Virginia Tech
Bradley Department of Electrical and Computer Engineering
ECE-3574: Applied Software Design * Fall 2012

Homework 3

Submission Details

You must submit the solutions for this homework as an electronic submissions using Scholar (under ECE3574 → Assignments → Homework 3). The submission must be a gzipped tar file (.tar.gz) with your source code. Include all necessary project files, but no binary or compiled files. Your program will be run to evaluate its correctness, and the source code will be reviewed for adherence to the Qt programming style. Your program must run on Ubuntu 12.04 and compile/build using the GNU C/C++ compiler and the qmake/make tools. The following information must be included at the top of each of your source files as comments: your full name, your student ID number, your email address, class (ECE 3574), and the title of the assignment (Homework 3). The submitted file must be given a name in the following form: ***LAST_FIRST_hw3.tar.gz*** where LAST is your last or family name and FIRST is your first or given name. You are only allowed to make one submission. Paper, email or Drop Box submissions will not be accepted. All work must be submitted by the announced due date/time. **Late submissions will not be accepted!** (Don't do it! You have been warned!)

Questions

Use the Homework 3 forum in the Discussion Board area of the class web site to ask questions about this assignment. Do not post questions that contain specific information about the solution.

Honor Code

As stated in the syllabus, in working on homework and projects, discussion and cooperative learning are allowed. However, copying or otherwise using another person’s detailed solutions to assigned problems is an honor code violation. See syllabus for details.

This homework is based on exercise 1 from
http://cartan.cas.suffolk.edu/oopdocbook/opensource/ex-visitor.html

diskusage reports, in various formats, the amount of disk space used in a directory tree. It traverses all files and computes total file sizes, providing different levels of detail (depending on command line arguments) in its report.

The command-line format of invoking diskusage is as follows:

```
./diskusage [-v|-w|-c|-d] [-b|-k|-m] arg1 [arg2 arg3 ...]
```

Options for diskusage

Recursion arguments:

-v Recursively show each directory visited and their total sizes. If one of the arguments given is a file you don't need to display its information.

-w Recursively show each directory and each file visited and their total sizes.

-c Show the total sizes for the arguments given as well as each file and directory 1 level inside of it. The size of a given file or folder should be the same as for the –w flag but this option won't necessarily print every item that –w does.

-d Show the total sizes for the arguments give. This should give the same sizes as –c but won't necessarily print out every item that –c does.

Display Arguments:

-b show totals in bytes

-k show totals in kilobytes (1024 bytes)

-m show totals in megabytes (1024*1024 bytes)

If no flag is given for the recursion arguments it defaults to -v and if no flag is given for display arguments it defaults to -k.

Find the sizes of each item in bytes and only round the number being printed (do not round values that will be used in later size totals). Round displayed values **up** to the next whole unit. For example, if a folder has two files of 400 bytes then –v –k would show both files as being 1k in size and the folder would also be 1k in size (for the sake of this example we'll assume the folder has a size of 0 bytes).

Don't forget that even empty folders have a size that needs to be taken into consideration when calculating the total.

Check that subdirectories in the traversal each have their own totals properly calculated. Compare your results to the *nix command **du**.

Examples:

Suppose the files and directories we’re testing look as follows:

```
./diskusage
./A/
./A/Empty/  An empty folder
./A/1k
./A/2k
./A/E       An empty file
./1k
./B/
./B/A/
./B/A/3.5k
./B/A/2k
./B/1k
./B/2k
./B/B/
./B/B/3.5k
./B/B/2k
```

diskusage should behave in the following ways:

```
$ ./diskusage –v –b A/ B/ 1k
A/Empty 4096
A       11264

B/A 9728
B/B 9728
B   26624

$ ./diskusage –w –b A/ B/ 1k
A/1k    1024
A/2k    2048
A/E     0
A/Empty 4096
A       11264

B/1k     1024
B/2k     2048
B/A/2k   2048
B/A/3.5k 3584
B/A      9728
B/B/2k   2048
B/B/3.5k 3584
B/B      9728
B        26624

1k 1024
```

```
$ ./diskusage –c –b A/ B/ 1k
A/1k    1024
A/2k    2048
A/E     0
A/Empty 4096
A       11264

B/1k 1024
B/2k 2048
B/A  9728
B/B  9728
B    26624

1k 1024

$ ./diskusage –d –b A/ B/ 1k
A 11264

B 26624

1k 1024

$ ./diskusage –v –k A/ B/ 1k
A/1k    1k
A/2k    2k
A/E     0k
A/Empty 4k
A       11k

B/1k     1k
B/2k     2k
B/A/2k   2k
B/A/3.5k 4k
B/A      10k
B/B/2k   2k
B/B/3.5k 4k
B/B      10k
B        26k

1k 1k
```

```
$ ./diskusage –v –m A/ B/ 1k
A/1k    1M
A/2k    1M
A/E     0M
A/Empty 1M
A       1M

B/1k     1M
B/2k     1M
B/A/2k   1M
B/A/3.5k 1M
B/A      1M
B/B/2k   1M
B/B/3.5k 1M
B/B      1M
B        1M

1k 1M
```