

Assignment 2

CSE 2031 3.0 Software Tools, Fall 2012, Section E

Due: Thursday, November 29, 2012, 5pm.

Format: Individual.

Specification

Your task is to create a shell script (called `a2.sh`) that will be used to test a C program. The script will receive as input the source of the C program, as well as test cases for this C program in the form of input with corresponding output (see details below). Your script must validate that the arguments it receives are valid. If so, the script prints the number of test cases passed. If not, an appropriate error message is printed. In more detail:

Your script must expect 3 arguments:

1. The path to a directory (may be either an absolute or a relative path) that contains the test cases (the path name may or may not include a trailing slash). Any file in this directory with the extension `.in` is assumed to be the input for one test case. The corresponding output for this input must be in a file in the same directory with the same basename and the extension `.out`. All other files in the directory must be ignored. `.in` or `.out` files without a corresponding `.out` or `.in` file must also be ignored. The following is a path to a sample directory:

```
/cse/dept/course/2012-13/F/2031/A2/testCases
```

2. The path to a file (may be either an absolute or a relative path) that contains the C program to be tested. A corresponding sample program to the test case directory above is:

```
/cse/dept/course/2012-13/F/2031/A2/alsol.c
```

3. An integer that indicates how long to wait for a particular test case to run. This is necessary since the C program may enter an infinite loop.

Assuming that all three arguments are valid, i.e. the directory exists and is readable, the file exists, is readable, and compiles, and the third argument is an integer, then your script must run the C program with every test case input and confirm that it produces the corresponding output (using `diff`). The **only** output from your script in this case must be a line that indicates how many test cases passed. The output must look like this:

```
18 out of 18 tests have passed.
```

The two numbers in the output will of course differ depending on how many test cases were found in the test case directory, and how many of them produced the expected output.

To help you run the test cases with the time limit provided by the third argument, an executable has been provided in:

```
/cse/dept/course/2012-13/F/2031/A2/timeout
```

It takes two arguments:

1. An integer indicating the time limit in seconds.
2. A path (absolute or relative) to another executable (in your case the result of the compilation of the C program).

If something is wrong with its arguments, your script must produce **only** an one line error message as described below:

- If the number of arguments is not 3, the output must be:

```
Usage: a2.sh <testdir> <C program> <time limit>
```

- If the first argument is not a directory, the output must be (arg1 must be replaced by the argument provided):
arg1 is not a directory.
- If the first argument is a directory, but it is not readable or “executable”, the output must be (arg1 must be replaced by the argument provided):
arg1 is not accessible.
- If the second argument does not exist or is not readable, the output must be (arg2 must be replaced by the argument provided):
arg2 is not accessible.
- If the second argument does not compile (use the cc compiler with no options), the output must be (arg2 must be replaced by the argument provided):
arg2 does not compile.
- If the third argument is not an integer, the output must be:
The third argument must be an integer.

There must be only one line of error output. If multiple arguments are invalid, the error message must be about the first argument in order to be invalid.

The exit status of your script must be 1 if an error occurred, or 0 if there was no error. Finding 0 test cases in the provided directory is not considered an error. The output in this case must be:
0 out of 0 tests have passed.

There must be single SPACE characters between the different words of the output, and no SPACES before or after it. The output must be followed by a single newline character. There must not be any empty lines in your output.

Hint: You might find the Unix utilities `basename` and `mktmp` useful.

Submission

Name your script `a2.sh`. Before submitting, make sure that it runs correctly in the lab. Once ready, submit with

```
submit 2031 A2 a2.sh
```

Grading

This assignment will be marked automatically. This means:

1. There will be no marks for comments, meaningful identifier names, etc. However, you should still follow appropriate style guidelines for your own benefit.
2. Scripts with syntax errors will receive an automatic F.
3. The output produced by your script must match the expected output exactly, otherwise you will receive an F.
4. Your script will be tested with lots of different input. It has to behave according to the specification in every case. You must test your program with any possible input you can think of to ensure it works correctly.
5. You must work on this assignment on your own. Submissions will be tested for similarity. Suspicious cases will be reported to the Faculty of Science and Engineering for plagiarism.