**CDAC MUMBAI**

Concepts of Operating System

**Assignment 1**

**Problem 1:** **Read the instructions carefully and answer accordingly. If there is any need to insert some data then do that as well.**

**a) Navigate and List:**

a. Start by navigating to your home directory and list its contents. Then, move into a directory named "LinuxAssignment" if it exists; otherwise, create it.

-> cd~ (to go to home directory)

-> ls (to list contents)

-> mkdir -p LinuxAssignment (creating directory)

-> mv LinuxAssignment /

**b) File Management:**

a. Inside the "LinuxAssignment" directory, create a new file named "file1.txt". Display its contents.

->touch file1.txt (created file)

->echo " Hello...I'm Rugvedi!" (printed content)

->cat file1.txt (to show contents)

c) **Directory Management:**

a. Create a new directory named "docs" inside the "LinuxAssignment" directory.

-> mkdir LinuxAssignment/docs

**d) Copy and Move Files:**

a. Copy the "file1.txt" file into the "docs" directory and rename it to "file2.txt".

-> cp file1.txt (copy file)

-> docs/file2.txt (renaming it)

**e) Permissions and Ownership:**

a. Change the permissions of "file2.txt" to allow read, write, and execute permissions for the owner and only read permissions for others. Then, change the owner of "file2.txt" to the current user.

-> chmod 744 docs/file2.txt (744 – binary nums addition )

-> chown $USER docs/file2.txt ($USER – holds recent logged in data)

**f) Final Checklist:**

a. Finally, list the contents of the "LinuxAssignment" directory and the root directory to ensure that all operations were performed correctly.

-> ls LinuxAssignment

-> ls /

**g) File Searching:**

a. Search for all files with the extension ".txt" in the current directory and its subdirectories.

-> find -name \*.txt” (searchin)

b. Display lines containing a specific word in a file (provide a file name and the specific word to search).

-> grep “world” file2.txt (to search particular word)

**h) System Information:**

a. Display the current system date and time.

->date (we use date to display current date as well as time)

**i) Networking:**

a. Display the IP address of the system.

-> hostname -I (hostname = prints system’s hostname or info)

b. Ping a remote server to check connectivity (provide a remote server address to ping).

-> ping -c 5 Twitter.com( const num is discovering the no.of packets)

**j) File Compression:**

a. Compress the "docs" directory into a zip file.

-> zip -r docs.zip docs (r stands to include all files and folders)

b. Extract the contents of the zip file into a new directory.

-> unzip docs.zip -d newdocs (extracts into new directory = -d and newdocs means into newdocs)

**k) File Editing:**

a. Open the "file1.txt" file in a text editor and add some text to it.

->nano file1.txt (nano text editor)

->echo “I’m adding the text” >>file1.txt

b. Replace a specific word in the "file1.txt" file with another word (provide the original word and the word to replace it with).

-> sed -I 's/Hello/Rugvedi/g' file1.txt (hello – oldword, Rugvedi – newword)

(sed – editor to replace or find txt, -I -changes directly in files, s – substitute and g – globally replace word)

**Problem 2: Read the instructions carefully and answer accordingly. If there is any need to insert some data then do that as well.**

a. Suppose you have a file named "data.txt" containing important information. Display the first 10 lines of this file to quickly glance at its contents using a command.

-> head -n 10 data.txt ( head – top lines, n is for num)

b. Now, to check the end of the file for any recent additions, display the last 5 lines of "data.txt" using another command.

-> tail -n 5 data.txt (tail – last lines)

c. In a file named "numbers.txt," there are a series of numbers. Display the first 15 lines of this file to analyze the initial data set.

-> head -n 15 numbers.txt

d. To focus on the last few numbers of the dataset, display the last 3 lines of "numbers.txt".  
-> tail -n 3 numbers.txt

e. Imagine you have a file named "input.txt" with text content. Use a command to translate all lowercase letters to uppercase in "input.txt" and save the modified text in a new file named "output.txt."

-> tr ‘a-z’ to ‘A-Z’<input.txt> output.txt (taking i/p from input.tx to out this i/p into output.txt)

f. In a file named "duplicate.txt," there are several lines of text, some of which are duplicates. Use a command to display only the unique lines from "duplicate.txt."

-> uniq duplicate.txt (uniq – removes duplicate lines)

g. In a file named "fruit.txt," there is a list of fruits, but some fruits are repeated. Use a command to display each unique fruit along with the count of its occurrences in "fruit.txt."

->sort fruit.txt (sorts file to see duplicates)

-> uniq -c