. ```Command: sudo mkdir /mnt/mydisk ``` 3. **Mount the Disk:** Use the `mount` command to attach the disk to the chosen mount point. Specify the disk device and the mount point as arguments. ```Command: sudo mount /dev/sdb1 /mnt/mydisk ```	Steps – 5 Marks Explanation – 2 Marks	7

	Replace `/dev/sdb1` with the actual disk or partition you want to mount. 4. **Verify the Mount:** Confirm that the disk is mounted by checking the output of the `df` command. ```Command: df -h ``` Ensure that the mount point (`/mnt/mydisk` in this case) is listed with the correct disk space information.		
IX	The role of Linux log files in system administration is crucial for maintaining and managing a Linux system effectively. Log files serve as a record of system events, errors, and activities, playing a significant role in system monitoring and diagnostics. Here's a detailed description of their role: 1. **Recording System Events:** - Log files capture a wide range of system events, including startup and shutdown processes, hardware and software errors, user logins and logouts, and changes to system configurations. These records provide a comprehensive view of the system's activities over time. 2. **Troubleshooting and Diagnostics:** - Log files are an invaluable resource for troubleshooting issues and diagnosing problems within a Linux system. When an error occurs, administrators can refer to relevant log files to identify the cause, analyze error messages, and trace the sequence of events leading to the issue. 3. **Monitoring System Health:** - By regularly reviewing log files, administrators can monitor the overall health of the system. System logs contain information about resource usage, performance metrics, and potential bottlenecks, allowing administrators to proactively address issues before they escalate. 4. **Security Auditing:** - Log files play a critical role in security auditing and monitoring. They record authentication attempts, failed login attempts, and security-related events. Monitoring these logs helps detect unauthorized access, potential security threats, and suspicious activities, contributing to the overall security posture of the system. 5. **Resource Utilization:** - Log files provide insights into resource utilization, including CPU usage, memory consumption, and disk activity. Monitoring resource-related logs helps administrators identify performance bottlenecks, optimize resource allocation, and ensure efficient system operation. 6. **Application-Specific Logging:** - Many applications and services on a Linux system generate their own log files. These application-specific logs provide detailed information about the behavior and performance of individual software components. Analyzing these logs helps in fine-tuning applications and identifying issues specific to particular services. 7. **Historical Analysis:** - Log files serve as a historical record of system activities. By analyzing	Any 4 points- 4 Mark Explanation – 3 Marks	7

	logs over an extended period, administrators can identify patterns, trends, and recurring issues. Historical analysis contributes to long-term system optimization, planning for upgrades, and understanding the system's evolution. 8. **Automated Monitoring Tools Integration:** - System administrators often use automated monitoring tools that rely on log files to provide real-time alerts and notifications. These tools continuously analyze logs for predefined patterns or anomalies, allowing administrators to respond promptly to critical events. In summary, Linux log files are essential tools for system administrators, offering a wealth of information for monitoring, diagnostics, and maintenance. They provide a detailed account of system activities, assist in troubleshooting, contribute to security auditing, and support overall system health monitoring. Regular review and analysis of log files empower administrators to maintain a stable, secure, and efficiently performing Linux system.		
X	1. **Power-On:** - When the computer is powered on, the Basic Input/Output System (BIOS) or Unified Extensible Firmware Interface (UEFI) firmware, which is stored in non-volatile memory on the motherboard, becomes active. 2. **Firmware Initialization:** - The BIOS/UEFI firmware performs initialization tasks, including a Power-On Self-Test (POST) to check hardware components like RAM, CPU, and storage devices for proper functionality. 3. **Boot Device Selection:** - The firmware locates and selects a bootable device, typically a hard drive, solid-state drive, or other storage media, based on the boot order configured in the system's firmware settings. 4. **Master Boot Record (MBR) or EFI:** - For BIOS systems, the firmware reads the Master Boot Record (MBR) from the boot device. For UEFI systems, it reads the EFI System Partition (ESP) and executes the bootloader specified in the UEFI variables. 5. **Bootloader Execution:** - The bootloader (e.g., GRUB) is loaded into memory and takes control of the boot process. The bootloader presents a menu to the user (if configured) and allows the selection of the operating system to load. 6. **Loading the Kernel:** - The selected bootloader loads the operating system kernel into memory. The kernel is the core component of the operating system and contains essential functions for managing hardware, processes, and system resources. 7. **Initial Ramdisk (initrd):** - If configured, an initial ramdisk (initrd) or initramfs may be loaded. This temporary file system provides essential drivers and modules needed to mount the root file system. 8. **Root File System Mounting:** - The kernel mounts the root file system specified in its configuration. The root file system contains the core operating system files and directories.	Steps – 5 Marks Explanation – 2 Marks	7

	9. **Init Process:** - The kernel executes the init process, which is the first user-space process. On modern systems using systemd, this could be `systemd` itself. 10. **User-Space Initialization:** - The init process initializes user-space components, starts system services, and launches other essential processes, transitioning the system from the kernel space to the user space. 11. **Graphical User Interface (GUI) or Command-Line Interface (CLI):** - Depending on the system configuration, the init process may launch a graphical user interface (GUI) or a command-line interface (CLI), providing a user-friendly environment for interaction.		
XI	To implement the configuration of vsftpd to allow anonymous FTP access, follow these steps: 1. **Open the vsftpd configuration file:** - Use a text editor to open the vsftpd configuration file. The typical location is `/etc/vsftpd.conf`. ```Command: sudo nano /etc/vsftpd.conf ``` 2. **Enable Anonymous FTP:** - Find the line that begins with `anonymous_enable` in the configuration file. By default, it is usually set to `NO`. - Change it to: ```conf anonymous_enable=YES ``` 3. **Set the Anonymous FTP Root Directory (Optional):** - If you want to specify a particular directory for anonymous FTP, you can use the `anon_root` option. Add or modify the line: ```conf anon_root=/path/to/your/directory ``` Replace `/path/to/your/directory` with the actual path you want to set as the root for anonymous FTP. If not set, anonymous users will typically be restricted to the system's default FTP directory. 4. **Save and Close the Configuration File:** - Save the changes you made in the configuration file. 5. **Restart vsftpd:** - To apply the changes, restart the vsftpd service. ```Command: sudo systemctl restart vsftpd ``` If your system doesn't use systemd, you might use `service`: ```Command: sudo service vsftpd restart ```	Steps – 5 Marks Explanation – 2 Marks	7
XII	To implement the installation of the Apache web server on a Linux system, you can use the appropriate package manager for your Linux distribution. Here are the steps for two commonly used package managers, `apt` (used by Debian-based systems like Ubuntu) and `yum` (used by Red Hat-based systems like CentOS and Fedora): ### Using `apt` (Debian/Ubuntu): 1. Open a terminal on your Debian/Ubuntu system.	Steps – 5 Marks Explanation – 2 Marks	7

	2. Update the package list to ensure you have the latest information about available packages: ```Command: sudo apt update ``` 3. Install the Apache web server using the following command: ```Command: sudo apt-get -y install apache2 ``` 4. During the installation, you may be prompted to confirm the action. Type 'Y' and press Enter to proceed. 5. Once the installation is complete, Apache will be automatically started. You can verify its status: ```Command: sudo systemctl status apache2 ``` If it's running, you should see an "active (running)" message. ### Using `yum` (CentOS/Fedora): 1. Open a terminal on your CentOS/Fedora system. 2. Update the package list: ```Command: sudo yum update ``` 3. Install the Apache web server: ```Command: sudo yum install httpd ``` 4. Confirm the installation when prompted by typing 'y' and pressing Enter. 5. Start the Apache service: ```Command: systemctl start httpd.service ``` 6. Check the status to ensure Apache is running: ```Command: sudo systemctl status httpd ``` Verify that the status is "active (running)." 7. Optionally, enable Apache to start on boot: ```Command: sudo systemctl enable httpd ``` Now, Apache is installed and running on your Linux system. You can access the default Apache page by opening a web browser and navigating to `http://localhost` or `http://your-server-IP`.		
XIII	NFS (Network File System) is widely used in network environments for various scenarios due to its ability to facilitate remote file access and sharing. Here are common scenarios, applications, and benefits of NFS: 1. **Centralized File Storage:** - *Scenario:* NFS is often used to create a centralized file storage system. Multiple clients can access and share files stored on a central NFS server. - *Application:* This is particularly useful in environments where a single, centralized repository of data is required, such as in enterprise settings with shared documents, user home directories, or software repositories. - *Benefits:* Simplifies data management, ensures consistency across multiple systems, and reduces the need for redundant local storage. 2. **Diskless Workstations:** - *Scenario:* NFS allows diskless workstations to boot and operate	Any 4 points – 1 mark each Explanation – 3 Marks	7

	without local storage. The necessary operating system and application files are provided by the NFS server. - *Application:* Diskless workstations are common in environments where maintenance and updates are streamlined, and a consistent operating environment is required for multiple systems. - *Benefits:* Simplifies workstation management, reduces maintenance efforts, and ensures uniformity in the software environment. 3. **Data Sharing Among Servers:** - *Scenario:* NFS enables efficient sharing of data among servers in a network. Multiple servers can mount and access common directories or files hosted on an NFS server. - *Application:* This is valuable in scenarios where multiple servers need access to shared data, such as database servers accessing common data files or web servers sharing static content. - *Benefits:* Facilitates collaboration and coordination between servers, avoids data duplication, and ensures consistency in shared resources. 4. **Backup and Disaster Recovery:** - *Scenario:* NFS is used for storing backups on a central server, allowing multiple clients to back up and restore data easily. - *Application:* Backup servers can access and store data from various clients on a centralized NFS server, simplifying backup and recovery processes. - *Benefits:* Streamlines backup procedures, centralizes data for recovery, and ensures data integrity across backup systems. 5. **Virtualization Environments:** - *Scenario:* NFS is commonly used in virtualized environments, where virtual machines (VMs) store and share their virtual disks on NFS-mounted directories. - *Application:* Virtualization platforms, such as VMware or KVM, can leverage NFS to store VM images and share them across multiple hosts. - *Benefits:* Simplifies VM storage management, allows for dynamic resource allocation, and supports features like live migration of VMs between hosts. 6. **Cross-Platform File Sharing:** - *Scenario:* NFS provides a platform-independent file-sharing solution, allowing systems running different operating systems to access shared data. - *Application:* NFS can be used in heterogeneous environments, enabling Linux, Unix, and Windows systems to share and access files seamlessly. - *Benefits:* Promotes interoperability, facilitates collaboration in mixed-platform environments, and reduces compatibility issues. Overall, NFS offers a flexible and scalable solution for network file sharing, providing efficiency, centralized management, and seamless collaboration in various computing environments.		

XIV	Dynamic Host Configuration Protocol (DHCP) is a network protocol that automates the process of assigning IP addresses and other network configuration parameters to devices in a network. Here is an explanation of how DHCP works and the process by which a device obtains an IP address from a DHCP server: 1. **DHCP Discovery:** - When a device, such as a computer or a network-enabled device, connects to a network, it begins the DHCP process. The device sends a DHCP Discover message to the network. This broadcast message is a request for a DHCP server to provide network configuration information. 2. **DHCP Offer:** - DHCP servers on the network receive the DHCP Discover message and respond with a DHCP Offer. In this Offer, the DHCP server proposes an IP address, subnet mask, and other configuration details to the requesting device. Multiple DHCP servers on the network may respond, but the device typically accepts the first Offer it receives. 3. **DHCP Request:** - Upon receiving DHCP Offers, the device selects one and sends a DHCP Request message. This message confirms the chosen DHCP server's offer and informs other servers that their offers were not accepted. The Request also includes the device's unique identifier, known as the Client Identifier. 4. **DHCP Acknowledgment:** - The chosen DHCP server responds to the DHCP Request with a DHCP Acknowledgment (ACK). This message finalizes the configuration process. It includes the approved IP address, subnet mask, default gateway, DNS server information, lease duration, and other network settings. The device is now configured for communication on the network. 5. **Lease Duration:** - The DHCP server assigns an IP address to the device for a specific lease duration. This is the amount of time the device is allowed to use the assigned IP address. Before the lease expires, the device can choose to renew the lease, ensuring continued network connectivity. 6. **DHCP Renewal:** - As the lease duration approaches expiration, the device may attempt to renew its IP address lease. It sends a DHCP Request to the DHCP server that originally provided the lease. If the DHCP server is available and the lease is still valid, it responds with a DHCP Acknowledgement, renewing the lease for the device. 7. **DHCP Release:** - When a device disconnects from the network or no longer requires an IP address, it can send a DHCP Release message to inform the DHCP server that it is releasing the assigned IP address. This allows the DHCP server to reclaim and reuse the IP address. In summary, DHCP works through a sequence of Discover, Offer, Request, and Acknowledgement messages. The DHCP server dynamically	Steps – 5 Marks Explanation – 2 Marks	7

	assigns IP addresses and provides essential network configuration details to devices, streamlining the process of connecting to and communicating on a network.		