**Assignment 2:**

# **MTMC-509 Programming Lab-I**

**Searching Commands, File Commands**

**Searching Commands**

 **Basic File Search with find**

* Use the find command to search for all files with the extension .txt in your home directory.  
  Command: find [path] -name "\*.txt"
* Record the names of the files found and the directories they are located in.

 **Search for Large Files**

* Use the find command to search for files larger than 100MB in the /var directory.  
  Command: find [path] -size +100M
* List the names and sizes of the files found. How many large files were found in the directory?

 **Searching for Text in Files with grep**

* Create a text file named logfile.txt with some sample content. Ensure the file contains multiple lines, including the word "ERROR" in a few of them.
* Use the grep command to search for the word "ERROR" in logfile.txt.  
  Command: grep [search\_pattern] [file\_name]
* Record the lines containing the word "ERROR".

 **Recursive Search with grep**

* Create a directory named Logs and move logfile.txt into it. Create two additional text files in the directory, each containing some sample content with varying occurrences of the word "WARNING".
* Use the grep -r command to search recursively for the word "WARNING" within the Logs directory.  
  Command: grep -r [search\_pattern] [directory\_name]
* List all occurrences of the word "WARNING" in the directory.

 **File Location with locate**

* Use the locate command to find all files and directories related to the name passwd.  
  Command: locate [name]
* Describe the type of files and directories that were found. Are they system files or user files?

 **Command Path with which and whereis**

* Use the which command to find the path to the python3 executable.  
  Command: which [command]
* Use the whereis command to find the source, binary, and manual page locations for python3.  
  Command: whereis [command]
* Compare the output of which and whereis. Which command provides more detailed information?

 **Find and Replace with sed**

* Use the sed command to replace the word "ERROR" with "WARNING" in the logfile.txt file.  
  Command: sed 's/[old\_text]/[new\_text]/' [file\_name]
* Display the content of logfile.txt after the changes. Did all occurrences of "ERROR" get replaced with "WARNING"?

**File Commands**

 **Directory Creation and File Operations**

* Create a new directory called Projects in your home directory using the mkdir command.
* Inside the Projects directory, create three empty text files: fileA.txt, fileB.txt, and fileC.txt using the touch command.
* Open fileA.txt using the nano text editor and write the following lines:

css

Copy code

This is file A.

It contains some sample content.

* Save and exit the file.

 **Copying and Moving Files**

* Copy the contents of fileA.txt to a new file named fileD.txt inside the Projects directory using the cp command.
* Move fileB.txt to a new subdirectory called Archives inside Projects. If Archives does not exist, create it first using mkdir.

 **Recursively Copying Directories**

* Copy the entire Archives directory and its contents to a new directory called Backup inside the Projects directory using the cp -r command.
* Verify that the Backup directory contains the same files as Archives.

 **Displaying File Content**

* Use the cat command to display the contents of fileA.txt.
* Use the head command to display the first ten lines of fileA.txt.
* Use the tail command to display the last ten lines of fileA.txt.
* Use the more command to view the contents of fileD.txt page by page.
* Use the less command to navigate through the contents of fileA.txt.

 **Appending and Comparing Files**

* Append the content of fileC.txt to fileA.txt using the cat [source\_file] >> [destination\_file] command.
* Use the diff command to compare fileA.txt and fileD.txt and display any differences between the two files.

 **Creating Symbolic Links**

* Create a symbolic link to fileA.txt named link\_to\_A inside the Projects directory using ln -s.
* Verify that the symbolic link works by using cat to display the contents of link\_to\_A.

 **Text Processing with Word Count and Cut**

* Use the wc -w command to count the number of words, lines, and bytes in fileA.txt.
* Create a new text file named data.csv with the following content:

Copy code:

*echo "Name, Age, Location" > people.csv*

*echo "John, 30, New York" >> people.csv*

*echo "Alice, 25, Los Angeles" >> people.csv*

*echo "Bob, 22, Chicago" >> people.csv*

Name, Age, Location

John, 30, New York

Alice, 25, Los Angeles

Bob, 22, Chicago

* Use the cut -d ',' -f 1 people.csv command to extract and display the "Name" column from the CSV file.

 **Encrypting and Decrypting Files**

* Encrypt fileA.txt using the gpg -c command, which will prompt you for a passphrase.
* Decrypt the encrypted file (fileA.txt.gpg) back to its original form using the gpg command.

 **Secure Deletion**

* Use the shred -u command to securely overwrite and delete fileC.txt.  
  **(Note: This will permanently destroy the file.)**

 **Executing Scripts and Redirection**

* Create a new script file named example.sh with the following content:

bash

Copy code

#!/bin/bash

echo "Hello, this is a test script!"

* Make the script executable and run it using the source command.
* Use the ls command and store its output in a file named file\_list.txt using the tee command. Suppress terminal output during the operation.

 **Advanced File Operations**

* List the number of lines, words, and characters in each file in the Projects directory using the ls | xargs wc command.