**OTHER EXTRA TASK ON SHELL.**

**Write a script that prints the absolute path name of the current working directory**.

#!/bin/bash

current\_dir=$(pwd)

echo "Current working directory: $current\_dir"

**Write a script that changes the working directory to the user’s home directory.**

#!/bin/bash

cd ~

echo "Changed working directory to: $(pwd)"

**Display current directory contents in a long format**

#!/bin/bash

ls -l

**Display current directory contents, including hidden files (starting with .). Use the long format.**

#!/bin/bash

ls -la

**Display current directory contents.**

* Long format
* with user and group IDs displayed numerically
* And hidden files (starting with .)

**Create a script that creates a directory named my\_first\_directory in the /tmp/ directory.**

#!/bin/sh

# Check if the directory already exists

if [ -d "/tmp/my\_first\_directory" ]; then

echo "The directory /tmp/my\_first\_directory already exists."

else

# Create the directory

mkdir /tmp/my\_first\_directory

echo "The directory /tmp/my\_first\_directory has been created."

fi

**Move the file betty from /tmp/ to /tmp/my\_first\_directory.**

mv /tmp/betty /tmp/my\_first\_directory/

**Write a script that lists all files (even ones with names beginning with a period character, which are normally hidden) in the current directory and the parent of the working directory and the /boot directory (in this order), in long format.**

#!/bin/bash

# List files in the current directory

echo "Current Directory:"

ls -la

# Move to the parent directory

cd ..

# List files in the parent directory

echo -e "\nParent Directory:"

ls -la

# Move to the /boot directory

cd /boot

# List files in the /boot directory

echo -e "\n/boot Directory:"

ls -la

**Write a script that prints the type of the file named iamafile. The file iamafile will be in the /tmp directory when we will run your script.**

#!/bin/bash

file\_path="/tmp/iamafile"

if [ -f "$file\_path" ]; then

file\_type=$(file -b "$file\_path")

echo "The type of the file is: $file\_type"

else

echo "File not found: $file\_path"

fi

**Create a symbolic link to /bin/ls, named \_\_ls\_\_. The symbolic link should be created in the current working directory.**

#!/bin/bash

link\_name="\_\_ls\_\_"

target\_path="/bin/ls"

ln -s "$target\_path" "$link\_name"

echo "Symbolic link created: $link\_name"

**Create a script that copies all the HTML files from the current working directory to the parent of the working directory, but only copy files that did not exist in the parent of the working directory or were newer than the versions in the parent of the working directory.**

#!/bin/bash

current\_dir=$(pwd)

parent\_dir=$(dirname "$current\_dir")

for file in \*.html; do

if [[ ! -e "$parent\_dir/$file" || "$file" -nt "$parent\_dir/$file" ]]; then

cp "$file" "$parent\_dir"

echo "Copied $file to $parent\_dir"

else

echo "$file already exists in $parent\_dir and is newer"

fi

done

**15. Create a script that moves all files beginning with an uppercase letter to the directory /tmp/u.**

**You can assume that the directory /tmp/u will exist when we will run your script**

#!/bin/bash

# Move files beginning with uppercase letters to /tmp/u

for file in [A-Z]\*; do

if [ -f "$file" ]; then

mv "$file" /tmp/u/

echo "Moved file: $file"

fi

done

echo "File movement complete."

**16. Create a script that deletes all files in the current working directory that end with the character ~.**

#!/bin/bash

# Delete files ending with ~

for file in \*~; do

if [ -f "$file" ]; then

rm "$file"

"Deleted file: $file"

fi

done

"File deletion complete."

**17. Create a script that creates the directories welcome/, welcome/to/ and welcome/to/school in the current directory.**

**You are only allowed to use two spaces (and lines) in your script, not more.**

#!/bin/bash

mkdir -p welcome/to/school

echo "Directories created successfully."

**18. Write a command that lists all the files and directories of the current directory, separated by commas (,).**

ls –m

file1.txt, file2.jpg, directory1, directory2, script.sh