Andreas Landgrebe
Pledge:
Feburary 9, 2014
Laboratory Assignment Two
Implementing and Using a File System Traversal Tool

3. The report from an experimental study that characterizes the use of the Linux file system:

In this laboratory assignment, we had implemented a traversal file system to be able to go to a specific root in the computer and display as output the maximum, minimum, and average size of a file in a specific root specified. In this assignment, we were also asked do to be able to implemenet this task using JCommander to be able to prase the command line for running the File System Traversal tool. For the purpose of this laboratory assignment, I was succesful in implementing a tool for File Sytem Traversal. For this laboratory assignment, I had decided to implement this assignment using my mac book pro instaed of the ubuntu work station since I have now become more accostumed to the mac os x operating system and the commands are similar if not the same as the commands on the ubuntu workstation computers. I had decided to implemented the assignment using eclipse so I add my arguments in when I run the run configurations tabs.

In order to complete this laboratory assignment, one of the requirements was to use JCommander. JCommander is a tool a parse command line arguements. In order to so, I would need to download the source code of the tool which is avaialble at http://www.jcommander.org, at the bottom of this page, there is a link that will direct you to the git hub repository where you can download the source code as a zip file which I was succesfull in doing. After this, I had saved it all of the source in a working directory. In order for JCommander to work succesfully you would need to implement parameters in order to parse specific commands in the terminal to make sure JCommander works properly. In the parameters.java file that I have created, I created a root paramteres to access a specific directory to run the traversal tool. I also set up a levels parameters to set up the degree in which how many files are going to be accessed. After the parameters have been implemented, the source code the TreeTraversal needed to implemented successfully. This is the main method to set up how the input and output are going to be displayed. For the TreeTraversal, the first thing you needed to do is to set up some universally variables. The universal variables that needed to be decalred would be the number of files, the size of files, the maximum size of a file, the minumum size of a file, the number of directories, the max number of levels and the current level. The number of files is used to count the number of total files that have beene examined in running the traversal tool. The size of files is important to displayed the average size of all the files since in order to calculate the average size of the files, you would need take the total size of files and divide it by the total number of files. The java.io.file. tool, I saw that you could calculate the number of direcories in the root which I thought to be an interesting statistic to examine. The maximum and minimum size of the files was also an important variable to declare. In order to determine the maximum size of the file, you would need to declare the maximum size variable to zero and compare to each file that if the maximum size is less than the file being examined, then you would be the maximum size variable equal to the

current file being examined. For the minumum size of a file, you need perform a similar process in calculating the maximum size except you would need to assess if the minumum size of a file is greater than the current file being examined. For the maximum and current level, it was important to implement so I was able to change the root and assign in the command line the depth in which all of the files in the directory. Below is a table that displays all of the parts of the file system and displays the different minimum, maximum and average file sizes for each different part of the file system.

	Minimum File Size	Maximum File Size	Average File Size
Desktop	10 byte	576,716,800 bytes	123,276 bytes
Documents	1 byte	20,325,153 bytes	39,648 bytes
Downloads	8 bytes	255,649,794 bytes	1,990,342 bytes
CS290 Final Project	8 bytes	7,383,085 bytes	87,694 bytes
Workspace from Eclipse	1 byte	11,277,826 bytes	43,253 bytes

This is another table that displays the number of fiels that has been examined, the number of directories that have been examined and the total time spend to complete the traversal file system tool.

	Number of Files	Number of Directories	Total Time Spend
Desktop	30862 files	4167 directories	8.982 seconds
Documents	3837 files	945 directories	1.476 seconds
Downloads	202 files	39 directories	0.086 seconds
CS290 Final Project	233 files	72 directories	0.026 seconds
Workspace from Eclipse	461 files	233 directories	0.069 seconds

Some important tools to examine in the future implementations of future file systems is the types of files that are in a specific directory. In the specfic directory, the file system traversal tool should be able to display how many mp3, png, pdf, or mp4 files they are in a directory. When these specifications are implemented, the traversal tool should also be able to tell you the maximum, miniumum and average size of these different types of files specified.