**1. How would you create a new user account in Linux?**

The adduser command streamlines the creation of new user accounts by handling the necessary system configurations, including creating a home directory and setting up default user settings. This promotes consistent and secure user management.

**sudo adduser username**

* **sudo**: This command grants elevated privileges (root access) to the command that follows. Root access is required to modify system files and create new user accounts.
* **adduser**: This is a user-friendly command-line utility for creating new user accounts in Linux. It automates the process of creating the user's home directory, setting up initial files, and prompting for a password.
* **username**: This is the desired name for the new user account. Replace "username" with the actual username you want to create.

**2. Explain the difference between relative and absolute paths**.

* **Absolute:** Full path from the root (e.g., /home/user/file.txt). Always the same location.
* **Relative:** Path from your current location (e.g., documents/file.txt). Location changes with your current directory

**3. What command would you use to change file ownership?**

File ownership is crucial for security. The owner has special privileges. You might use chown when you create a file as root and want to give it to your regular user.

* **Examples:**
  + sudo chown myuser mydocument.txt (Makes "myuser" the owner of "mydocument.txt")
  + sudo chown :mygroup myfolder (Changes the group ownership of "myfolder" to "mygroup")
  + sudo chown -R myuser:mygroup mydirectory (Recursively changes ownership of a directory and all its contents)

**4. How can you schedule a task to run automatically at a specific time?**

Cron jobs are perfect for automating repetitive tasks, like backups or system checks.

* **Examples:**
  + 0 0 \* \* 0 /home/user/weekly\_backup.sh (Runs "weekly\_backup.sh" at midnight every Sunday)
  + \*/5 \* \* \* \* /home/user/check\_status.sh (Runs "check\_status.sh" every 5 minutes)
  + crontab -l (Lists your current cron jobs)

**5. Describe how you would compress and decompress files in Linux.**

tar is great for bundling files together, which makes them easier to move or back up. The gzip or xz compression options make the archives smaller.

* **Examples:**
  + tar -cvf myfiles.tar documents/ images/ (Creates an uncompressed archive of "documents" and "images" folders)
  + tar -czvf myfiles.tar.gz myfolder/ (Creates a compressed archive of "myfolder")
  + tar -xvf myfiles.tar -C /tmp (Extracts "myfiles.tar" to the "/tmp" directory)

**6. What's the purpose of the 'grep' command and how would you use it?**

* **Explanation:** grep is incredibly powerful for searching through large files or outputs.
* **Examples:**
  + grep -i "case" myfile.txt (Finds lines containing "case", ignoring case)
  + grep -r "word" /path/to/directory (Recursively searches for "word" in all files within the directory)
  + command | grep "filter" (Filters the output of "command" for lines containing "filter")

**7.** **How do you check and manage running processes in Linux?**

* **Explanation:** Understanding how to manage processes is essential for system stability.
* **Examples:**
  + ps aux | grep firefox (Finds the process ID of Firefox)
  + kill -9 1234 (Forcefully kills process ID 1234)
  + pkill firefox (Kills all processes named firefox)

**8. How do you create and manage symbolic links? (ln -s)**

* **Explanation:** Symbolic links are like shortcuts. They're useful for creating aliases or making files appear in multiple locations.
* **Examples:**
  + ln -s /var/www/html/website /home/user/mywebsite (Creates a shortcut named "mywebsite" in your home directory)
  + ln -s /usr/bin/python3 python (Creates a shortcut named "python" to the Python 3 executable)

**9.** **Describe the process of mounting and unmounting file systems. (mount, umount)**

* **Explanation:** Mounting makes storage devices accessible. Unmounting safely detaches them.
* **Examples:**
  + sudo mount /dev/sdb1 /media/usb (Mounts the first partition of the second drive to "/media/usb")
  + sudo mount -t vfat /dev/sdb1 /media/usb (Mounts a FAT32 formatted device)
  + sudo umount -l /media/usb (Lazy unmount, which is useful if the device is busy)

**10. How would you search for files larger than a specific size? (find)**

* **Explanation:** find is versatile for locating files based on various criteria.
* **Examples:**
  + find /home/user -name "\*.txt" (Finds all text files in your home directory)
  + find / -type d -name "config" (Finds directories named "config" anywhere on the system)
  + find . -mtime -7 (Finds files modified in the last 7 days)

**11.** **How do you check and modify file permissions recursively? (chmod)**

* **Explanation:** File permissions control who can read, write, and execute files.
* **Examples:**
  + chmod u+x script.sh (Adds execute permission for the owner)
  + chmod g-w myfile.txt (Removes write permission for the group)
  + chmod 644 myfile.txt (Sets read/write for owner, read-only for group and others)