UNIX

About UNIX:

UNIX is a powerful, multiuser, multitasking operating system designed for portability, efficiency and flexibility. It was originally developed in the in 1969-1970 at AT&T Bell Labs by Ken Thompson and Dennis Ritchie. It was initially implemented on a PDP-7 minicomputer.

(PDP-7 minicomputer, developed by Digital Equipment corporation (DEC) in 1965, was an early, compact computer system that provided a cost-effective alternative to larger mainframes and played a significant role in the development of the original UNIX operating system.)

Uses of UNIX:

* **Servers and Data Centers:**
* Many servers, especially in data centers, run on UNIX-like operating systems such as Linux and BSD. UNIX systems are known for their stability and performance in handling server tasks.
* **Web Hosting:**
* UNIX-based systems are commonly used for web hosting. Linux, in particular, is popular for hosting web servers and applications due to its reliability and open-source nature.
* **Cloud Computing:**
  + UNIX-like systems are prevalent in cloud computing environments. Many cloud platforms use Linux-based virtual machines and containers (e.g., Docker).
* **Embedded Systems:**
  + UNIX principles are applied in various embedded systems, including routers, network appliances, and other specialized hardware that requires a stable and efficient operating environment.
* **Development Environments:**
  + UNIX-like systems provide a powerful development environment for software engineers, especially those working with open-source projects, system-level programming, or network applications.
* **Scientific and Research Computing:**
  + UNIX systems are used extensively in scientific computing and research due to their stability, scalability, and support for high-performance computing tasks.
* **Educational Institutions:**
  + UNIX and its derivatives are often used in educational settings to teach operating system concepts, computer science fundamentals, and programming skills.

Basic UNIX Commands:

Certainly! Here’s a list of basic UNIX commands with a brief explanation for each:

➡️ File and Directory Management:

- `ls`: List directory contents.

- `cd`: Change directory.

- `pwd`: Print working directory.

- `mkdir`: Create a new directory.

- `rmdir`: Remove an empty directory.

- `rm`: Remove files or directories.

- `cp`: Copy files or directories.

- `mv`: Move or rename files or directories.

- `touch`: Create an empty file or update file timestamps.

➡️ Viewing and Editing Files:

- `cat`: Concatenate and display file contents.

- `more`: View file contents one screen at a time.

- `less`: View file contents with forward and backward navigation.

- `head`: Display the beginning of a file.

- `tail`: Display the end of a file.

➡️ File Permissions and Ownership:

- `chmod`: Change file permissions.

- `chown`: Change file owner and group.

- `chgrp`: Change group ownership of a file.

➡️ System Information:

- `df`: Report disk space usage.

- `du`: Estimate file space usage.

- `top`: Display real-time system processes.

- `ps`: Display a snapshot of current processes.

- `uptime`: Show system uptime.

- `who`: Show who is logged in.

➡️ Networking:

- `ping`: Check connectivity to a host.

- `ifconfig` (or `ip a`): Display or configure network interfaces.

- `netstat`: Display network connections and statistics.

➡️ Searching and Filtering:

- `grep`: Search for patterns within files.

- `find`: Search for files and directories.

➡️ File Compression and Archiving:

- `tar`: Archive files and directories.

- `gzip`: Compress files.

- `gunzip`: Decompress files.

➡️ File Linking:

- `ln`: Create hard and symbolic links.

➡️ Process Management:

- `kill`: Terminate processes.

- `bg`: Resume suspended processes in the background.

- `fg`: Bring a background process to the foreground.

➡️ Miscellaneous:

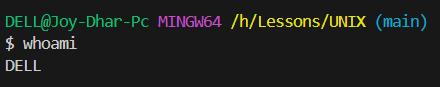
- `echo`: Display a line of text.

- `date`: Display or set the system date and time.

- `man`: Display manual pages for commands.

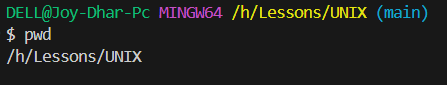
These commands form the core set of tools for managing and interacting with a UNIX-based system.

**Whoami**: it will show the username

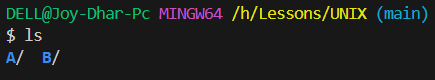


Man: manual

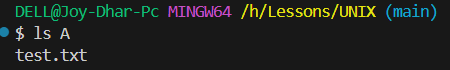
PWD: Print Working Directory

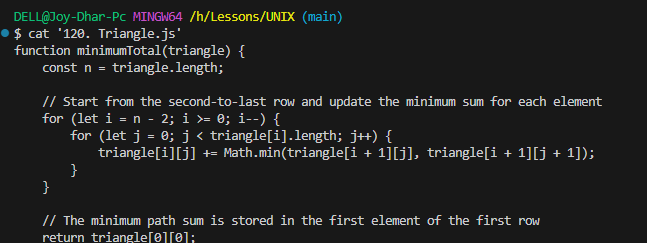


LS: List of all file and folder of current directory



Showing all properties inside folder A



Read File: cat filename.extension  


Read File & Search for a Word: less filename after that f /keyword

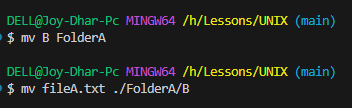
Create File: touch fileA.txt  


Rename a File or Directory



Move Folder or File

Answer: The mv command in UNIX is used to move or rename files and directories.

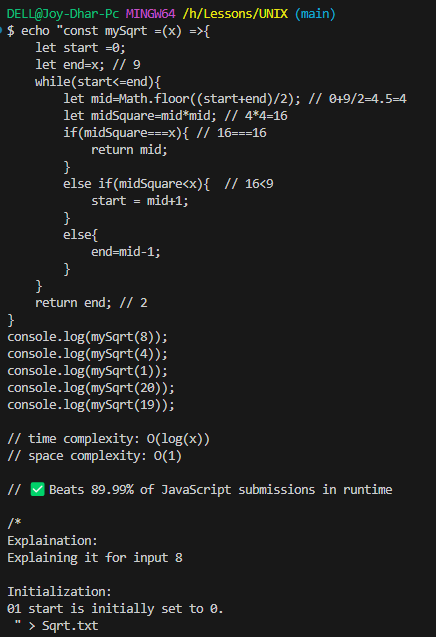


**Grep**: The **grep** command is a powerful tool in UNIX and UNIX-like systems used to search for specific patterns within files or input streams. It stands for "**Global Regular Expression Print**." grep searches through text using regular expressions (patterns) and outputs lines that match the pattern. grep

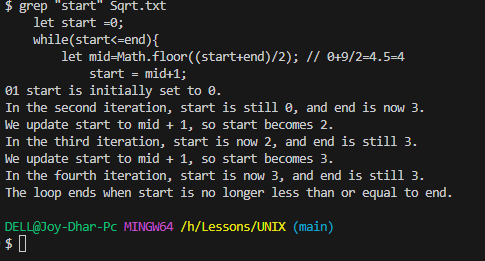
**[options] pattern [file...]**

* -v: Invert match (show lines that do not match the pattern).
* -r or -R: Recursively search directories.
* -l: Print the names of files with matching lines.
* -n: Show line numbers with matching lines.

1. Creating a file name sqrt.txt



Search start from the file



Piping in UNIX and UNIX-like systems allows you to pass the output of one command as the input to another command. This is done using the pipe operator (`|`). Piping is useful for chaining commands together to perform complex operations in a single line of command.

### Basic Syntax

```bash

command1 | command2

```

- `command1`: The command whose output will be used as the input for `command2`.

- `command2`: The command that processes the input received from `command1`.

### Example

Suppose you want to find the number of lines in a file that contain a specific pattern. You can combine the `grep` and `wc` (word count) commands using a pipe.

1. \*\*Using `grep` to search for a pattern and `wc` to count lines:\*\*

```bash

grep "search\_term" filename.txt | wc -l

```

- `grep "search\_term" filename.txt`: Searches for lines containing "search\_term" in `filename.txt`.

- `wc -l`: Counts the number of lines in the output.

This command will output the number of lines in `filename.txt` that contain "search\_term".

### Another Example

You can use piping to sort and display the top 10 entries of a file:

1. \*\*Using `sort` and `head` commands:\*\*

```bash

cat filename.txt | sort | head -n 10

```

- `cat filename.txt`: Displays the content of `filename.txt`.

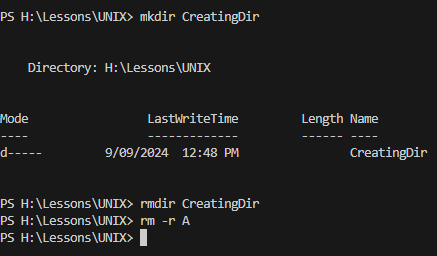
- `sort`: Sorts the lines of the file.

- `head -n 10`: Shows the first 10 lines of the sorted output.

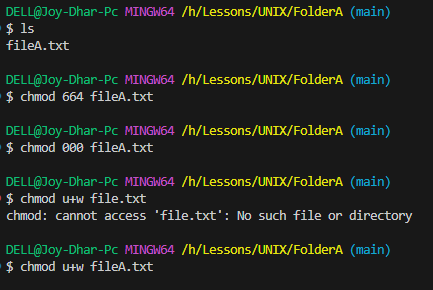
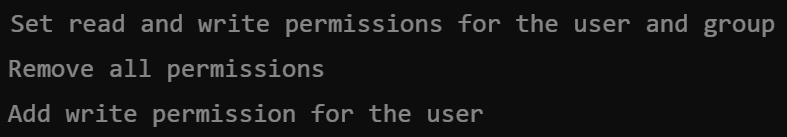
This command will display the first 10 lines of the sorted content of `filename.txt`.

Piping is a powerful feature that allows you to combine simple commands to perform more complex tasks efficiently.

Questions and Answers:  
Q 1: How do you create a directory in UNIX, and how can you remove it?



Q2: How can you change the permissions of a file in UNIX?

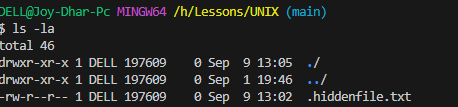


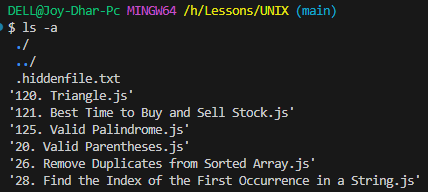
Q3: How do you recursively delete files and directories in UNIX?



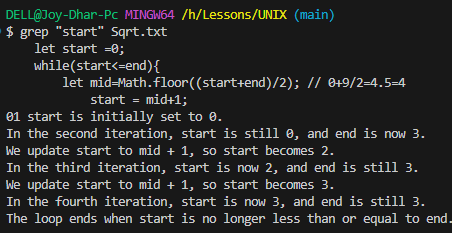
Q4 : How do you list all files in a directory, including hidden files?

Answer: The -a option shows hidden files, and -l provides detailed information about the files.





Q5: How can you search for a pattern in a file using the grep command in UNIX?



Q6: How do you change the permissions of all files in a directory to be readable by everyone?

Answer: You can use the following chmod command to change the permissions of all files to be readable by everyone

**chmod a=r \***

This will apply read permission (r) to all (a) users.

Q7: What command is used to display the current working directory in UNIX?

Answer: Using PWD (Print Working Directory)



Q8: How do you redirect the output of a command to a file in UNIX?

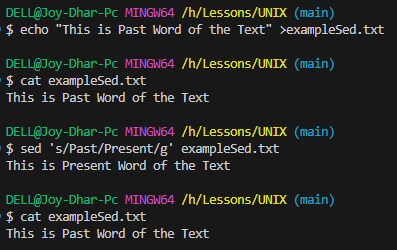
Answer: You can use the > operator to redirect the output of a command to a file. For example:



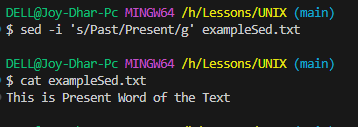
Q9: How do you use the **sed** command to replace text in a file?

Without modification the main text

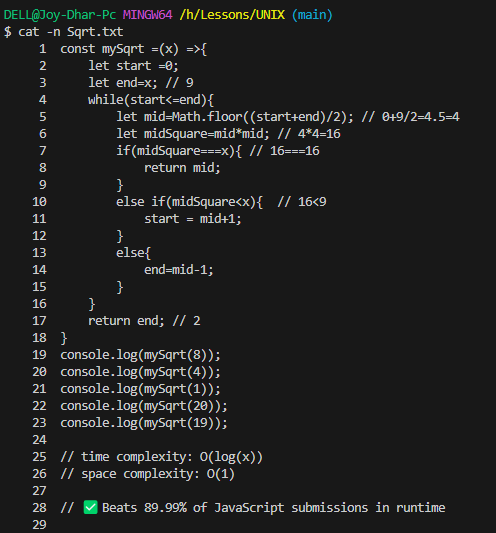
**sed 's/old\_text/new\_text/g' filename**



Modify the main text Need to place **-I command**



Q10: How do you display the contents of a file with line numbers in UNIX?



Certainly! Here are 10 more advanced UNIX questions based on the previous topics:

---

\*\*Q1: How can you use `grep` to search for a pattern across multiple files and display the filename and line number?\*\*

\*\*Answer:\*\*

You can use `grep` with the `-H` (filename) and `-n` (line number) options to search across multiple files:

```bash

grep -H -n "pattern" file1.txt file2.txt

```

---

\*\*Q2: How do you use `sed` to delete lines matching a specific pattern from a file?\*\*

\*\*Answer:\*\*

To delete lines containing a specific pattern using `sed`, use the following command:

```bash

sed '/pattern/d' filename

```

This command removes all lines containing "pattern" from `filename`.

---

\*\*Q3: How can you move a file to a new location and also rename it in a single command?\*\*

\*\*Answer:\*\*

You can use `mv` to move and rename a file in one command:

```bash

mv /path/to/oldfile.txt /newpath/newfile.txt

```

---

\*\*Q4: How do you use `find` to locate files modified in the last 7 days and delete them?\*\*

\*\*Answer:\*\*

To find and delete files modified in the last 7 days, use:

```bash

find /path/to/directory -type f -mtime -7 -exec rm {} +

```

---

\*\*Q5: How can you use `xargs` with `find` to execute a command on the files found?\*\*

\*\*Answer:\*\*

You can use `xargs` to execute a command on each file found by `find`:

```bash

find /path/to/directory -type f -name "\*.log" | xargs rm

```

This command finds all `.log` files and deletes them.

---

\*\*Q6: How can you use `tar` to create a compressed archive of a directory?\*\*

\*\*Answer:\*\*

To create a compressed archive of a directory using `tar`, use:

```bash

tar -czvf archive.tar.gz /path/to/directory

```

- `c` creates an archive

- `z` compresses with gzip

- `v` verbose output

- `f` specifies the filename

---

\*\*Q7: How do you schedule a command to run at a specific time using `cron`?\*\*

\*\*Answer:\*\*

To schedule a command using `cron`, edit the crontab with:

```bash

crontab -e

```

Then add a line like:

```bash

0 5 \* \* \* /path/to/command

```

This schedules the command to run daily at 5 AM.

---

\*\*Q8: How can you change the ownership of a file to a specific user and group?\*\*

\*\*Answer:\*\*

You can change the ownership of a file using `chown`:

```bash

chown user:group filename

```

This changes the owner to `user` and the group to `group`.

---

\*\*Q9: How do you use `grep` to perform a case-insensitive search and display only matched text?\*\*

\*\*Answer:\*\*

To perform a case-insensitive search and display only matched text, use:

```bash

grep -i -o "pattern" filename

```

- `-i` makes the search case-insensitive

- `-o` displays only the matched parts

---

\*\*Q10: How can you use `awk` to print the first and third columns of a file separated by a comma?\*\*

\*\*Answer:\*\*

You can use `awk` to print specific columns and separate them with a comma:

```bash

awk '{print $1 "," $3}' filename

```

---

These questions involve advanced usage of UNIX commands and utilities, focusing on file management, searching, scheduling, and text processing.

Sure! Here are the next 10 advanced UNIX questions and their answers:

---

\*\*Q1: How can you use `grep` to search for a pattern and also exclude lines containing another pattern?\*\*

\*\*Answer:\*\*

You can use `grep` with multiple patterns and the `-v` option to exclude lines containing another pattern:

```bash

grep "pattern1" filename | grep -v "pattern2"

```

This command finds lines containing `pattern1` but excludes those containing `pattern2`.

---

\*\*Q2: How do you use `sed` to replace text only on specific lines in a file?\*\*

\*\*Answer:\*\*

You can specify line numbers or ranges with `sed` to replace text only on those lines:

```bash

sed '2,4s/old\_text/new\_text/' filename

```

This command replaces `old\_text` with `new\_text` only on lines 2 through 4.

---

\*\*Q3: How can you use `find` to locate files larger than a specific size and move them to another directory?\*\*

\*\*Answer:\*\*

You can use `find` with `-size` and `-exec` to locate and move files:

```bash

find /path/to/directory -type f -size +100M -exec mv {} /path/to/destination/ \;

```

This finds files larger than 100MB and moves them to `/path/to/destination/`.

---

\*\*Q4: How do you use `tar` to extract files from a specific directory within an archive?\*\*

\*\*Answer:\*\*

To extract files from a specific directory within a tar archive, use:

```bash

tar -xzvf archive.tar.gz directory/

```

This extracts only the contents of `directory` from the archive.

---

\*\*Q5: How can you use `xargs` to limit the number of processes started at once?\*\*

\*\*Answer:\*\*

You can use the `-P` option with `xargs` to limit the number of processes:

```bash

find /path/to/directory -type f | xargs -P 4 -I {} command {}

```

This runs `command` on up to 4 files in parallel.

---

\*\*Q6: How do you append the output of a command to an existing file?\*\*

\*\*Answer:\*\*

To append the output of a command to an existing file, use the `>>` redirection operator:

```bash

command >> filename

```

This appends the output of `command` to `filename`.

---

\*\*Q7: How can you use `chmod` to set permissions for a file using symbolic notation?\*\*

\*\*Answer:\*\*

You can set file permissions using symbolic notation with `chmod`:

```bash

chmod u+x file.txt # Add execute permission for the user

chmod g-w file.txt # Remove write permission for the group

chmod o=r file.txt # Set read-only permission for others

```

---

\*\*Q8: How do you use `grep` to search for multiple patterns within a file and print only matching lines?\*\*

\*\*Answer:\*\*

You can use `grep` with the `-e` option to specify multiple patterns:

```bash

grep -e "pattern1" -e "pattern2" filename

```

This prints lines containing either `pattern1` or `pattern2`.

---

\*\*Q9: How can you use `awk` to calculate the average of a numeric column in a file?\*\*

\*\*Answer:\*\*

You can use `awk` to calculate the average of a column:

```bash

awk '{sum += $1} END {if (NR > 0) print sum / NR}' filename

```

This calculates the average of the first column in `filename`.

---

\*\*Q10: How do you use `find` to locate files that have not been accessed in the last 30 days?\*\*

\*\*Answer:\*\*

To find files that have not been accessed in the last 30 days, use:

```bash

find /path/to/directory -type f -atime +30

```

Certainly! Here are 10 more advanced UNIX questions and their answers:

---

\*\*Q1: How can you use `grep` to search for a pattern in a file and also show the line number and file name for each match?\*\*

\*\*Answer:\*\*

To search for a pattern and include the line number and file name, use:

```bash

grep -H -n "pattern" filename

```

- `-H` includes the filename.

- `-n` includes the line number.

---

\*\*Q2: How do you use `sed` to insert a line of text after a specific line number in a file?\*\*

\*\*Answer:\*\*

To insert a line after a specific line number, use:

```bash

sed '3a\New line of text' filename

```

This inserts "New line of text" after line 3 in `filename`.

---

\*\*Q3: How can you use `find` to search for files that contain a specific string in their content?\*\*

\*\*Answer:\*\*

You can combine `find` with `grep` to search for files containing a specific string:

```bash

find /path/to/directory -type f -exec grep -l "search\_string" {} +

```

- `-l` lists filenames containing the string.

---

\*\*Q4: How do you use `tar` to create an archive of multiple directories and exclude specific files?\*\*

\*\*Answer:\*\*

To create an archive and exclude files, use:

```bash

tar --exclude='\*.log' -czvf archive.tar.gz dir1 dir2

```

This creates a compressed archive of `dir1` and `dir2`, excluding `.log` files.

---

\*\*Q5: How can you use `xargs` to execute a command on files with specific extensions in a directory?\*\*

\*\*Answer:\*\*

To execute a command on files with a specific extension, use:

```bash

find /path/to/directory -type f -name "\*.txt" | xargs command

```

This runs `command` on all `.txt` files found.

---

\*\*Q6: How do you use `chmod` to set the permissions for a file so that only the owner can read and write, and others have no permissions?\*\*

\*\*Answer:\*\*

To set permissions so that only the owner can read and write, use:

```bash

chmod 600 filename

```

- `6` gives read and write permissions to the owner.

- `0` gives no permissions to the group and others.

---

\*\*Q7: How can you use `awk` to extract and sum values from the second column of a file?\*\*

\*\*Answer:\*\*

To extract and sum values from the second column, use:

```bash

awk '{sum += $2} END {print sum}' filename

```

This sums all values in the second column of `filename`.

---

\*\*Q8: How do you use `grep` to search for lines matching a pattern but not include lines that contain another pattern?\*\*

\*\*Answer:\*\*

You can use `grep` with negative lookaheads to exclude lines with another pattern:

```bash

grep "pattern1" filename | grep -v "pattern2"

```

This finds lines with `pattern1` but excludes those with `pattern2`.

---

\*\*Q9: How can you use `find` to search for files by modification time within the last 24 hours?\*\*

\*\*Answer:\*\*

To find files modified in the last 24 hours, use:

```bash

find /path/to/directory -type f -mtime -1

```

- `-mtime -1` finds files modified within the last day.

---

\*\*Q10: How do you use `sed` to replace text in a file only if the line contains a specific pattern?\*\*

\*\*Answer:\*\*

To replace text only if the line contains a specific pattern, use:

```bash

sed '/pattern/s/old\_text/new\_text/' filename

```

This replaces `old\_text` with `new\_text` only on lines containing `pattern`.

Certainly! Here are 10 more advanced UNIX questions and their answers:

---

\*\*Q1: How can you use `grep` to search for a pattern in a file and also show the line number and file name for each match?\*\*

\*\*Answer:\*\*

To search for a pattern and include the line number and file name, use:

```bash

grep -H -n "pattern" filename

```

- `-H` includes the filename.

- `-n` includes the line number.

---

\*\*Q2: How do you use `sed` to insert a line of text after a specific line number in a file?\*\*

\*\*Answer:\*\*

To insert a line after a specific line number, use:

```bash

sed '3a\New line of text' filename

```

This inserts "New line of text" after line 3 in `filename`.

---

\*\*Q3: How can you use `find` to search for files that contain a specific string in their content?\*\*

\*\*Answer:\*\*

You can combine `find` with `grep` to search for files containing a specific string:

```bash

find /path/to/directory -type f -exec grep -l "search\_string" {} +

```

- `-l` lists filenames containing the string.

---

\*\*Q4: How do you use `tar` to create an archive of multiple directories and exclude specific files?\*\*

\*\*Answer:\*\*

To create an archive and exclude files, use:

```bash

tar --exclude='\*.log' -czvf archive.tar.gz dir1 dir2

```

This creates a compressed archive of `dir1` and `dir2`, excluding `.log` files.

---

\*\*Q5: How can you use `xargs` to execute a command on files with specific extensions in a directory?\*\*

\*\*Answer:\*\*

To execute a command on files with a specific extension, use:

```bash

find /path/to/directory -type f -name "\*.txt" | xargs command

```

This runs `command` on all `.txt` files found.

---

\*\*Q6: How do you use `chmod` to set the permissions for a file so that only the owner can read and write, and others have no permissions?\*\*

\*\*Answer:\*\*

To set permissions so that only the owner can read and write, use:

```bash

chmod 600 filename

```

- `6` gives read and write permissions to the owner.

- `0` gives no permissions to the group and others.

---

\*\*Q7: How can you use `awk` to extract and sum values from the second column of a file?\*\*

\*\*Answer:\*\*

To extract and sum values from the second column, use:

```bash

awk '{sum += $2} END {print sum}' filename

```

This sums all values in the second column of `filename`.

---

\*\*Q8: How do you use `grep` to search for lines matching a pattern but not include lines that contain another pattern?\*\*

\*\*Answer:\*\*

You can use `grep` with negative lookaheads to exclude lines with another pattern:

```bash

grep "pattern1" filename | grep -v "pattern2"

```

This finds lines with `pattern1` but excludes those with `pattern2`.

---

\*\*Q9: How can you use `find` to search for files by modification time within the last 24 hours?\*\*

\*\*Answer:\*\*

To find files modified in the last 24 hours, use:

```bash

find /path/to/directory -type f -mtime -1

```

- `-mtime -1` finds files modified within the last day.

---

\*\*Q10: How do you use `sed` to replace text in a file only if the line contains a specific pattern?\*\*

\*\*Answer:\*\*

To replace text only if the line contains a specific pattern, use:

```bash

sed '/pattern/s/old\_text/new\_text/' filename

```

This replaces `old\_text` with `new\_text` only on lines containing `pattern`.

---

These questions cover more advanced aspects of file management, text processing, and command-line utilities in UNIX.