

Jan 2020 CSE 314: Operating Systems Sessional

Offline on Shell Script

Description:

One of your friends from another department is currently struggling with a lot of files. He wants to filter out the text files containing some specific string in the beginning or ending portion of the files from his working directory. His working directory contains lots of different types of files and subdirectories. He wants to gather the required files in a specific format in a separate directory. He desperately needs your help.

Input:

He will give you a directory (with lots of files and sub-directories) and another input file.

The input file will have only three (3) lines.

- In the first line, he will tell you where to search in the files
 - ***begin*** for beginning portion of the files
 - ***end*** for ending portion of the files
- In the second line, he will tell you the **number of lines (an integer)** to consider in the specified portion
- In the third line, he will tell you the **string (case insensitive)** to search

Tasks:

In this assignment, you have to do the following

1. Take the working directory name (optional) and input file name as command line argument
2. If the user does not provide any working directory name, consider your script is lying in the **root working directory**
3. If the user does not provide any input file name, show him a **usage message** (i.e. how to use this script of yours)
4. Read the inputs from the input file (if the input file does not exist, prompt the user to give a valid input file name)
5. In a recursive manner, from the root working directory and all of its sub-directories, find out the **readable text** files(not only .txt files but any ASCII/Unicode text files) which matches the criteria described in the input file
6. Gather all the matched files in a separate output directory (created by you outside the root working directory)

7. Add the full relative path(from the root working directory) to the file as prefix and the line number as suffix to the filename without the extension. Replace the '/' in path with '.'. For clarification, let us assume, you are searching for the string "contribute" and found it in "root/a/b/dummy.txt"
 - a. If you are searching in the beginning 20 lines of the file and found it in 5, 8, and 17th line, then you need to copy dummy into the output directory as **"root.a.b.dummy5.txt"** not just dummy.txt (the first line number where the string was found is added as suffix)
 - b. If you are searching in the last 25 lines of the files and found it in line 55, 60, and 67 where the total number of lines in dummy.txt is 73, then you need to copy dummy into the output directory as **"root.a.b.dummy67.txt"** not just dummy.txt (the last line number where the string was found is added as suffix)
8. Create a .csv file containing three columns, namely filepath, line number and line containing the searched text. Append all the lines with the searched text found in this file.
9. At the end of the script, report the **number of files** which matched the criteria

Output:

You need to give three(3) outputs

- An output directory where all the files are organized as mentioned above
- A .csv file
- Printing the total number of files which matched the criteria

Help:

You may find the *cut*, *sed*, *find* commands useful.

Have a look into shell parameter expansion

If you need help working with csv files using LINUX command:

<https://bconnelly.net/working-with-csvs-on-the-command-line/>

Restrictions:

Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet etc.). Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks in the offline. It is your duty to protect your code.

Disclaimer:

The sample input and output files are simply for clarifying the formats. No guarantee is given that the outputs will be exactly the same for the provided input.

Marks Distribution:

Task(s)	Mark (100)
Handling command line arguments (Task 1, 2, 3)	15
Reading input from files properly (Task 4)	10
Finding proper files (Task 5)	30
Properly organizing the files in the output directory (Task 6, 7)	25
Properly creating the .csv file (Task 8)	15
Giving a summary output (Task 8)	5

Submission Guideline:

1. Create a directory with your 7 digit student id as name
2. Rename your shell script file with your 7 digit student id
3. Put this script into the directory created in 1
4. Zip the directory
5. Upload the zip into moodle

For example, if your student id is 1405123, create a directory named 1405123, rename your .sh file into 1405123.sh. Put 1405123.sh into 1405123. Zip 1405123 into 1405123.zip and upload the 1405123.zip into moodle.

Failure to follow the above mentioned submission guideline will result in some penalty.

Submission Deadline:

March 07, 2020 11:55 PM

Deadline is ***Deadline.***