CS 346 – Lab #11 – Counting files

Today, I'd like you to work on another useful application for a file system: counting files. A user might be surprised to learn how many user files are on the hard drive, and wonder which directory is most responsible. Maybe there are duplicate backups or other obsolete files that are no longer needed.

This will be a recursive descent program similar to the one you wrote last week. Ask the user to enter the name of a directory, which could be entered as an absolute or relative path name. The program should then report the total number of files contained in this directory and in all of its recursively subdirectories. The program should also give a total count of files contained in each subdirectory that is located *exactly one level* below the desired directory. In other words: although the counting of the files is to be done recursively, the reports should only be made at the top two levels of the user's input.

Special notes:

- If the the user enters the name of a file instead of a directory, then your program should patiently report that there is one file.
- Your count should include only files, not directories.

Here is an example. In the lab11 directory you can find a "test" directory that contains a simulated file hierarchy. The recursive directory listing (ls –R) of test reveals the following information:

```
test:
a.txt  b.txt  c.txt  one  three  two

test/one:
A  B  d.txt  e.txt

test/one/A:
f.txt

test/one/B:
g.txt  h.txt  i.txt  j.txt

test/three:
D  p.txt  q.txt  r.txt  s.txt

test/three/D:
hide  seek  t.txt  u.txt  v.txt

test/three/D/hide:
w.txt  x.txt  y.txt

test/three/D/seek:
z.txt
```

```
test/two:
C  k.txt  l.txt

test/two/C:
m.txt  n.txt  o.txt
```

To check the output of your program, here are some example I/O based on this test directory. Note that in this output, “top level” refers to the user’s desired directory, so that it is more easily distinguished from the subdirectories.

```
[chealy@csserver2016 file]$ java Count
Which directory to examine?  test
test is a directory.
     7 files in /home/chealy/cs346/file/test/one
     5 files in /home/chealy/cs346/file/test/two
    11 files in /home/chealy/cs346/file/test/three
     3 files at top level
Total number of files is 26

[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/one
test/one is a directory.
     1 files in /home/chealy/cs346/file/test/one/A
     4 files in /home/chealy/cs346/file/test/one/B
     2 files at top level
Total number of files is 7

[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/A
test/A does not exist.
     0 files at top level
Total number of files is 0

[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/one/A
test/one/A is a directory.
     1 files at top level
Total number of files is 1

[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/two
test/two is a directory.
     3 files in /home/chealy/cs346/file/test/two/C
     2 files at top level
Total number of files is 5
```

```
[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/three
test/three is a directory.
     7 files in /home/chealy/cs346/file/test/three/D
     4 files at top level
Total number of files is 11

[chealy@csserver2016 file]$ java Count
Which directory to examine?  test/three/D
test/three/D is a directory.
     3 files in /home/chealy/cs346/file/test/three/D/hide
     1 files in /home/chealy/cs346/file/test/three/D/seek
     3 files at top level
Total number of files is 7
```