CS 346 – Lab #12 – File experiments

The purpose of this lab is to gather some empirical data regarding the performance of our memory and file systems. Create a new folder on your Linux account called lab12, and do all your work in there. From the class Web site, please download the file Files.java, which you will be modifying today.

Here is a handy Linux command:

```
du –s <name of file or directory>
```

This will tell you how much space is occupied by the file or directory you specify. Usually, the unit of measure is in kilobytes. For example, if you enter this shell command: `du –s ~`, it will tell you the total amount of disk space taken up by all your files in your account.

As another example, use the `mkdir` command to create a subdirectory called `dummy`. Use the `du –s` command to verify that this directory is indeed empty. You will be using the dummy directory later in the lab, so don't get rid of it.

Empirical testing in Files.java

The purpose of Files.java is to measure the amount of time it takes the system to perform various file and memory tasks. There will be five time experiments. For each one, there is some code that you will need to write. In other words, you will be writing some loops. Follow the comments in the program to see where your code should be inserted. Your five experiments answer these questions:

Experiment 1: How long does it take to create files? Write a loop that creates 1000 empty files.

Experiment 2: How long does it take to delete files? Delete those 1000 files you just created.

Experiment 3: How long does it take to write to a file? Write 100 MB to a new file.

Experiment 4: How long does it take to read from a file? Read the file you just created in #3.

Experiment 5: How long does it take to delete a big file? Delete the file used in #3 and #4.

Question #1: Run your program five times. Write down the millisecond results that you see.

Experiment	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Creating 1000 files					
Deleting 1000 files					
Writing 100 MB					
Reading 100 MB					
Deleting 100 MB file					

Question #2: Give two reasons why your results would vary from one trial to the next.

Question #3: Why does deleting a 100-MB file take so little time?

Is Zero greater than Zero?

We are now ready for one last experiment:

Experiment 6: How much space is taken up by 1000 empty files?

In order to answer this question, modify Files.java – comment out the loop pertaining to the second experiment where you deleted the 1000 files. Compile and run Files.java again. This time, the 1000 empty files in the dummy directory are retained. Run the du –s command on the dummy directory.

Question #4: How much space does `dummy` occupy? Use ls –l to verify that the files in dummy are indeed 0 bytes apiece. Evidently, the whole is greater than the sum of the parts, and we can say 0 > 0 ! Why does the dummy directory occupy a nonzero amount of space?

When you are done with Experiment 6, uncomment the loop around Experiment 2 before continuing.

Question #5: Which of the Experiments 1-6 above pertain to the file system, and which pertain to the memory system?

Summary

Question #6: Based on the results of your program, compute the following averages. For the first two measurements, give units of microseconds or milliseconds, as appropriate.

1. The time it takes to create one file is _____.
2. The time it takes to remove one file is _____.
3. When writing to a file, the data transfer speed is _____ per second.
4. When reading a file, the data transfer speed is _____ per second.
5. The overhead of an empty file takes up this much space _____.

Finally, let's run the time experiments on a different machine, one other than our Linux server. Copy your program to that machine, and re-run the results of Experiments 1-5. If you are using the Windows environment, please note that the directory delimiter is \ instead of /.

Question #7: How do your results differ on the second machine? Are all the operations uniformly faster or slower on the second machine?

Please also submit the source code of the program you wrote to obtain your measurements.