CPS 536: Homework 5 **Dr. Clark**

Assignment #4: A Look At File Caching
Due Monday, April 21

The goal of this assignment is to gain some understanding of the real operation and effect of file system caching. For this assignment you will not be writing any code. (Please try not to be too disappointed!) Instead, you will be running a sample program I have written that performs various file I/O operations. You will run the program on three different types of file systems and comparing the effect of caching in each case.

The program is called *fstest* and you can copy it from the directory */home/rjc/pub/cps536/fstest*. To run the program you give it a directory name indicating where you want the program to generate its test files. In this way you can identify which file system you want to test.

The program performs six different I/O tests:

1. Write a sequential 20MB file. Actually, the file is 5000 blocks of 4096 bytes each so it's roughly 20MB.
2. Write the same amount of data but write it all to a single block. (i.e. write to the same block 5000 times)
3. Write a sequential 20MB file. Then, immediately read the same file.
4. Write the same amount of data to a single block then immediately read the same amount from that block.
5. Write a sequential 20MB file using *synchronous* writes. Synchronous writes cause the file system to write each byte to disk immediately.
6. Write the same amount of data using synchronous writes to a single block.

Data Collection

You should run the program on the following three different file system configurations. You should record the results of your runs as reported from the fstest program. You should run each test several times and use an average of the observed results. For these experiments, 3 to 5 trials should be sufficient.

1. **Local Disk File**
   With this test you should see the effects of basic disk caching, scheduling and file system implementation. You can perform this test by creating files in the */local/tmp* directory which is a publicly accessible directory that is always local to each machine. Run with *fstest /local/tmp*
2. **Local Virtual Disk File**
   Many Unix programs make frequent use of the */tmp* directory for storing temporary files needed while running a program. For instance, the *vi* editor makes a copy of the file you are editing so that changes you make are not made to the original until you specifically write them. Because of the frequency of use, Solaris makes this directory more efficient by placing it in a virtual file system that exists only in memory. Your tests here should not involve any real disk access! Run with *fstest /tmp*
3. **Network File System File**
   The files in your home directory are actually stored on the remote server named *sunspot*. As we discussed in class, this remote file service involves several layers of caching. Note, in order to perform this test correctly you must be logged into one of the lab systems. **Do not log in directly to sunspot.** Run with *fstest .* in your home directory.

Data Analysis

Now that you've gathered your raw performance data you should analyze the numbers to find out what is going on. You should provide me with a written analysis of the behavior you observed during your testing. In your analysis, you should explain the operation of the system during each test. In particular, you should identify the issues discussed in class and in the book: caching, file allocation, file header management, memory vs. disk vs. network access times and disk scheduling. I want you to explain the results you are seeing in terms of these issues.

You should summarize the results of your tests. For this you can use tables. You should also identify some interesting results that you can compare using graphs. You will have 6 tests on 3 different file types for 18 different combinations of parameters. Make sure you discuss all 18 distinct trials.