Bash

#!/bin/bash

((count=0))

for f in $( ls $1 ); do

if [[ "$f" =~ \.py$ ]]; then

rm "$1/$f"

echo "Deleted" $f >> log.txt ;

((count = count +1))

fi

done

echo $count

Write a bash script to delete all the files having the extension .py in a given directory, and log all these deletions in a file “log.txt” in the same directory as the bash script (NOT the directory where the .py files are to be deleted). Finally print the number of files deleted to the console (NOT to log.txt).

The directory path will be given as an argument to the bash script. Along with deletion of all files, the script should also add the line :- “Deleted <file\_name>” (without quotes) to log.txt.

Usage :- bash submission.sh <path-to-directory>

Eg.

Consider the directory “/home/labDirectory/testfolder” containing files “a.py”, “b.py”, “c.py”,”a.pdf” (quote for emphasis only).

On running the command :- “ bash submission.sh /home/labDirectory/testfolder/”

(inverted commas are not part of the command and meant only for emphasis),

all the .py files in the testfolder directory should be deleted. T he only file remaining in the testfolder directory should be “a.pdf” as it does not have a .py extension.

The log.txt file (created in the same directory as submission.sh) should have the following lines:-

Deleted a.py

Deleted b.py

Deleted c.py

Log each individual file on a new line. Be sure to log ONLY the filename and NOT the whole address of the file (i.e. Do NOT write logs of the form “Deleted /home/labDirectory/testfolder/a.py”).

Hint - The basename command might be useful here, use the man pages to see what it does.

You can log the files in any order. Assume log.txt does not exist before running the command.

The terminal should have the output :-

3

Because 3 files are deleted.

Tip - When locally testing, add the delete command when you are sure that the log.txt and count of files is working properly. Else, you will have the hassle of recreating the deleted files after each local test.