

Writing Bash scripts

2016-17, CSCI 3150 - Assignment 1

Release: 22 Sep 2016

Early Bird Deadline: 29 Sep 2016 11:59AM

Deadline: 6 Oct 2016 11:59AM

Contents

1	Introduction	2
1.1	Example Usage	2
2	Your assignment	6
2.1	Remarks	6
2.2	To begin	7
2.3	Your job	7
2.4	Submission	8
2.5	Assumptions	8
3	Grading	8
3.1	Late Submission Policy	9
4	Questions	9
5	Academic Honesty	9

1 Introduction

In this assignment, you are going to write shell scripts to process files.

The main shell script is `run.sh`. Its syntax is:

```
./run.sh [operation type] [files]
```

The first argument is “`operation type`”, which is an integer that denotes the operation to be applied to “`files`”. `files` specifies the list of filenames. The possible file operations corresponding to their `operation type` are as follows:

operation type	Operation
1	Build a word-length histogram for all files
2	Count the number of appearances of given phrases in the files of interest
3	Phrase finding: lists the locations of given phrases in the files of interest

1.1 Example Usage

Operation 1. Build a word-length histogram for file `a.txt`

Input	<code>./run.sh 1 testcases/data/a.txt</code>
<code>a.txt</code>	<code>hello world,csci3150,:D</code> <code>a.b.c.d.</code>
Output	<code>1 4</code> <code>2 1</code> <code>5 2</code> <code>8 1</code>

In this example, 5 delimiters are used: newline characters (`\n`), tabs (`\t`), space (`␣`), full-stop (`.`), and comma (`,`). (The first three are system default, the remaining two are extra.)

So, in `a.txt`, “`hello world`” is regarded as two words, but “`:D`” is regarded as one word. The output is a histogram. Each line has the format of “`<length>_<count>`”. So, the line “`5 2`” means there are 2 words of length 5.

Operation 2. Given the phrases in file `b.txt`, list the total number of their appearance in the files `c.txt` and `d.txt`

Input	<code>./run.sh 2 testcases/data/b.txt testcases/data/c.txt testcases/data/d.txt</code>
<code>b.txt</code>	hello world hello world of OS 404 not found
<code>c.txt</code>	hello hello world OS hello world hello world of OS
<code>d.txt</code>	hello world of OS 404 found?
Output	3 5 2 0

Each line in file `b.txt` is a phrase of interest. Each output line corresponds to the total number of appearances of a phrase in the files of interest, which are `c.txt` and `d.txt` in this example. For phrase “hello world”, it appears 3 times in total: twice in file `c.txt` and one in file `d.txt`. So, the first output line is 3. The last line in `b.txt` is the phrase “404 not found”, which is not found in any files. So, the last output line is 0.

Operation 3. Given the phrases in file `b.txt`, list the locations of their appearance in files `c.txt` and `d.txt`

Input	<code>./run.sh 3 testcases/data/b.txt testcases/data/c.txt testcases/data/d.txt</code>
Output	<pre> hello world testcases/data/c.txt:1:6 testcases/data/c.txt:2:6 testcases/data/d.txt:1:0 hello testcases/data/c.txt:1:0 testcases/data/c.txt:1:6 testcases/data/c.txt:1:21 testcases/data/c.txt:2:6 testcases/data/d.txt:1:0 world of OS testcases/data/c.txt:2:12 testcases/data/d.txt:1:6 404 not found </pre>

Each line in file `b.txt` is a phrase of interest. For each phrase, the operation lists all locations of its appearance in the files of interest, which are `c.txt` and `d.txt` in this example. The output is in the following format:

```

<phrase 1>
<file-name>:<line-number>:<start-pos>
<file-name>:<line-number>:<start-pos>
...
<phrase 2>
<file-name>:<line-number>:<start-pos>
<file-name>:<line-number>:<start-pos>
...
```

with `line number` starting from 1, and `start-pos` starting from 0.

For example, the phrase “`world of OS`” appears in both file `c.txt` and `d.txt` for once, so the output contains two locations after the phrase. The phrase “404 not found” does not appear in any file, and hence the phrase is not followed by any location in the output.

Error checking.

Input	<code>./run.sh 2 e.txt f.txt</code>
Output	<code>File-error: e.txt!</code> <code>File-error: f.txt!</code>

Given files “`e.txt`” and “`f.txt`” do not exist (or is not a regular file), the script shall output “`File-error: e.txt!`” and “`File-error: f.txt!`”.

2 Your assignment

You are given the following files:

Name	Description
<code>/run.sh</code>	The main script (Don't touch).
<code>/histogram.sh</code>	Script to achieve Operation 1 (Your work)
<code>/phrasecount.sh</code>	Script to achieve Operation 2 (Your work)
<code>/phrasefind.sh</code>	Script to achieve Operation 3 (Your work)
<code>/testcases/data/ {data{a,b,c,d,1,2,3,4}, phrase{1,2,3}}.txt</code>	Test data.
<code>/testcases/expected/{0, 1a, 1b, 1c, 2a, 2b, 2c, 3a, 3b, 3c}.txt</code>	Expected output of 10 given test cases
<code>/grader.sh</code>	We will run this script to grade your assignment (Don't touch)

2.1 Remarks

`run.sh` The main script. Given to you. Don't modify this file or you risk **0 mark**.

1. It parses the input argument and exports ENV variables:

- `ASGN1_OP_TYPE` : operation type
- `ASGN1_FILES` : the file list, files
- `ASGN1_IFS` : the set of extra delimiters, separated by a newline character (`\n`)

2. Then it invokes `histogram.sh`, `/phrasecount.sh`, or `/phrasefind.sh` based on the operation type chosen by the user.

`grader.sh` It contains 10 test cases and it will match your output with our expected output.

2.2 To begin

Make sure all shell scripts (*.sh) are executable by `chmod 755 *.sh`. Then, run `grader.sh`, you shall see something like this:

```
> ./testcases/data/words3.txt:3:12
> ./testcases/data/words3.txt:3:24
> _
> ./testcases/data/words3.txt:2:5
> ./testcases/data/words3.txt:3:5
> ./testcases/data/words3.txt:3:17
> ./testcases/data/words3.txt:3:29
Failed case 3c (left: your answer, right: correct answer)
> pattern1
> ./testcases/data/words3.txt:1:0
> ./testcases/data/words3.txt:1:36
> ./testcases/data/words4.txt:1:1
> ./testcases/data/words4.txt:1:38
> ./testcases/data/words4.txt:5:0
> ./testcases/data/words4.txt:5:59
> hello_world
> ./testcases/data/words3.txt:2:0
> ./testcases/data/words3.txt:3:0
> ./testcases/data/words3.txt:3:12
> ./testcases/data/words3.txt:3:24
> ./testcases/data/words4.txt:1:25
> ./testcases/data/words4.txt:3:28
> world
> _
> ./testcases/data/words3.txt:2:5
> ./testcases/data/words3.txt:3:5
> ./testcases/data/words3.txt:3:17
> ./testcases/data/words3.txt:3:29
> ./testcases/data/words4.txt:1:16
> ./testcases/data/words4.txt:1:30
> ./testcases/data/words4.txt:2:48
> ./testcases/data/words4.txt:3:33
[Result] 0/10 test cases passed
```

For each failing test case, `grader.sh` prints your output on the left against our expected output on the right. At the end, it prints the total number of test cases passed.

2.3 Your job

Your job is to fill up the contents for `histogram.sh`, `phrasecount.sh`, or `phrasefind.sh` in order to pass all given test cases like the following:

```
[Result] 10/10 test cases passed
```

2.4 Submission

Zip only 3 files:

- `histogram.sh`
- `phrasecount.sh`
- `phrasefind.sh`

Name the zip file as “asgn1.zip” and submit it to eLearning.

Warning: DON’T change the file names or otherwise you get 0 marks

2.5 Assumptions

You can assume the following always holds in this assignment:

Input files

- No spaces or colons (:) within a filename
- Files contains ASCII characters only

Phrases

- No colons (:) within a phrase
- Spaces () and tabs (\t) may appear in a phrase
- Only count a phrase that completely appears on the same line

3 Grading

1. 10 test cases in total. They are named after the operation type. So, testcase2a, testcase2b, testcase2c are testing operation 2.
2. Passing one test case will get 10 marks. To encourage early submission, students who submit their **final** version before the early bird deadline, passing one test case will get 11 marks.

3. Your script shall output results to standard output stream but not to any file. Otherwise, you will get 0 mark.
4. Your script shall NOT write any intermediate result to file. Otherwise, you will get 0 mark.
5. **Hardcoding results won't work. We will use another set of test data and expected output when grading.**
6. The grading will be fully automated. Once we start the grading processing, we reserve the right to deduct marks from you for any request that requires TA's extra manual effort.

3.1 Late Submission Policy

We follow the late submission policy specified in the course outline.

4 Questions

If you have doubts about the assignment, you are encouraged to ask questions on our Facebook group.

5 Academic Honesty

We follow the University guide on academic honesty against any plagiarism.