**Cd -commands**

**Linux Commands**

1. Navigating to a Specific Directory

* *Scenario*: You are in the /home/spoorthi/documents directory. You want to go to the /home/spoorthi/projects/docker directory.
* **Question**: What cd command would you use to navigate directly to /home/spoorthi/projects/docker?

cd /home/spoorthi/projects/docker

2. Moving Up the Directory Tree

* *Scenario*: You are currently in /home/spoorthi/projects/docker/node-app. You need to navigate back to the /home/spoorthi directory.
* **Question**: What command would you use to move back two levels and reach /home/spoorthi?

cd ../..

3. Using Relative Paths

* *Scenario*: You are in the /home/spoorthi/projects/docker directory. You want to navigate to /home/spoorthi/projects/python.
* **Question**: How would you use a relative path to move from the docker directory to the python directory?

cd ../python

4. Returning to the Previous Directory

* *Scenario*: You are in /home/spoorthi/projects/python. Earlier, you were in /home/spoorthi/projects/docker. You want to switch back to the docker directory.
* **Question**: What command would you use to quickly return to the previous directory?

cd -

5. Navigating to the Home Directory

* *Scenario*: You are in /var/www/html. You now want to quickly navigate back to your home directory.
* **Question**: Which cd command would take you directly to your home directory from any location?

cd ~

6. Handling Spaces in Directory Names

* *Scenario*: You have a directory named My Projects under /home/spoorthi. You are currently in the /home/spoorthi/documents directory and want to navigate to /home/spoorthi/My Projects.
* **Question**: How would you write the cd command to handle spaces in the directory name?

[cd /home/spoorthi/My\ Projects ] - cd "/home/spoorthi/My Projects"

7. Navigating with Tilde (~) Shortcut

* *Scenario*: You want to move from the /tmp directory to a directory named reports located at /home/spoorthi/reports, but you want to use the tilde ~ shortcut.
* **Question**: How would you use cd with ~ to reach the reports directory from /tmp?

cd ~/reports

8. Navigating Using an Absolute Path

* *Scenario*: You are in the /home/spoorthi/documents/notes directory. You want to go directly to /etc/apache2.
* **Question**: What command would you use to navigate to /etc/apache2 using the absolute path?

cd /etc/apache2

9. Navigating Back to Root Directory

* *Scenario*: You are in the /home/spoorthi/projects/docker. You need to navigate to the root directory /.
* **Question**: Which cd command would take you to the root directory?

cd /

10. Navigating to Parent Directory

* *Scenario*: You are in /home/spoorthi/projects/python/automation-scripts and want to go back to the python directory.
* **Question**: What command would you use to move one level up to python?

cd ..

**LS – commands**

1: Listing Hidden Files   
You are in a directory, and you know there are hidden files, but the ls command does not show them. How can you list these hidden files along with normal files?

**Answer**: To display hidden files (files that start with a .), use the -a flag:

ls -a

2: Listing Files with Detailed Information  
You want to view more information about each file, such as file size, permissions, number of links, owner, group, and modification date. How can you do that?

**Answer**: Use the -l flag to display detailed information in long format:

ls -l

3: Sorting Files by Modification Time  
You need to list the files in a directory, but you want to sort them by modification time (most recent files first). How do you do this?

**Answer**: Use the -t flag to sort by modification time:

ls -lt

4: Listing Files with Human-Readable File Sizes

ls -ls  
You are using the ls -l command to see file details, but file sizes are in bytes. How can you show file sizes in human-readable formats (KB, MB, GB)?

**Answer**: Use the -h flag along with -l to show human-readable sizes:

ls -lh

5: Listing Files Recursively in Subdirectories  
You want to list all the files, not only in the current directory but also in all subdirectories. How can you achieve this?

**Answer**: Use the -R flag to list files recursively:

ls -R

*Scenario* 6: Sorting Files by File Size

**Question**: You want to list files sorted by their size, with the largest file appearing first. How would you do this?

**Answer**: Use the -S flag to sort files by size:

ls -lS

*Scenario* 7: Combining Options for Detailed and Sorted Listing

**Question**: You need to list all the files in a directory, including hidden files, in long format, sorted by modification time, and sizes should be in human-readable form. Which command will you use?

**Answer**: Combine multiple options:

ls -altlh

*Scenario* 8: Listing Only Directory Names

**Question**: You want to list only the directories (not files) in the current directory. What command would you use?

**Answer**: Use the -d and \*/ combination to list only directories:

ls -d \*/

*Scenario* 9: Displaying File Types

**Question**: You want to differentiate between files and directories in the output of ls. How can you do that?

**Answer**: Use the -F option, which appends a / to directories, an \* to executable files, and other symbols to specific file types:

ls -F

*Scenario* 10: Displaying Files in Reverse Order

**Question**: You need to list the files in a directory, but you want them displayed in reverse order. What option should you use?

**Answer**: Use the -r flag to reverse the order of the listing:

ls -lr

**df – commands**

*Scenario* 1: Checking Disk Usage in Human-Readable Format

**Question**: You want to check the disk usage on your system, but the sizes are displayed in blocks. How can you display the disk usage in a human-readable format (e.g., MB, GB)?

**Answer**: Use the -h flag to display sizes in a human-readable format:

df -h

*Scenario* 2: Displaying Disk Usage for a Specific Filesystem

**Question**: You are troubleshooting a full filesystem and need to check the disk usage only for the /home filesystem. What command will you use?

**Answer**: Specify the filesystem or mount point as an argument:

df -h /home

*Scenario* 3: Checking Disk Usage Including File System Types

**Question**: You need to list all the filesystems along with their types (e.g., ext4, xfs, etc.). How can you display this information?

**Answer**: Use the -T option to show the filesystem type:

df -T

*Scenario* 4: Checking Disk Usage in Inodes

**Question**: A system is running out of inodes rather than disk space. How do you check the inode usage on the system?

**Answer**: Use the -i flag to display inode information:

df -i

*Scenario* 5: Checking Disk Usage Without Including Specific Filesystems

**Question**: You want to check the disk usage but want to exclude certain filesystems like tmpfs or devtmpfs. How can you exclude these filesystems from the output?

**Answer**: Use the -x flag to exclude a specific filesystem type. For example, to exclude tmpfs:

df -h -x tmpfs

*Scenario* 6: Displaying Disk Usage for All Filesystems, Including Dummy or Special Filesystems

**Question**: You suspect that some special filesystems, like tmpfs or proc, are using disk space. How can you include those in the output of df?

**Answer**: Use the -a option to display all filesystems, including dummy and special ones:

df -a

*Scenario* 7: Monitoring Disk Usage for NFS or Remote Filesystems

**Question**: You want to see the disk usage for network filesystems (e.g., NFS or CIFS) but exclude local filesystems. How can you display only network-mounted filesystems?

**Answer**: Use the -t option and specify the network filesystem type (e.g., nfs):

df -h -t nfs

*Scenario* 8: Displaying Disk Usage in 1K, 1M, or 1G Blocks

**Question**: You want to view the disk usage in specific block sizes such as 1K blocks, 1M blocks, or 1G blocks. How can you set the output to use a specific block size?

**Answer**: You can set the block size using the -B option:

* For 1K blocks:

df -B 1K

* For 1M blocks:

df -B 1M

* For 1G blocks:

df -B 1G

*Scenario* 9: Showing Disk Usage for All Partitions in Human-Readable Form with File System Types

**Question**: You need a detailed disk usage report that includes the partition size, used space, available space, and filesystem types in human-readable form. What command will you use?

**Answer**: Combine the -h and -T options to show human-readable sizes with filesystem types:

df -hT

*Scenario* 10: Displaying Usage Statistics with No Header

**Question**: You want to display disk usage statistics but remove the column headers from the output for use in a script. How can you achieve this?

**Answer**: Use the --no-sync option to skip syncing and --output to format specific columns without headers:

df --output=source,size,used,avail

**dU – commands**

*Scenario* 1: Checking the Size of a Directory in Human-Readable Format

**Question**: You want to find out how much disk space a specific directory, such as /var/log, is using. How can you display the directory size in a human-readable format?

**Answer**: Use the -h option to display the size in human-readable form:

du -h /var/log

*Scenario* 2: Finding the Size of a Directory and Its Subdirectories

**Question**: You want to see the size of the /home/user directory, including the sizes of all its subdirectories. How can you do that?

**Answer**: Run the du command without any additional options to see sizes for the directory and its subdirectories:

du /home/user

*Scenario* 3: Displaying Only the Total Size of a Directory

**Question**: You are only interested in the total size of the /usr/local directory, without listing sizes of individual subdirectories. How can you achieve this?

**Answer**: Use the -s (summarize) option to display only the total size:

du -sh /usr/local

*Scenario* 4: Sorting Files and Directories by Size

**Question**: You need to find the largest directories and files under /var. How can you list the directories and files sorted by their size?

**Answer**: You can use the du command along with the sort command:

du -h /var | sort -hr

*Scenario* 5: Displaying the Disk Usage of Files in a Directory

**Question**: You want to see the size of all the files in a directory, not just the directories themselves. How can you list the size of files as well?

**Answer**: Use the -a option to display the disk usage of all files:

du -ah /path/to/directory

*Scenario* 6: Finding Files or Directories Larger than a Certain Size

**Question**: You are troubleshooting disk space usage and want to find files or directories larger than 100MB in the /home directory. How would you do this?

**Answer**: Use du with the find command to locate files and directories above a specific size:

du -ah /home | grep '[0-9.]\*M'

Alternatively, you can use find directly for a more targeted search:

find /home -size +100M

*Scenario* 7: Excluding Specific Directories from Disk Usage Calculation

**Question**: You are checking disk usage for the /var directory but want to exclude the /var/log subdirectory from the calculation. What command would you use?

**Answer**: Use the --exclude option to exclude a specific directory:

du -h --exclude=/var/log /var

*Scenario* 8: Displaying Disk Usage for Multiple Directories

**Question**: You want to check the disk usage of several directories, such as /etc, /usr, and /opt, in one command. How can you achieve this?

**Answer**: List the directories separated by spaces:

du -h /etc /usr /opt

*Scenario* 9: Monitoring Disk Usage Growth Over Time

**Question**: You want to monitor how the disk usage of a directory changes over time. How can you check its usage every minute?

**Answer**: Use the watch command with du to monitor disk usage in real-time:

watch -n 60 du -sh /path/to/directory

*Scenario* 10: Displaying Disk Usage in 1K, 1M, or 1G Blocks

**Question**: You want to see the disk usage of files and directories, but the sizes should be displayed in specific block sizes, such as 1K blocks, 1M blocks, or 1G blocks. How can you do this?

**Answer**: Use the -B option to specify the block size:

* For 1K blocks:

du -BK /path/to/directory

* For 1M blocks:

du -BM /path/to/directory

* For 1G blocks:

du -BG /path/to/directory

*Scenario* 11: Estimating the Space Taken by File System Links (Symbolic Links)

**Question**: You are troubleshooting a directory with symbolic links and want to include them in the disk usage calculation. How can you include the size of files pointed to by symbolic links?

**Answer**: Use the --dereference option to include the sizes of files pointed to by symbolic links:

du -h --dereference /path/to/directory

**FIND – commands**

Here are *Scenario*-based **Question**s for the find command, along with explanations:

*Scenario* 1: Finding Files by Name

**Question**: You are looking for a specific file named error.log in the /var/log directory. How can you find this file?

**Answer**: Use the -name option to search for files by name:

find /var/log -name "error.log"

*Scenario* 2: Finding Files Modified in the Last 7 Days

**Question**: You need to find all files in the /home/user directory that were modified in the last 7 days. How can you achieve this?

**Answer**: Use the -mtime option to search for files based on modification time:

find /home/user -mtime -7

*Scenario* 3: Finding Files Larger Than 100MB

**Question**: You suspect large files are taking up too much space in the /var directory. How can you find all files larger than 100MB?

**Answer**: Use the -size option to search for files larger than a specific size:

find /var -size +100M

*Scenario* 4: Finding Empty Files and Directories

**Question**: You want to clean up empty files and directories in the /tmp directory. How can you find all empty files and directories?

**Answer**: Use the -empty option to find empty files or directories:

find /tmp -empty

*Scenario* 5: Finding Files by File Type

**Question**: You are troubleshooting and need to list all directories under /etc. How can you find only directories?

**Answer**: Use the -type d option to search for directories:

find /etc -type d

*Scenario* 6: Finding Files by Permissions

**Question**: You want to find files in /usr/bin that have 755 permissions. What command will you use?

**Answer**: Use the -perm option to search for files by permission:

find /usr/bin -perm 755

*Scenario* 7: Finding Files and Deleting Them

**Question**: You need to clean up old temporary files in /tmp that are older than 30 days. How can you find and delete these files in one command?

**Answer**: Use the -mtime option along with -delete to find and delete the files:

find /tmp -mtime +30 -delete

Note: Use the -delete option with caution as it permanently deletes files.

*Scenario* 8: Finding Files and Executing a Command

**Question**: You want to change the ownership of all files in /var/www to www-data. How can you find all files in this directory and change their ownership?

**Answer**: Use the -exec option to run a command on the found files:

find /var/www -type f -exec chown www-data {} \;

*Scenario* 9: Excluding Specific Directories

**Question**: You are searching for files in the /home directory but want to exclude the /home/user/temp subdirectory from the search. How can you do this?

**Answer**: Use the -path and -prune options to exclude specific directories:

find /home -path /home/user/temp -prune -o -type f -print

*Scenario* 10: Finding Recently Accessed Files

**Question**: You need to find files that have been accessed in the last 2 days in the /var/log directory. What command would you use?

**Answer**: Use the -atime option to find files based on access time:

find /var/log -atime -2

*Scenario* 11: Finding Symbolic Links

**Question**: You are auditing symbolic links on your system and want to list all symbolic links under /usr/local. How can you find them?

**Answer**: Use the -type l option to search for symbolic links:

find /usr/local -type l

*Scenario* 12: Finding Files Based on Multiple Criteria

**Question**: You need to find all files in /home/user that are larger than 10MB and were modified in the last 5 days. How can you combine these search criteria?

**Answer**: Use multiple conditions to combine the criteria:

find /home/user -size +10M -mtime -5

*Scenario* 13: Finding Files Owned by a Specific User

**Question**: You want to list all files in /var that are owned by the user john. What command will you use?

**Answer**: Use the -user option to search for files owned by a specific user:

find /var -user john

*Scenario* 14: Finding Files and Printing Their Full Path

**Question**: You need to list all files in the /etc directory and display their full paths. How can you print the full path of each file?

**Answer**: Use the -print option to display the full path:

find /etc -type f -print

*Scenario* 15: Searching for Files with Case-Insensitive Names

**Question**: You want to find a file called README.txt, but you're not sure if the file name is in uppercase, lowercase, or a mix of both. How can you search for the file case-insensitively?

**Answer**: Use the -iname option for a case-insensitive name search:

find /path/to/directory -iname "readme.txt"

*Scenario* 16: Finding Files Not Owned by a Specific User

**Question**: You want to find all files in /home/user that are not owned by the user john. How would you do this?

**Answer**: Use the -not option to invert the search condition:

find /home/user -not -user john

*Scenario* 17: Finding Files Based on File Depth

**Question**: You want to search for files in /var but only want to search one directory level deep (ignoring deeper subdirectories). How can you limit the depth of the search?

**Answer**: Use the -maxdepth option to limit the search depth:

find /var -maxdepth 1 -type f

*Scenario* 18: Finding Files with a Specific Extension

**Question**: You want to list all .log files under /var/log. What command would you use to find all files with this extension?

**Answer**: Use the -name option with a wildcard:

find /var/log -name "\*.log"

**Diff -commands**

*Scenario* 1: Comparing Two Text Files

**Question**: You have two text files, file1.txt and file2.txt, and you want to see the differences between them. What command will you use?

**Answer**: Use the diff command to compare the two files:

diff file1.txt file2.txt

*Scenario* 2: Ignoring Case Differences

**Question**: You want to compare two files, fileA.txt and fileB.txt, but you want to ignore case differences in your comparison. How can you do this?

**Answer**: Use the -i option to ignore case:

diff -i fileA.txt fileB.txt

*Scenario* 3: Comparing Directories

**Question**: You need to compare the contents of two directories, dir1 and dir2, to find any differences in files. What command should you use?

**Answer**: Use the -r option to compare directories recursively:

diff -r dir1 dir2

*Scenario* 4: Displaying Differences Side by Side

**Question**: You want to compare two files, source.txt and target.txt, and display their differences side by side for easier comparison. What command will you use?

**Answer**: Use the -y option to display differences side by side:

diff -y source.txt target.txt

*Scenario* 5: Suppressing White Space Differences

**Question**: When comparing file1.txt and file2.txt, you want to ignore differences in whitespace. How can you achieve this?

**Answer**: Use the -w option to ignore all white space:

diff -w file1.txt file2.txt

*Scenario* 6: Generating a Unified Diff

**Question**: You are working with patches and need a unified diff format of the differences between old.txt and new.txt. How can you generate this format?

**Answer**: Use the -u option to create a unified diff:

diff -u old.txt new.txt

*Scenario* 7: Checking Differences in Binary Files

**Question**: You have two binary files, image1.png and image2.png, and you want to check if they are different. How can you compare these binary files?

**Answer**: Use the --binary option to compare binary files:

diff --binary image1.png image2.png

*Scenario* 8: Displaying Only the Differences

**Question**: You want to see only the lines that are different between file1.txt and file2.txt, without the context. What command should you use?

**Answer**: Use the -q option to report only whether the files differ:

diff -q file1.txt file2.txt

*Scenario* 9: Creating a Patch File

**Question**: You want to create a patch file from the differences between original.txt and modified.txt. How can you generate a patch file?

**Answer**: Use the -u option to create a unified diff and redirect the output to a patch file:

diff -u original.txt modified.txt > changes.patch

*Scenario* 10: Comparing Files with Specific Line Numbers

**Question**: You need to compare two files but only want to see the differences starting from line 10. How can you achieve this?

**Answer**: Use the -n option to specify the starting line (though diff does not have a direct option for this, you can use sed to preprocess the files):

diff <(sed '1,9d' file1.txt) <(sed '1,9d' file2.txt)

*Scenario* 11: Comparing Files with Different Line Endings

**Question**: You need to compare two text files, file1.txt (created on Unix) and file2.txt (created on Windows), which may have different line endings. What command can you use?

**Answer**: Use the -b option to ignore changes in the amount of white space (which may include line endings):

diff -b file1.txt file2.txt

*Scenario* 12: Viewing Only the Differences in Specific Files

**Question**: You want to compare only the .txt files in two directories, dir1 and dir2. How can you list only the differences for those files?

**Answer**: Use the find command in combination with diff to limit to .txt files:

diff -r <(find dir1 -name "\*.txt" -exec cat {} \;) <(find dir2 -name "\*.txt" -exec cat {} \;)

*Scenario* 13: Using Context Lines

**Question**: You want to see the differences between file1.txt and file2.txt, but you also want to include a few lines of context around the changes. What command will you use?

**Answer**: Use the -C option followed by the number of context lines you want (e.g., 3 lines of context):

diff -C 3 file1.txt file2.txt

*Scenario* 14: Displaying Differences with Line Numbers

**Question**: You want to compare two files and see the differences along with the corresponding line numbers. How can you do this?

**Answer**: Use the --unified option with -c to see context along with line numbers:

diff -c file1.txt file2.txt

*Scenario* 15: Using Diff with Git

**Question**: You are using Git and want to see the differences between the working directory and the last commit for example.txt. What command should you use?

**Answer**: Use the git diff command:

git diff example.txt.

**locate Commands**

*Scenario* 1: Finding a File by Name

**Question**: You want to quickly find a file named report.pdf anywhere on your system. How can you do this using the locate command?

**Answer**: Use the locate command followed by the file name:

locate report.pdf

*Scenario* 2: Searching for Files with a Specific Extension

**Question**: You need to find all .txt files on your system. What command will you use to locate these files?

**Answer**: Use a wildcard character to search for files with a specific extension:

locate "\*.txt"

*Scenario* 3: Finding Files in a Specific Directory

**Question**: You want to find all files named config.yaml located specifically in the /etc directory. How can you limit your search to this directory?

**Answer**: You can use the -r option with a regex pattern to limit the search:

locate -r "^/etc/config.yaml$"

*Scenario* 4: Updating the Database

**Question**: You just created several new files and want to ensure that the locate command can find them. How can you update the database?

**Answer**:Run the following command to update the locate database:

sudo updatedb

*Scenario* 5: Finding Files with Similar Names

**Question**: You are looking for files that have "report" in their names but are unsure of the full name. How can you locate all files that contain "report"?

**Answer**: Use the locate command with the search term:

locate report

*Scenario* 6: Searching for Files Based on Case Sensitivity

**Question**: You need to find files named README.md but are unsure if it might be in uppercase or lowercase. How can you perform a case-insensitive search?

**Answer**: By default, locate is case-sensitive, but you can use the -i option for case-insensitive search:

locate -i readme.md

*Scenario* 7: Limiting the Number of Results

**Question**: You want to find all files with "log" in their names but only want to see the first 10 results. How can you limit the output?

**Answer**: You can pipe the output to head to limit the number of results:

locate log | head -n 10

*Scenario* 8: Searching for Files Modified Recently

**Question**: You want to find files that were modified recently but are unsure of their names. How can you locate files based on modification time using locate?

**Answer**: While locate does not directly provide options for modification time, you can use the find command instead:

find / -type f -mtime -7

However, if you want to continue using locate, you'll need to update your database and filter results based on modification dates manually afterward.

*Scenario* 9: Searching for Directories

**Question**: You need to find all directories named backup in your file system. What command should you use?

**Answer**: Use the locate command to search for directories specifically:

locate /backup

*Scenario* 10: Searching in a Specific Path

**Question**: You want to find all files with "image" in their names within the /home/user/pictures directory. How can you perform this search?

**Answer**: Limit the search to the specified directory:

locate /home/user/pictures/image

*Scenario* 11: Excluding Certain Patterns

**Question**:  
You want to find all files containing "doc" in their names but want to exclude any results that contain "draft". How can you accomplish this?

**Answer**:  
You can use grep to exclude results:

locate doc | grep -v draft

*Scenario* 12: Finding Files with Special Characters

**Question**:  
You need to find files that contain special characters, such as spaces or hyphens, in their names. How can you locate such files?

**Answer**: You can quote the filename or use backslashes to escape special characters:

locate "my file-name.txt"

*Scenario* 13: Searching for Recently Created Files

**Question**: You want to find files created in the last few days, but the locate command does not track creation time. What is a better command to use?

**Answer**: Use the find command with -ctime to find files created in the last few days:

find / -type f -ctime -3

*Scenario* 14: Viewing the Location of Files

**Question**: You want to know the full path of all files that have "config" in their name. How can you do this?

**Answer**: Simply use locate to display the full paths:

locate config

*Scenario* 15: Searching with Multiple Patterns

**Question**: You want to find files that either contain "data" or "report" in their names. How can you perform this search?

**Answer**:Use grep to filter results from locate:

locate | grep -E "data|report"

**Copy commands**

1. Copying a Single File

*Scenario*: You have a file named report.txt in your current directory, and you want to make a copy of it called report\_backup.txt.  
Question: What command would you use?  
Answer:

cp report.txt report\_backup.txt

2. Copying a File to Another Directory

*Scenario*: You are in the /home/spoorthi/documents directory, and you want to copy a file named data.csv to the /home/spoorthi/backup directory.  
Question: What command would you use?

cp data.csv /home/spoorthi/backup/

3. Copying Multiple Files

*Scenario*: You want to copy multiple files, file1.txt, file2.txt, and file3.txt, from the /home/spoorthi/downloads directory to the /home/spoorthi/documents directory.  
Question: What command would you use?  
cp /home/spoorthi/downloads/file1.txt /home/spoorthi/downloads/file2.txt /home/spoorthi/downloads/file3.txt /home/spoorthi/documents/

or

cp /home/spoorthi/downloads/file{1,2,3}.txt /home/spoorthi/documents/

4. Copying a Directory Recursively

*Scenario*: You have a directory named projects that you want to copy to a new location called projects\_backup.  
Question: What command would you use to copy the entire directory and its contents?

cp -r projects/ projects\_backup/

5. Overwriting Files with Confirmation

*Scenario*: You want to copy a file named config.json from the current directory to the /etc/myapp/ directory, but you want to be prompted before overwriting any existing files with the same name.  
Question: What command would you use?  
cp -i config.json /etc/myapp/

6. Preserving File Attributes

*Scenario*: You want to copy a file named script.sh to a backup directory and also preserve the file's original attributes (such as timestamps and permissions).  
Question: What command would you use?

cp -p script.sh /home/spoorthi/backup/

7. Using Wildcards to Copy Files

*Scenario*: You want to copy all .jpg files from the /home/spoorthi/pictures directory to the /home/spoorthi/backup directory.  
Question: What command would you use?  
cp /home/spoorthi/pictures/\*.jpg /home/spoorthi/backup/

8. Copying Files and Renaming Them

*Scenario*: You want to copy a file named summary.docx from the /home/spoorthi/reports directory and rename it to summary\_backup.docx in the same directory.  
Question: What command would you use?  
cp /home/spoorthi/reports/summary.docx /home/spoorthi/reports/summary\_backup.docx

9. Verbose Output During Copying

*Scenario*: You are copying files and want to see detailed output about the files being copied.  
Question: What command would you use?  
cp -v source\_file.txt destination\_file.txt

or for multiple files:

cp -v \*.txt /destination\_directory/

10. Copying Files to a Non-Existent Directory

*Scenario*: You want to copy a file named archive.zip to a directory called backup, but the directory does not exist yet.  
Question: What will happen if you run the command without creating the directory first?  
Answer:

cp archive.zip backup/

It will result in an error message indicating that the destination directory does not exist. You need to create the directory first:

mkdir backup

cp archive.zip backup/

**Move Commands**

1. Moving a Single File

*Scenario*: You have a file named report.txt in your current directory, and you want to move it to the /home/spoorthi/documents directory.  
Question: What command would you use?

mv report.txt /home/spoorthi/documents/

2. Renaming a File

*Scenario*: You want to rename a file named old\_name.txt to new\_name.txt in the current directory.  
Question: What command would you use?

mv old\_name.txt new\_name.txt

3. Moving Multiple Files

*Scenario*: You want to move multiple files, file1.txt, file2.txt, and file3.txt, from the /home/spoorthi/downloads directory to the /home/spoorthi/documents directory.  
Question: What command would you use?

mv /home/spoorthi/downloads/file1.txt /home/spoorthi/downloads/file2.txt /home/spoorthi/downloads/file3.txt /home/spoorthi/documents/

or using wildcards:

mv /home/spoorthi/downloads/file{1,2,3}.txt /home/spoorthi/documents/

4. Moving a Directory

*Scenario*: You have a directory named projects that you want to move to the /home/spoorthi/backup directory.  
Question: What command would you use to move the entire directory?

mv projects/ /home/spoorthi/backup/

5. Moving Files with Confirmation

*Scenario*: You want to move a file named config.json from the current directory to the /etc/myapp/ directory, but you want to be prompted before overwriting any existing files with the same name.  
Question: What command would you use?

mv -i config.json /etc/myapp/

6. Moving Files Using Wildcards

*Scenario*: You want to move all .jpg files from the /home/spoorthi/pictures directory to the /home/spoorthi/backup directory.  
Question: What command would you use?

mv /home/spoorthi/pictures/\*.jpg /home/spoorthi/backup/

7. Moving Files Over SSH

*Scenario*: You want to move a file named data.zip from your local machine to a remote server at user@remote\_host in the /home/user/ directory.  
Question: What command would you use?

scp data.zip user@remote\_host:/home/user/

8. Moving Files and Renaming Them

*Scenario*: You want to move a file named summary.docx from the /home/spoorthi/reports directory and rename it to final\_summary.docx in the same directory.  
Question: What command would you use?

mv /home/spoorthi/reports/summary.docx /home/spoorthi/reports/final\_summary.docx

9. Verbose Output During Moving

*Scenario*: You are moving files and want to see detailed output about the files being moved.  
Question: What command would you use?

mv -v source\_file.txt /destination\_directory/

or for multiple files:

mv -v \*.txt /destination\_directory/

10. Handling Spaces in Filenames

*Scenario*: You have a file named My Report.txt in your current directory, and you want to move it to the /home/spoorthi/documents directory.  
Question: How would you write the mv command to handle spaces in the filename?

mv "My Report.txt" /home/spoorthi/documents/

or using a backslash to escape the space:

mv My\ Report.txt /home/spoorthi/documents/

11. Moving Hidden Files

*Scenario*: You want to move all hidden files (files starting with a dot) from your home directory to a backup directory.  
Question: What command would you use?

mv ~/.??\* /home/spoorthi/backup/

12. Moving Files Based on Extension

*Scenario*: You want to move all .txt files from the /home/spoorthi/documents directory to the /home/spoorthi/backup directory.  
Question: What command would you use?

mv /home/spoorthi/documents/\*.txt /home/spoorthi/backup/

**Ln Commands**

1. Creating a Hard Link

*Scenario*: You have a file named data.txt in your current directory, and you want to create a hard link to it called data\_link.txt.  
Question: What command would you use?

ln data.txt data\_link.txt

2. Creating a Symbolic Link

*Scenario*: You want to create a symbolic link to a file named config.json located in /home/spoorthi/settings and place the link in your current directory. You want to name the link config\_link.json.  
Question: What command would you use?

ln -s /home/spoorthi/settings/config.json config\_link.json

3. Linking to a Directory

*Scenario*: You have a directory named scripts in your home directory, and you want to create a symbolic link to it in your current directory named my\_scripts.  
Question: What command would you use?

ln -s ~/scripts my\_scripts

4. Checking the Type of Link

*Scenario*: You created a symbolic link named link\_to\_data for a file called data.txt. Now, you want to check whether link\_to\_data is a hard link or a symbolic link.  
Question: What command would you use?

ls -l link\_to\_data

5. Removing a Symbolic Link

*Scenario*: You have a symbolic link named old\_config\_link in your current directory that you no longer need.  
Question: What command would you use to remove the symbolic link?

rm old\_config\_link

6. Creating a Relative Symbolic Link

*Scenario*: You are in the /home/spoorthi/projects directory, and you want to create a symbolic link to a file named example.txt located in the /home/spoorthi/projects/documents directory. You want to name the link example\_link.txt.  
Question: What command would you use?

ln -s documents/example.txt example\_link.txt

7. Creating a Hard Link in a Different Directory

*Scenario*: You want to create a hard link to report.txt located in the /home/spoorthi/reports directory and place the link in the /home/spoorthi/backup directory.  
Question: What command would you use?

ln /home/spoorthi/reports/report.txt /home/spoorthi/backup/report\_link.txt

8. Link to a File with Spaces in the Name

*Scenario*: You have a file named My Report.txt in your current directory, and you want to create a symbolic link to it called My\_Report\_Link.txt.  
Question: How would you write the ln command to handle spaces in the filename?

ln -s "My Report.txt" "My\_Report\_Link.txt"

or using a backslash to escape the space:

ln -s My\ Report.txt My\_Report\_Link.txt

9. Linking to an External Storage Device

*Scenario*: You have an external drive mounted at /media/external\_drive, and you want to create a symbolic link to a file named backup.tar.gz located in that drive. You want to place the link in your current directory and name it backup\_link.tar.gz.  
Question: What command would you use?  
ln -s /media/external\_drive/backup.tar.gz backup\_link.tar.gz

10. Creating Multiple Links

*Scenario*: You want to create multiple symbolic links for the same file, script.sh, with different names: script1.sh, script2.sh, and script3.sh.  
Question: What command would you use?  
Answer:

ln -s script.sh script1.sh

ln -s script.sh script2.sh

ln -s script.sh script3.sh

or using a loop:

for i in 1 2 3; do ln -s script.sh script$i.sh; done

**chmod Commands**

1. Granting Read Permission

*Scenario*: You have a script named backup.sh, and you want to give read permission to all users.  
Question: What command would you use?

chmod a+r backup.sh

2. Revoking Write Permission

*Scenario*: You have a file named report.txt, and you want to remove write permission for the group.  
Question: What command would you use?

chmod g-w report.txt

3. Granting Execute Permission

*Scenario*: You have a file named script.sh, and you want to allow all users to execute it.  
Question: What command would you use?

chmod a+x script.sh

4. Setting Permissions Using Numeric Mode

*Scenario*: You want to set the permissions of a file named data.txt to read and write for the owner, and read-only for the group and others.  
Question: What command would you use?

chmod 644 data.txt

5. Setting Permissions Recursively

*Scenario*: You have a directory named my\_folder and you want to give read, write, and execute permissions to the owner and the group for all files and subdirectories inside it.  
Question: What command would you use?

chmod -R 770 my\_folder

6. Making a Directory Accessible

*Scenario*: You have a directory named shared\_folder, and you want to give read and execute permissions to everyone, allowing users to list and access the directory's contents.  
Question: What command would you use?

chmod a+rx shared\_folder

7. Removing All Permissions

*Scenario*: You want to remove all permissions (read, write, execute) for a file named private.txt for all users.  
Question: What command would you use?

chmod a-rwx private.txt

8. Granting Special Permissions (Set uid)

*Scenario*: You have an executable file named install.sh, and you want it to run with the permissions of the file owner, regardless of who executes it.  
Question: What command would you use?

chmod u+s install.sh

9. Using Symbolic Mode for Multiple Changes

*Scenario*: You have a file named archive.zip, and you want to remove write permission for others and add read permission for the group.  
Question: What command would you use?

chmod o-w,g+r archive.zip

10. Making a Script Executable

*Scenario*: You have a script named deploy.sh that you want to make executable only for the owner.  
Question: What command would you use?

chmod u+x deploy.sh

11. Setting Group Permissions

*Scenario*: You want to set the group permission of a file named team.txt to read and write, while leaving the owner and others' permissions unchanged.  
Question: What command would you use?

chmod g+rw team.txt

12. Ensuring No Write Permission

*Scenario*: You want to ensure that a file named important.doc has no write permissions for anyone (owner, group, others).  
Question: What command would you use?

chmod a-w important.doc

**Chown commands**

1. Changing the Owner of a File

*Scenario*: You have a file named report.txt and you want to change its owner to a user named alice.  
Question: What command would you use?

chown alice report.txt

2. Changing the Owner and Group of a File

*Scenario*: You want to change both the owner and group of a file named data.csv to bob and admins, respectively.  
Question: What command would you use?

chown bob:admins data.csv

3. Changing Ownership Recursively

*Scenario*: You have a directory named project\_files, and you want to change the ownership of all files and subdirectories within it to a user named charlie.  
Question: What command would you use?

chown -R charlie project\_files

4. Changing Ownership with Sudo

*Scenario*: You want to change the owner of a system file named config.ini to the user admin, but you need superuser privileges to do this.  
Question: What command would you use?

sudo chown admin config.ini

5. Changing Group Ownership Only

*Scenario*: You have a file named logfile.log, and you want to change its group ownership to developers while keeping the owner unchanged.  
Question: What command would you use?

chown :developers logfile.log

6. Changing Ownership of Multiple Files

*Scenario*: You want to change the owner of three files named file1.txt, file2.txt, and file3.txt to dave.  
Question: What command would you use?

chown dave file1.txt file2.txt file3.txt

7. Changing Ownership for a Directory and Its Contents

*Scenario*: You have a directory named shared\_data and you want to change its owner to frank along with all its contents.  
Question: What command would you use?

chown -R frank shared\_data

8. Changing Ownership to a User and Group

*Scenario*: You have a file named presentation.pptx, and you want to change its owner to grace and its group to marketing.  
Question: What command would you use?

chown grace:marketing presentation.pptx

9. Confirming Ownership Change

*Scenario*: After changing the owner of file.txt to henry, you want to confirm the ownership change.  
Question: What command would you use to check the ownership?

ls -l file.txt

10. Changing Ownership of a Symbolic Link

*Scenario*: You have a symbolic link named link\_to\_data pointing to data.txt, and you want to change the owner of the original data.txt file to julia.  
Question: What command would you use?

chown julia data.txt

11. Setting Ownership Using Numeric UID and GID

*Scenario*: You want to change the ownership of a file named test.txt to a user with UID 1001 and a group with GID 1001.  
Question: What command would you use?

chown 1001:1001 test.txt

12. Preventing Ownership Change on a Read-Only File

*Scenario*: You have a file named secret.txt, and you want to ensure it remains owned by admin, even if someone tries to change it.  
Question: What can you do to prevent changes to the ownership?  
Answer: You can set the immutable attribute:

sudo chattr +i secret.txt

**ISOF commands**

*Scenario* 1: Finding Open Files by a Specific Process

Question: You want to see all the files opened by a process with the PID 1234. How can you achieve this using the lsof command?

Answer: Use the following command to list all open files for that specific PID:

lsof -p 1234

*Scenario* 2: Checking Files Opened by a User

Question: You need to check all files opened by the user john. What command should you use?

Answer: You can filter the results by the username:

lsof -u john

*Scenario* 3: Finding Files Opened on a Specific Port

Question: You want to find out which processes are using port 80. How can you do this?

Answer:  
Use the -i option to specify the internet address:

lsof -i :80

*Scenario* 4: Listing All Open Network Connections

Question: You need to see all open network connections and the associated processes. What command will you use?

Answer: Run the following command to list all network connections:

lsof -i

*Scenario* 5: Finding Files Opened by a Specific Command

Question: You want to find all files opened by processes that were started by the apache2 command. How can you do this?

Answer: You can filter the results by command name:

lsof -c apache2

*Scenario* 6: Checking for Deleted Files Still in Use

Question: You suspect that there are deleted files still being used by processes. How can you list these files?

Answer: Use the +L1 option to find deleted files:

lsof +L1

*Scenario* 7: Finding the Parent Process ID (PPID)

Question: You want to find the parent process ID of all the processes that are currently using files. How can you do this?

Answer:You can include the PPID in the output:

lsof -F p

*Scenario* 8: Finding Open Files on a Specific Filesystem

Question: You want to check all the files that are opened on the /var/log filesystem. What command should you use?

Answer: You can specify the directory:

lsof +D /var/log

*Scenario* 9: Displaying File Descriptors

Question: You want to see all file descriptors for a specific process with PID 5678. How can you list them?  
Use the following command:

lsof -p 5678 -F

*Scenario* 10: Finding Open Files with Specific Extensions

Question: You need to check for open files with the .log extension. How can you filter the output for this?  
You can use the grep command to filter the output:

lsof | grep '\.log$'

*Scenario* 11: Checking for Open Files on a Specific Device

Question: You want to list all files that are opened on the device /dev/sda1. What command will you use?

Answer: You can filter by the device:

lsof /dev/sda1

*Scenario* 12: Listing Open Files by Process Name

Question: You want to find all open files for the sshd service. What command should you use?  
Run the following command:

lsof -c sshd

*Scenario* 13: Checking for Processes Using a Specific File

Question: You want to find out which processes are currently using a specific file, /etc/passwd. How can you do this?  
You can use the following command:

lsof /etc/passwd

*Scenario* 14: Finding All Open Files with a Specific User

Question: You want to list all open files by the user with UID 1001. What command would you use?  
You can filter by UID:

lsof -u 1001

*Scenario* 15: Displaying Open Files with Size Information

Question: You want to display open files along with their sizes. How can you accomplish this?

You can use the following command:

lsof -s

**SUID commands**

SUID (Set User ID) is a special type of permission that allows users to run an executable file with the permissions of the file owner, typically root. This feature is crucial for certain system programs that need elevated privileges to perform specific tasks. Below are some common commands and concepts related to SUID.

Common SUID Commands

1. Setting the SUID Bit
   * Command: chmod u+s filename
   * Description: Sets the SUID bit on the specified file, allowing it to be executed with the privileges of the file owner.

sudo chmod u+s my\_script.sh

1. Removing the SUID Bit
   * Command: chmod u-s filename
   * Description: Removes the SUID bit from the specified file.

sudo chmod u-s my\_script.sh

1. Listing SUID Files
   * Command: find / -type f -perm -4000 2>/dev/null
   * Description: Searches for all files on the system with the SUID bit set.

find / -type f -perm -4000 2>/dev/null

1. Viewing File Permissions
   * Command: ls -l filename
   * Description: Displays detailed file permissions, including whether the SUID bit is set.

ls -l /usr/bin/passwd

1. Running a SUID Program
   * Command: ./filename
   * Description: Executes the SUID program as the user who runs it, but with the privileges of the file owner.

./suid\_program

1. Finding SUID Programs Owned by a Specific User
   * Command: find / -user username -type f -perm -4000 2>/dev/null
   * Description: Lists SUID programs owned by a specified user.

find / -user root -type f -perm -4000 2>/dev/null

1. Checking the Owner of SUID Files
   * Command: find / -type f -perm -4000 -exec ls -l {} \; 2>/dev/null
   * Description: Lists SUID files along with their owners and permissions.

find / -type f -perm -4000 -exec ls -l {} \; 2>/dev/null

1. Finding All SUID and SGID Programs Together
   * Command: find / -type f \( -perm -4000 -o -perm -2000 \) 2>/dev/null
   * Description: Lists both SUID and SGID programs.

find / -type f \( -perm -4000 -o -perm -2000 \) 2>/dev/null

1. Checking for Vulnerabilities in SUID Programs
   * Command: Using tools like lynis or rkhunter.
   * Description: These tools can scan for vulnerabilities in SUID programs.

sudo lynis audit system

**SGID commands**

SGID (Set Group ID) is a special permission that allows users to run an executable file with the permissions of the file's group owner. It can also be applied to directories to ensure that files created within inherit the group ownership of the directory. Here are some common commands and concepts related to SGID.

Common SGID Commands

1. Setting the SGID Bit
   * Command: chmod g+s filename
   * Description: Sets the SGID bit on the specified file, allowing it to be executed with the permissions of the file's group owner.

chmod g+s my\_script.sh

1. Removing the SGID Bit
   * Command: chmod g-s filename
   * Description: Removes the SGID bit from the specified file.

chmod g-s my\_script.sh

1. Listing SGID Files
   * Command: find / -type f -perm -2000 2>/dev/null
   * Description: Searches for all files on the system with the SGID bit set.

find / -type f -perm -2000 2>/dev/null

1. Viewing File Permissions
   * Command: ls -l filename
   * Description: Displays detailed file permissions, including whether the SGID bit is set.

ls -l /usr/bin/sgid\_program

1. Finding SGID Programs Owned by a Specific Group
   * Command: find / -group groupname -type f -perm -2000 2>/dev/null
   * Description: Lists SGID programs owned by a specified group.

find / -group developers -type f -perm -2000 2>/dev/null

1. Checking the Owner of SGID Files
   * Command: find / -type f -perm -2000 -exec ls -l {} \; 2>/dev/null
   * Description: Lists SGID files along with their owners and permissions.

find / -type f -perm -2000 -exec ls -l {} \; 2>/dev/null

1. Finding All SGID and SUID Programs Together
   * Command: find / -type f \( -perm -4000 -o -perm -2000 \) 2>/dev/null
   * Description: Lists both SGID and SUID programs.

find / -type f \( -perm -4000 -o -perm -2000 \) 2>/dev/null

1. Checking for Vulnerabilities in SGID Programs
   * Command: Use tools like lynis or rkhunter.
   * Description: These tools can scan for vulnerabilities in SGID programs.

sudo lynis audit system

1. Creating a SGID Directory
   * Command: mkdir directory\_name && chmod g+s directory\_name
   * Description: Creates a directory and sets the SGID bit so that files created within inherit the group ownership.

mkdir shared\_dir && chmod g+s shared\_dir

1. Viewing Directory SGID Permissions
   * Command: ls -ld directory\_name
   * Description: Displays detailed permissions for a directory, indicating if it has the SGID bit set.

ls -ld shared\_dir

**stickybits**

The sticky bit is a special permission that can be set on directories (and occasionally on files) to control how files within those directories can be deleted or renamed. When the sticky bit is set on a directory, only the owner of a file can delete or rename the file, regardless of the directory's write permissions. This is particularly useful in shared directories, such as /tmp, where many users have write access.

Common Sticky Bit Commands

1. Setting the Sticky Bit
   * Command: chmod +t directory\_name
   * Description: Sets the sticky bit on the specified directory.

chmod +t /tmp

1. Removing the Sticky Bit
   * Command: chmod -t directory\_name
   * Description: Removes the sticky bit from the specified directory.

chmod -t /tmp

1. Listing Sticky Bit Directories
   * Command: find / -type d -perm -1000 2>/dev/null
   * Description: Searches for all directories with the sticky bit set.

find / -type d -perm -1000 2>/dev/null

1. Viewing Directory Permissions
   * Command: ls -ld directory\_name
   * Description: Displays detailed permissions for a directory, including whether the sticky bit is set. The sticky bit is represented by a t at the end of the permissions.

ls -ld /tmp

1. Setting Sticky Bit with Other Permissions
   * Command: chmod 1777 directory\_name
   * Description: Sets the permissions of a directory to allow full access for everyone (read, write, execute) and sets the sticky bit.

chmod 1777 /tmp

Example Usage

1. Creating a Shared Directory with Sticky Bit
   * If you want to create a shared directory where users can add files but can only delete their own files, you would set the sticky bit:

mkdir shared\_directory

chmod 1777 shared\_directory

1. Checking Sticky Bit Status
   * You can check if the sticky bit is set on the /tmp directory:

ls -ld /tmp

* + The output might look like this:

**drwxrwxrwt 10 root root 4096 Sep 29 14:32 /tmp**

**Linux user management commands**

**useradd commands**

1. Creating a New User with a Custom Home Directory

* Scenario: You are asked to create a new user named john on a server. However, instead of the default /home/john, the home directory should be set to /data/john.
* Question: How would you create the user john with a custom home directory /data/john using the useradd command?
* Expected Command:

sudo useradd -m -d /data/john john

2. Creating a User with an Expiration Date

* Scenario: A temporary user guest needs to be added to the system. The account should expire on a specific date (e.g., 2024-10-15).
* Question: How would you create the user guest and set their account to expire on October 15, 2024?
* Expected Command:

sudo useradd -e 2024-10-15 guest

3. Creating a User without a Home Directory

* Scenario: You are tasked with creating a service account for an application, and the account doesn’t need a home directory.
* Question: How would you create a user appuser without generating a home directory?
* Expected Command:

sudo useradd -M appuser

4. Creating a User and Adding Them to Multiple Groups

* Scenario: You are asked to create a new user devuser who needs access to multiple groups (developers and docker).
* Question: How would you create the user devuser and add them to both the developers and docker groups?
* Expected Command:

sudo useradd -G developers,docker devuser

5. Setting a Default Shell for a New User

* Scenario: You need to create a user testuser who will use /bin/bash as their default shell instead of the system’s default shell.
* Question: How would you set up testuser with /bin/bash as the default shell using useradd?
* Expected Command:

sudo useradd -s /bin/bash testuser

6. Creating a User with a Specific User ID (UID)

* Scenario: You are required to create a new user devops with a specific UID of 1500 for auditing purposes.
* Question: What command would you use to create the user devops with a UID of 1500?
* Expected Command:

sudo useradd -u 1500 devops

7. Changing the Default User Creation Settings

* Scenario: By default, the new users are created with a home directory in /home. You need to change this default directory to /srv/users for all new users.
* Question: How would you modify the useradd command’s default behavior so that future users are created with home directories in /srv/users?
* Expected Command:

sudo useradd -D -b /srv/users

8. Locking a User Account upon Creation

* Scenario: For security purposes, you are required to create a user audituser, but their account should remain locked until the password is set.
* Question: How would you create audituser and ensure their account is locked?
* Expected Command:

sudo useradd -L audituser

9. Creating a User with a Predefined Password

* Scenario: You need to create a new user newhire and assign a predefined password Welcome123 at the time of creation.
* Question: How would you add the user newhire and set their password to Welcome123 using useradd?
* Expected Steps:
  1. Create the user:

sudo useradd newhire

* 1. Set the password:

echo 'newhire:Welcome123' | sudo chpasswd

10. Checking the Default Values for User Creation

* Scenario: You want to review the default values that the useradd command uses (e.g., default home directory location, shell, etc.) before creating new users.
* Question: What command would you use to view the current default settings for the useradd command?
* Expected Command:

sudo useradd -D

**groupadd commands**

1. Creating a New Group with a Specific GID

* Scenario: You need to create a new group called devteam, and it must have a specific GID of 2001 for consistency across multiple servers.
* Question: How would you create the group devteam with the GID 2001?
* Expected Command:

sudo groupadd -g 2001 devteam

2. Creating a System Group

* Scenario: You need to create a system group called sysadmins for system-related users. This group should have a GID below 1000 (reserved for system groups).
* Question: How would you create a system group sysadmins?
* Expected Command:

sudo groupadd -r sysadmins

3. Modifying an Existing Group’s GID

* Scenario: The GID of an existing group finance needs to be changed from 1500 to 1600 due to company policy changes.
* Question: How would you modify the GID of the finance group from 1500 to 1600?
* Expected Command:

sudo groupmod -g 1600 finance

4. Creating a Group with a Password

* Scenario: You are required to create a group devgroup that needs to have a password for certain users to join it.
* Question: How would you create the group devgroup and assign a password to it?
* Expected Steps:
  1. Create the group:

sudo groupadd devgroup

* 1. Assign a password to the group:

sudo gpasswd devgroup

5. Adding Multiple Users to a Group Upon Creation

* Scenario: You are tasked with creating a new group qa and immediately adding the users qauser1 and qauser2 to this group.
* Question: How would you create the group qa and add the users qauser1 and qauser2 to it in one step?
* Expected Command:

sudo groupadd qa && sudo usermod -aG qa qauser1 && sudo usermod -aG qa qauser2

6. Deleting a Group Safely

* Scenario: The testgroup is no longer needed, and you need to delete it. However, you want to ensure that no files or directories owned by testgroup will be affected.
* Question: How would you safely delete the group testgroup?
* Expected Command:

sudo groupdel testgroup

7. Creating a New Group with a Specific Range of GID

* Scenario: Your organization has a policy that the GID for new groups should be between 2000 and 3000. You need to create a new group called research that follows this policy.
* Question: How would you create the research group and ensure that its GID falls within the required range?
* Expected Command:

sudo groupadd -g 2005 research

8. Migrating Users to a New Group

* Scenario: The group marketing is being deprecated, and all users in that group need to be moved to a new group sales.
* Question: How would you create the sales group and migrate all users from the marketing group to sales?
* Expected Steps:
  1. Create the new group:

sudo groupadd sales

* 1. List all users in the marketing group:

grep marketing /etc/group

* 1. Add each user from marketing to sales:

sudo usermod -aG sales <username>

9. Viewing Group Details

* Scenario: You want to check the details of an existing group engineering, such as its GID and the users who are part of it.
* Question: How would you view the details of the engineering group?
* Expected Command:

getent group engineering

10. Preventing Group Duplication

* Scenario: Your organization has a strict policy against duplicate group names and GIDs. Before creating a group admi team, you want to ensure that there are no existing groups with this name or GID.
* Question: How would you check whether a group name or GID already exists before creating a new group?
* Expected Command:

getent group adminteam # Check for group name

getent group <GID> # Check for specific GI

**userdel commands**

1. Deleting a User Account

* Scenario: You need to remove a user named guestuser from the system because they no longer require access.
* Question: How would you delete the guestuser account while ensuring their home directory and files remain on the system?
* Expected Command:

sudo userdel guestuser

2. Deleting a User and Their Home Directory

* Scenario: You are tasked with deleting a user named olduser, and their home directory and mail spool should also be removed.
* Question: How would you delete the olduser account along with their home directory?
* Expected Command:

sudo userdel -r olduser

3. Removing a User Who Is Currently Logged In

* Scenario: You need to delete a user testuser, but they are currently logged in. You are unsure whether you can safely remove the account while they are logged in.
* Question: What would happen if you try to delete a user who is currently logged in, and how would you safely delete their account?
* Expected Explanation:
  + If a user is logged in, the userdel command will still remove the account, but it won’t terminate their session immediately.
  + To safely remove the account, first terminate the user’s session:

sudo pkill -u testuser

sudo userdel testuser

4. Handling User Files After Deletion

* Scenario: You have deleted a user devuser using userdel, but now you're unsure what happens to the files owned by the user outside their home directory.
* Question: What happens to files owned by a deleted user that are located outside the user's home directory, and how would you find and reassign them to another user?
* Expected Explanation:
  + Files outside the home directory remain on the system with the original user’s UID, even after the user is deleted.
  + To find these files:

sudo find / -uid <UID\_of\_deleted\_user>

* + To reassign ownership of the files to another user:

sudo chown newuser:newuser <file>

5. Force Deleting a User with Running Processes

* Scenario: You need to delete a user named dbuser, but they have some running processes, and the userdel command is failing.
* Question: How would you forcefully delete dbuser along with terminating their running processes?
* Expected Command:

sudo userdel -f dbuser

6. Deleting a User without Removing Group Membership

* Scenario: A user alice is part of several groups. You need to delete her account but ensure that the groups she was part of remain intact for other users.
* Question: How would you delete the alice account without affecting the groups she was a member of?
* Expected Command:

sudo userdel alice

* + The userdel command only removes the user and doesn’t delete groups. The groups will remain, but alice will no longer be a member.

7. Deleting a System User

* Scenario: You need to delete a system user backupuser that was created for automated backups. The system user does not have a home directory.
* Question: How would you delete the system user backupuser?
* Expected Command:

sudo userdel backupuser

8. Verifying a User Account Has Been Deleted

* Scenario: After running the userdel command for the user bob, you want to verify that their account has been fully removed from the system.
* Question: How would you check if the bob account has been deleted?
* Expected Command:
  + Check the /etc/passwd file:

cat /etc/passwd | grep bob

* + Check the /etc/shadow file:

cat /etc/shadow | grep bob

9. Preventing Deletion of a Primary Group While Deleting a User

* Scenario: You have a user john who belongs to a group with the same name (i.e., john). You need to delete the user but want to avoid deleting the john group.
* Question: What happens to the group with the same name when you delete the user john and how would you ensure the group is retained?
* Expected Command:
  + By default, userdel will only delete the user and not their primary group unless explicitly instructed. The john group will remain unless you manually delete it using:

sudo groupdel john

* + If you want to retain the group, no extra action is needed; just run:

sudo userdel john

10. Deleting a User and Verifying Ownership of System Files

* Scenario: After deleting the user devops, you want to ensure that no system files are left behind under the deleted user's ownership.
* Question: How would you find and verify if any system files are still owned by the deleted user?
* Expected Command:

sudo find / -uid <UID\_of\_deleted\_user>

**usermod commands**

1. Changing a User’s Login Name

* Scenario: The user john\_doe has requested a change in their login name to john. The account details and home directory should remain the same.
* Question: How would you change the login name from john\_doe to john?
* Expected Command:

sudo usermod -l john john\_doe

2. Changing a User's Home Directory

* Scenario: A user devuser has their home directory in /home/devuser, but you need to move it to /data/devuser. You must ensure that their existing files are moved to the new location.
* Question: How would you change the home directory for devuser to /data/devuser and move all their files there?
* Expected Command:

sudo usermod -d /data/devuser -m devuser

3. Adding a User to a Secondary Group

* Scenario: You need to add the user alice to the docker group so she can use Docker services.
* Question: How would you add alice to the docker group without affecting her current group memberships?
* Expected Command:

sudo usermod -aG docker alice

4. Changing a User's Default Shell

* Scenario: A user bob is using /bin/sh as their default shell, but they prefer to use /bin/bash.
* Question: How would you change the default shell for bob to /bin/bash?
* Expected Command:

sudo usermod -s /bin/bash bob

5. Setting a User's Account Expiration Date

* Scenario: The user contractor is a temporary employee whose account should expire on 2024-12-31.
* Question: How would you set the expiration date for the contractor account to 2024-12-31?
* Expected Command:

sudo usermod -e 2024-12-31 contractor

6. Locking a User Account

* Scenario: You need to lock the user testuser account to prevent them from logging in until further notice, but you don't want to delete the account.
* Question: How would you lock the testuser account?
* Expected Command:

sudo usermod -L testuser

7. Unlocking a Locked User Account

* Scenario: The user james has requested access again, and you need to unlock his previously locked account.
* Question: How would you unlock the james account?
* Expected Command:

sudo usermod -U james

8. Changing a User's UID

* Scenario: A user developer was created with an incorrect user ID (UID). The current UID is 1501, but it needs to be changed to 2001.
* Question: How would you change the UID for developer to 2001?
* Expected Command:

sudo usermod -u 2001 developer

9. Changing a User's Primary Group

* Scenario: The user sam has sam as their primary group. You need to change their primary group to admins for better access control.
* Question: How would you change sam's primary group to admins?
* Expected Command:

sudo usermod -g admins sam

10. Setting or Changing a User's Comment (GECOS) Field

* Scenario: You want to add a full name for the user jenkinsuser, so it shows up when using the finger command or when querying the user information.
* Question: How would you add the full name "Jenkins Build User" for jenkinsuser?
* Expected Command:

sudo usermod -c "Jenkins Build User" jenkinsuser

11. Preventing a User from Logging in by Changing the Shell

* Scenario: You want to prevent the user backupuser from logging in to the system, but you still need their account for automated tasks.
* Question: How would you prevent backupuser from logging in by modifying their shell?
* Expected Command:

sudo usermod -s /sbin/nologin backupuser

12. Assigning a User to Multiple Secondary Groups

* Scenario: You are asked to add the user projectadmin to both the devops and admin groups without affecting any other group memberships.
* Question: How would you add projectadmin to both devops and admin groups?
* Expected Command:

sudo usermod -aG devops,admin projectadmin

13. Changing a User's Password Expiry Settings

* Scenario: You need to set the password expiry for the user securityuser so that the password expires every 60 days.
* Question: How would you set the password expiry for securityuser to 60 days?
* Expected Command:

sudo usermod -f 60 securityuser

14. Forcing a User to Change Password on Next Login

* Scenario: The user newuser was created, but you want to force them to change their password the first time they log in.
* Question: How would you ensure that newuser is prompted to change their password on their next login?
* Expected Command:

sudo chage -d 0 newuser

**passwd commands**

1. Changing Your Own Password

* Scenario: You want to change your current password for your account.
* Question: How would you change your own password?
* Expected Command:

passwd

2. Changing Another User’s Password

* Scenario: You need to change the password for the user bob because they forgot it.
* Question: How would you change bob’s password?
* Expected Command:

sudo passwd bob

3. Forcing a User to Change Password on Next Login

* Scenario: You want to force the user john to change his password the next time he logs in.
* Question: How would you make john change his password on the next login?
* Expected Command:

sudo passwd -e john

4. Locking a User Account

* Scenario: You want to lock the account of the user guestuser to prevent any logins.
* Question: How would you lock guestuser's account?
* Expected Command:

sudo passwd -l guestuser

5. Unlocking a Locked User Account

* Scenario: You need to unlock the user account dave after it was previously locked.
* Question: How would you unlock the dave account?
* Expected Command:

sudo passwd -u dave

6. Setting Password Expiry for a User

* Scenario: You need to set the password for alice to expire in 30 days.
* Question: How would you set the password expiry for alice to 30 days?
* Expected Command:

sudo passwd -x 30 alice

7. Disabling a User's Password (No Password Login)

* Scenario: You need to disable the password for the user scriptuser so they can only log in using SSH keys.
* Question: How would you disable scriptuser's password?
* Expected Command:

sudo passwd -d scriptuser

8. Setting a Password with a Minimum Change Time

* Scenario: You want to set a minimum of 5 days before the user testuser can change their password again.
* Question: How would you configure testuser’s account so that they must wait at least 5 days before changing their password?
* Expected Command:

sudo passwd -n 5 testuser

**su commands**

1. Switching to Root User

* Scenario: You need to perform administrative tasks, and you want to switch to the root user.
* Question: How would you switch to the root user using su?
* Expected Command:

su -

2. Switching to Another User's Account

* Scenario: You are logged in as alice, and you need to switch to the user bob to run a command under their account.
* Question: How would you switch to the bob account?
* Expected Command:

su - bob

3. Running a Single Command as Another User

* Scenario: You are logged in as admin and want to run the command ls /home/bob as the user bob without fully switching to their account.
* Question: How would you use su to run a single command as bob?
* Expected Command:

su - bob -c "ls /home/bob"

4. Switching Without Environment Variables

* Scenario: You need to switch to the testuser account but want to keep your current environment variables.
* Question: How would you switch to testuser without loading their environment?
* Expected Command:

su testuser

5. Switching to Root with a Specific Shell

* Scenario: You want to switch to the root user but use /bin/sh as the shell instead of the default shell.
* Question: How would you switch to the root user and use /bin/sh?
* Expected Command:

su -s /bin/sh -

6. Exiting from a Switched User Session

* Scenario: After switching to the devuser account, you have completed the tasks and need to return to your original account.
* Question: How would you exit from the devuser session and go back to the original user?
* Expected Command:

exit

7. Checking Whether You're Currently Using su

* Scenario: You are unsure if you’re currently using su to operate as another user.
* Question: How can you check whether you’ve switched to another user using su?
* Expected Command:

whoami

**sudo commands**

1. Running a Command with Elevated Privileges

* Scenario: You want to update the system using the apt update command, but you do not have root privileges.
* Question: How would you run the apt update command as root?
* Expected Command:

sudo apt update

2. Switching to Root User Temporarily

* Scenario: You need to perform multiple administrative tasks that require root privileges. Instead of using sudo for each command, you want to switch to the root user temporarily.
* Question: How would you switch to the root user using sudo?
* Expected Command:

sudo su

3. Editing a System File with Root Privileges

* Scenario: You need to edit the /etc/hosts file, but you don’t have write permissions.
* Question: How would you edit the /etc/hosts file using a text editor with root privileges?
* Expected Command:

sudo nano /etc/hosts

4. Adding a User to the Sudoers File

* Scenario: You want to give the user devuser the ability to run commands with sudo.
* Question: How would you add devuser to the sudoers file?
* Expected Command:

sudo usermod -aG sudo devuser

5. Running a Command Without Being Asked for a Password

* Scenario: You want to allow a specific command, such as shutdown, to run without asking for a password.
* Question: How would you configure sudo to allow password-less execution of the shutdown command?
* Expected Command:

sudo visudo

Add this line:

<username> ALL=(ALL) NOPASSWD: /sbin/shutdown

6. Viewing the Sudoers Configuration

* Scenario: You want to view the current sudoers configuration to ensure it has the correct settings.
* Question: How would you safely view and edit the sudoers file?
* Expected Command:

sudo visudo

7. Listing Available Commands for a Sudo User

* Scenario: You want to see what commands you, as a sudo user, are allowed to run.
* Question: How would you list the commands you are permitted to execute with sudo?
* Expected Command:

sudo -l

8. Checking System Logs for Sudo Actions

* Scenario: You suspect that a user has been misusing their sudo privileges and want to check the logs.
* Question: How would you check the system logs for sudo actions?
* Expected Command:

sudo cat /var/log/auth.log | grep sudo

9. Running a Command as Another User

* Scenario: You need to run a specific command as the user backupuser instead of root.
* Question: How would you use sudo to run a command as a different user?
* Expected Command:

sudo -u backupuser <command>

10. Giving Temporary Sudo Privileges

* Scenario: You want to temporarily allow a user tempuser to run commands with sudo for a limited time.
* Question: How would you give tempuser temporary sudo privileges?
* Expected Command:

sudo usermod -aG sudo tempuser

* + To revoke the privileges after the period ends:

sudo deluser tempuser sudo

**chage commands**

1. Checking Password Expiration Details for a User

Scenario: You want to see when the user bob last changed their password and when it will expire.  
Question: What command would you use to view the password expiration details for bob?  
Answer:

chage -l bob

2. Forcing a User to Change Their Password at the Next Login

Scenario: The user alice has not updated her password for a long time, and you want to force her to change her password the next time she logs in.  
Question: What command would you use to require alice to change her password upon next login?  
Answer:

chage -d 0 alice

3. Setting Password Expiration to 60 Days for a User

Scenario: You want to configure the user john so that his password expires every 60 days, forcing him to change it regularly.  
Question: What command would you use to set a 60-day password expiration for john?  
Answer:

chage -M 60 john

4. Disabling Password Expiration for a User

Scenario: You want to make sure that the user steve never has to change his password, i.e., disabling password expiration for his account.  
Question: What command would you use to disable password expiration for steve?  
Answer:

chage -M -1 steve

5. Setting the Minimum Number of Days Between Password Changes

Scenario: You want to prevent the user dave from changing his password too frequently, so you want to set a minimum of 7 days between password changes.  
Question: What command would you use to set a 7-day minimum between password changes for dave?  
Answer:

chage -m 7 dave

6. Setting an Account Expiry Date

Scenario: You want the user carol’s account to automatically expire on January 1st, 2025.  
Question: What command would you use to set the account expiry date for carol?  
Answer:

chage -E 2025-01-01 carol

7. Setting a Warning Before Password Expiration

Scenario: You want to notify the user mike 7 days before his password is set to expire, giving him time to change it.  
Question: What command would you use to set a 7-day warning for password expiration for mike?  
Answer:

chage -W 7 mike

8. Locking a User Account After Password Expiry

Scenario: You want to configure the user jenny’s account so that if she does not change her password within 5 days after it expires, her account gets locked.  
Question: What command would you use to set the account lockout after 5 days for jenny?  
Answer:

chage -I 5 jenny

**Linux process management commands**

1. top

* Scenario: You want to monitor the CPU and memory usage of your system in real-time.
  + Question: Which command would you use to view the real-time performance of your system?
    - Expected Command:

top

* Scenario: You want to sort processes by memory usage while using the top command.
  + Question: How would you achieve that?
    - Expected Action: Press M while top is running.
* Scenario: You need to refresh the top display every 2 seconds instead of the default 3 seconds.
  + Question: How can you set the refresh rate to 2 seconds?
    - Expected Action: Press d, then enter 2.
* Scenario: You want to quit the top command after monitoring.
  + Question: How would you exit the top interface?
    - Expected Action: Press q.
* Scenario: You need to see only processes owned by a specific user, devuser, in top.
  + Question: What action would you take to filter the processes?
    - Expected Action: Press u, then enter devuser.
* Scenario: You want to change the priority (nice value) of a running process from top.
  + Question: How can you do that for a process with PID 1234?
    - Expected Action: Press r, enter 1234, and then set the new nice value.
* Scenario: You are interested in monitoring a specific command, python, in top.
  + Question: How would you highlight this command?
    - Expected Action: Use Shift + H to show threads.
* Scenario: You want to get a summary of system uptime and load average.
  + Question: Where can you find this information in top?
    - Expected Action: Look at the top summary area.
* Scenario: You want to log the output of top to a file for later analysis.
  + Question: What command would you use to start logging the output?
    - Expected Command:

top -b -n 1 > top\_output.txt

* Scenario: You want to adjust the fields displayed in top.
  + Question: How can you customize the columns displayed in top?
    - Expected Action: Press f to select fields.

2. htop

* Scenario: You want to use an enhanced version of top that has a better interface.
  + Question: Which command would you use to launch it?
    - Expected Command:

htop

* Scenario: You want to kill a process directly from htop.
  + Question: How would you kill a process with PID 1234?
    - Expected Action: Navigate to the process, press F9, then select SIGTERM and press Enter.
* Scenario: You want to sort the processes by CPU usage in htop.
  + Question: How would you achieve that?
    - Expected Action: Press F6 to sort by CPU.
* Scenario: You need to search for a process named nginx.
  + Question: How can you search for the nginx process in htop?
    - Expected Action: Press F3, then type nginx.
* Scenario: You want to change the nice value of a process from within htop.
  + Question: How can you do that for PID 4567?
    - Expected Action: Select the process, press F7 to increase and F8 to decrease.
* Scenario: You want to display only processes related to the user devuser.
  + Question: How can you filter by user in htop?
    - Expected Action: Press F4, then enter devuser.
* Scenario: You want to quit htop after monitoring.
  + Question: What key do you press to exit htop?
    - Expected Action: Press F10.
* Scenario: You want to see system resource usage graphs in htop.
  + Question: Where are these graphs displayed?
    - Expected Action: They are displayed at the top of the htop interface.
* Scenario: You want to customize the appearance of htop.
  + Question: How can you access the setup menu to change settings?
    - Expected Action: Press F2 to enter the setup menu.
* Scenario: You want to log the output of htop.
  + Question: How can you start logging the output to a file?
    - Expected Command:

htop -b > htop\_output.txt

3. fg

* Scenario: You have a background process that needs to be brought to the foreground.
  + Question: How would you bring a job with job ID 1 to the foreground?
    - Expected Command:

fg %1

* Scenario: You want to resume the most recent background job.
  + Question: Which command would you use?
    - Expected Command:

fg

* Scenario: You need to check what background jobs are currently running.
  + Question: Which command shows the list of background jobs?
    - Expected Command:

jobs

* Scenario: You want to bring a specific background job that is not the most recent to the foreground.
  + Question: How would you specify the job ID to bring it to the foreground?
    - Expected Command:

fg %2

* Scenario: You accidentally started a long-running process in the foreground and want to stop it temporarily.
  + Question: How would you stop it and send it to the background?
    - Expected Action: Press Ctrl + Z, then use bg to resume it in the background.
* Scenario: You have multiple jobs running in the background and want to bring the one named my\_script.sh to the foreground.
  + Question: What command would you use?
    - Expected Command:

fg %my\_script.sh

* Scenario: After bringing a job to the foreground, you want to terminate it.
  + Question: How would you terminate it?
    - Expected Action: Press Ctrl + C.
* Scenario: You want to monitor the progress of a job running in the background.
  + Question: How can you bring it to the foreground to see its output?
    - Expected Action: Use fg to bring it to the foreground.
* Scenario: You have a background job that you want to stop without terminating it.
  + Question: What command would you use to stop it temporarily?
    - Expected Command:

fg

* Scenario: You mistakenly brought a job to the foreground that should be in the background. How would you send it back to the background?
  + Question: After stopping it, what command do you use?
    - Expected Command:

bg

4. bg

* Scenario: You have a process that was stopped (paused) with Ctrl + Z and want to resume it in the background.
  + Question: What command do you use to resume it in the background?
    - Expected Command:

bg

* Scenario: You want to continue a specific stopped job (job ID 2) in the background.
  + Question: How would you do that?
    - Expected Command:

bg %2

* Scenario: After stopping a process, you want to check the status of all jobs.
  + Question: What command would you use?
    - Expected Command:

jobs

* Scenario: You want to bring a stopped job back to the foreground instead of continuing it in the background.
  + Question: What command would you use?
    - Expected Command:

fg %1

* Scenario: You need to confirm that a job is running in the background.
  + Question: How can you check this?
    - Expected Command:

jobs

* Scenario: You have multiple jobs stopped, and you want to resume the last one in the background.
  + Question: How would you do that?
    - Expected Command:

bg

* Scenario: You started a process in the background but want to bring it to the foreground to monitor its output.
  + Question: How would you bring it to the foreground?
    - Expected Command:

fg %job\_id

* Scenario: You want to suspend a process that is currently running in the foreground.
  + Question: What key combination would you use to suspend it?
    - Expected Action: Press Ctrl + Z.
* Scenario: You have forgotten which jobs are running in the background and want to list them.
  + Question: What command do you use to view the background jobs?
    - Expected Command:

jobs

* Scenario: You have a process running in the background that you need to stop.
  + Question: How would you stop that process?
    - Expected Action: Use kill with the process ID.

5. kill

* Scenario: You want to terminate a process with PID 1234.
  + Question: What command would you use?
    - Expected Command:

kill 1234

* Scenario: You have a process that is unresponsive and need to force it to terminate.
  + Question: How would you do that?
    - Expected Command:

kill -9 1234

* Scenario: You want to terminate all instances of a process named firefox.
  + Question: What command would you use to kill all firefox processes?
    - Expected Command:

pkill firefox

* Scenario: You have a job running in the background and want to terminate it.
  + Question: How would you find its PID and kill it?
    - Expected Command:

jobs # to get the job ID

kill %job\_id # use job ID to kill

* Scenario: You want to check the status of a process before terminating it.
  + Question: Which command could you use to find its details?
    - Expected Command:

ps -ef | grep <process\_name>

* Scenario: You want to gracefully terminate a process but want to ensure it does so within a timeout.
  + Question: What would be your command?
    - Expected Command:

kill -s TERM 1234

* Scenario: You accidentally killed the wrong process and need to check which process you terminated.
  + Question: What command could you use to check process logs?
    - Expected Command:

dmesg | grep -i kill

* Scenario: You want to kill a process using its name instead of its PID.
  + Question: How would you do that?
    - Expected Command:

pkill process\_name

* Scenario: You want to see if a specific process is running before trying to kill it.
  + Question: What command would you use to check for a running process?
    - Expected Command:

ps aux | grep process\_name

6. nohup

* Scenario: You want to run a long-running script script.sh and ensure it continues running even after you log out.
  + Question: What command would you use?
    - Expected Command:

nohup bash script.sh &

* Scenario: You have a command that takes a long time to execute, and you want to avoid being interrupted if you close your terminal.
  + Question: How would you run that command?
    - Expected Command:

nohup <command> &

* Scenario: You want to run a process in the background and redirect its output to a file named output.log.
  + Question: How would you achieve that?
    - Expected Command:

nohup <command> > output.log 2>&1 &

* Scenario: You have a script that you want to run and log its output to a file while allowing it to run even if you log out.
  + Question: What would the command look like?
    - Expected Command:

nohup ./my\_script.sh > my\_script.log &

* Scenario: You need to check the output of a process that you ran with nohup.
  + Question: Where would you look for the output?
    - Expected Output: Check the nohup.out file in the current directory.
* Scenario: You want to run a Python script that should keep running after logout.
  + Question: What command do you use for that?
    - Expected Command:

nohup python my\_script.py &

* Scenario: You started a job with nohup and want to verify it's running.
  + Question: What command would you use?
    - Expected Command:

ps aux | grep <command>

* Scenario: You forgot to use nohup when starting a process and closed your terminal. How would you ensure the process runs in the background now?
  + Question: What would you do?
    - Expected Action: Restart the process with nohup.
* Scenario: You want to prevent a command from receiving the hangup signal when the terminal closes.
  + Question: Which command would you use?
    - Expected Command:

nohup command &

* Scenario: You wish to terminate a nohup process gracefully.
  + Question: How would you find its PID and kill it?
    - Expected Command:

ps aux | grep command # Find PID

kill PID

7. strace

* Scenario: You want to trace the system calls made by a process with PID 1234.
  + Question: What command would you use?
    - Expected Command:

strace -p 1234

* Scenario: You want to run a command and see its system calls in real-time.
  + Question: How would you do that?
    - Expected Command:

strace <command>

* Scenario: You need to save the output of strace to a file for later analysis.
  + Question: What command would you use to log the output?
    - Expected Command:

strace -o output.txt <command>

* Scenario: You want to trace the file access of a command.
  + Question: Which option would you use with strace?
    - Expected Command:

strace -e trace=file <command>

* Scenario: You want to see only the system calls that return an error.
  + Question: How would you do that using strace?
    - Expected Command:

strace -e trace=%error <command>

* Scenario: You are debugging a script and want to see all the syscalls made by it.
  + Question: What command would you use?
    - Expected Command:

strace -f -e trace=all ./my\_script.sh

* Scenario: You want to follow the child processes created by a command.
  + Question: What option would you use?
    - Expected Command:

strace -f <command>

* Scenario: You want to view the duration of each system call made by a process.
  + Question: Which option would you include?
    - Expected Command:

strace -tt <command>

* Scenario: You want to trace a process that requires root privileges.
  + Question: How would you run it?
    - Expected Command:

sudo strace -p <PID>

* Scenario: You are troubleshooting a program that hangs. You want to see where it gets stuck.
  + Question: How can strace help in this situation?
    - Expected Command:

strace -p <PID>

8. pgrep

* Scenario: You want to find the PID of a running process named apache2.
  + Question: What command would you use?
    - Expected Command:

pgrep apache2

* Scenario: You need to find the PIDs of processes that match a specific pattern, such as those containing java.
  + Question: What command would you use?
    - Expected Command:

pgrep -l java

* Scenario: You want to search for processes owned by the user devuser.
  + Question: How would you do that?
    - Expected Command:

pgrep -u devuser

* Scenario: You want to get detailed information about a process with the name mysqld.
  + Question: Which command would you use to find the PID and then get details?
    - Expected Command:

pgrep mysqld | xargs ps -fp

* Scenario: You want to count how many instances of bash are currently running.
  + Question: What command would you use?
    - Expected Command:

pgrep -c bash

* Scenario: You need to find the PID of a process started by a specific user, devuser, and that contains python in its name.
  + Question: What command would you use?
    - Expected Command:

pgrep -u devuser python

* Scenario: You want to find processes matching a pattern, but ignore case sensitivity.
  + Question: How would you perform this search?
    - Expected Command:

pgrep -i pattern

* Scenario: You want to list processes with their PIDs and names, but only those that are running.
  + Question: What command would you use?
    - Expected Command:

pgrep -l <pattern>

* Scenario: You want to use pgrep with a specific signal to send to a process.
  + Question: How would you send a TERM signal to a process with a name myapp?
    - Expected Command:

kill -TERM $(pgrep myapp)

9. pkill

* Scenario: You want to terminate all processes with the name httpd.
  + Question: What command would you use?
    - Expected Command:

pkill httpd

* Scenario: You need to send a SIGKILL signal to all instances of my\_script.sh.
  + Question: How would you do that?
    - Expected Command:

pkill -9 my\_script.sh

* Scenario: You want to kill processes by user, specifically all processes owned by devuser.
  + Question: What command would you use?
    - Expected Command:

pkill -u devuser

* Scenario: You need to find and terminate processes that match a regex pattern, say all Java processes.
  + Question: What command would accomplish that?
    - Expected Command:

pkill -f java

* Scenario: You want to confirm which processes will be killed before executing the command.
  + Question: What command would you use to preview before killing?
    - Expected Command:

pkill -n <process\_name>

* Scenario: You have a process called backup.sh running, and you want to terminate it without asking for confirmation.
  + Question: What command would you use?
    - Expected Command:

pkill backup.sh

* Scenario: You want to send a specific signal, such as SIGSTOP, to pause a running process named myapp.
  + Question: What command would you use to pause it?
    - Expected Command:

pkill -STOP myapp

* Scenario: You accidentally started two instances of a process and want to kill one but not the other.
  + Question: What command would you use to target a specific instance?
    - Expected Command:

pkill -SIGTERM -f <specific\_command>

* Scenario: You want to terminate a process that is running in the background but forget its name.
  + Question: How would you view running processes and then use pkill?
    - Expected Command:

ps aux | grep <partial\_process\_name> # Find the name

pkill <name>

* Scenario: You need to kill all instances of a process while ensuring you handle the kill signal gracefully.
  + Question: What command would you use for this purpose?
    - Expected Command:

pkill -SIGTERM <process\_name>