CS108 - Software Systems Lab

Lab 2 - Advanced Unix Commands

Student: Aditya Sanapala, 23b0912@iitb.ac.in
Lecturer: Kameswari Chebrolu, chebrolu@cse.iitb.ac.in

Problem 1: Copying all files

You have two folders: `input` and `output`. The `input` folder contains files and subfolders, while the `output` folder is empty.
Write a unix command that, when executed, will:

(a) Copy **only** the files from `input` to `output`.

(b) Ensure that any subfolder (empty or non-empty) in `input` is not copied.

(c) Copy files that may be present in any subfolder (or sub-subfolder or sub-sub-subfolder, and so on) of `input` to `output`.

Example: If the `input` folder directory structure is:

```
input
|----- .hidden
|----- report.pdf
|----- subfolder1
|     |----- file1.txt
|     |----- file2.txt
|     |----- file3.pdf
|     |___ sub-subfolder
|          |___ test.sql
|----- subfolder2
|     |----- file4.txt
|     |----- file5.txt
|     |___ file6.docx
|----- test.cpp
|___ test.zip
```

After executing your command line, the `output/` folder directory structure should look like:

```
output
|----- file1.txt
|----- file2.txt
|----- file3.pdf
|----- file4.txt
|----- file5.txt
|----- file6.docx
|----- .hidden
|----- report.pdf
```

```
|----- test.cpp
|___ test.sql
|___ test.zip
```

Notes

1. All file names will be distinct.
2. There may be hidden files (their names begin with .); copy those too.

Problem 2: grep the passed

A list of students is given with their first names, roll numbers, and a comment saying "PASSED" or "FAILED" for each student.
Write a unix command to print the names and roll numbers of those students who have passed, from the `result.csv` file present in the same directory as `submission.sh`. The list of students will be given in `result.csv`.
Example: If the `result.csv` file looks like:

```
Kavya,22B1053,PASSED
Saksham,22B1003,PASSED
Ayush,22B0001,FAILED
Guramrit,22B0002,PASSED
```

After executing your command line, your output should look like:

```
Kavya,22B1053
Saksham,22B1003
Guramrit,22B0002
```

Apart from the `result.csv` in `/home/labDirectory/`, there are 3 more such files present in the `testcases` folder. When you click the `Evaluate` button, the autograder will check the correctness of your `submission.sh` file against the 3 testcases.

Problem 3: Validate using regex

You have a file named `collect.txt` with the following contents:

```
sakshamrathi21 1 22b1003@iitb.ac.in submission.sh
malaikaarora01 3 22b0069@iitb.ac.in submsubission.sh
guramritsingh07 2 22b0001@iitb.ac.in submission.sh
rahulgandhi04 5 22b0010@iitb.ac.in subpappu.sh
nithinkamath10 2 zerodha@iitb.ac.in trade.sh
narendramodi24 4 22b0024@iitb.ac.in submission.sh
kavyagupta11 1 22b1053@iitb.ac.in submission.sh
```

You need to select the valid lines and print them back into `output.txt`. The validity will be verified on the basis of the following conditions:

(a) Every username contains some characters (of arbitrary length ¿ 0, lowercase letters) in the start and then a two digit number at the end (00 to 99).

(b) The next number should be between 1 and 4.

(c) The email address should be of the form `<22b><four digit number><@iitb.ac.in>`.

After executing you unix command, the output should be:

```
sakshamrathi21 1 22b1003@iitb.ac.in submission.sh
guramritsingh07 2 22b0001@iitb.ac.in submission.sh
narendramodi24 4 22b0024@iitb.ac.in submission.sh
kavyagupta11 1 22b1053@iitb.ac.in submission.sh
```

Explanation:

(a) 2nd line - submitted file name not valid

(b) 4th line - integer not valid

(c) 5th line - email id not valid

Problem 4: Let's extract

Suppose you have a file `data.txt` with the following contents:

```
sakrat:74:jodhpur
kavgup:95:lucknow
gursin:100:chandigarh
maykum:69:haryana
Ridsar:88:jodhpur
```

Write a unix commands that:

(a) Sorts the lines based on the second column (numeric sorting) and in reverse order. Redirect theoutput to `sort.txt`. (Redirection can be done through "¿")

(b) Extracts the lines where the third column is "jodhpur". (Perform this on sort.txt and redirect the result to `extract.txt`)

(c) Takes only the first two columns and prints them to `display.txt`. These columns should be separated by tab instead of colon. (This command has to be performed on `extract.txt`.)

(d) Combines these three files and tar them. The final file name will be `submission.tar.gz`.

You think its a lot of stuff to do??? Let's understand what's happening through the given example.
For the given `data.txt`, `sort.txt` will contain:

```
    gursin:100:chandigarh
    kavgup:95:lucknow
    ridsar:88:jodhpur
    sakrat:74:jodhpur
    Maykum:69:haryana
```

`extract.txt` will contain:

```
    ridsar:88:jodhpur
    Sakrat:74:jodhpur
```

`display.txt` will contain:

```
    ridsar 88
    sakrat 74
```

(This is because `display.txt` contains the first two columns separated by tab.)

Problem 5: Playing with access controls

You are given a zip file named `input.zip`. Write a unix command that achieves the following:

(a) Unzips `input.zip` to get the contents inside.

(b) All the files given in the unzipped folder have read, write and execute permission to all users. You must change the permissions to read, write and execute permission to the user who created the files (`user (u)`) and read and execute to any other user (`group (g)` and `others (o)`). This will be done for all the contents (files or subfolders) in the directory.

(c) Zip the modified directory and name it `output.zip`.

Problem 6: Line count

Note: *Please try this activity after you have completed all the other ones.*

There is a directory named `input`, which itself contains some sub-directories. These directories contain some files.
Write a unix command (or a sequence of commands) to list the total number of lines in all the text files (`*.txt`). Remember, you need to exclude all the empty lines within the `.txt` file. Also, consider only `.txt` files. All other files have to be ignored.
Example: If the `input` folder directory structure is:

```
input
|----- project1
|     |----- file1.txt
|     |----- file2.txt
|     |_____ file3.pdf
|
|----- project2
|     |----- file4.txt
|     |----- file5.txt
|     |_____ file6.docx
|
|_____ project3
      |----- file7.txt
      |_____ file8.txt
```

You are also given the number of non-empty lines in these files:

(a) `file1.txt` - 5

(b) `file2.txt` - 7

(c) `file3.pdf` - 4

(d) `file4.txt`- 3

(e) `file5.txt` - 6

(f) `file6.docx` - 7

(g) `file7.txt` - 2

(h) `file8.txt` - 0

You need to print the result in `output.txt`, which for this case, the `output.txt` file will contain a single integer $= 5 + 7 + 3 + 6 + 2 = 23$ (Exclude files having extensions other than `.txt`)