CS 740, Fall 1997

Assignment 4:

Memory Systems and I/O (Part II: I/O)

Assigned: Wed., Oct. 22

Due: Mon., Nov. 3

Policy

You may work in a group of up to 3 people in solving the problems for this assignment. You should turn in a single report for your entire group, identifying all of the group members.

Logistics

Any clarifications and revisions to the assignment will be posted on the class bboard and Web page. There is no electronic handin for this assignment. Turn in the two parts as separate documents, so that they can be graded independently.

1. I/O System Performance

In this part of the assignment you will assess the performance of the hard disk drive on your workstation. You may find it convenient to use the same platform for this part of the assignment as for the first part, but this is not a requirement.

1.1. Disk Performance Parameters

The goal of this problem will be to evaluate the rate at which your disk drive can perform a seek operation, and the rate at which data can be read and written once the seek has been performed. It will be interesting to examine these measurements both in units of time and in CPU cycles.

Making measurements of disk performance is somewhat complicated on a machine running Unix because the operating system performs various operations "under the covers" to improve the performance of the I/O system. For example, the operating system typically maintains a cache in memory of recently read disk blocks, so that future reads can avoid accessing the disk. The operating system may also prefetch disk blocks to improve the performance of sequential accesses to the disk. Prefetching has two benefits. First, if the sequential accesses are occuring at a rate comparable to the rate at which the disk can be accessed (e.g., some processing of the data is taking place between reads), then prefetching hides the latency of reading the disk. Furthermore, prefetching may reduce the number and cost of seek operations, since the head will typically be in the correct position to read the next block.

In order to evaluate the performance of your disk, you will have to ensure that you know when disk blocks are being cached and when they are not. There are several strategies for flushing the disk cache. Unmounting and then remounting a file system should flush the cache for that file system. In Solaris, a super-user can perform these functions using the `umount` and `mount` system commands. If you feel queasy about unmounting your file system, another possibility is to fill the cache with disk blocks from one file, and then make your measurements by accessing another file. This method should work if the first file is larger in size than the amount of physical memory on the machine. Note, however, that the operating system may have a policy of not allocating the entire disk cache to a single file, so to be safe you should use several files to fill the cache.

You should also make an effort to ensure that no other applications are accessing the disk during your experiments. Finally, watch out for buffered or delayed I/O operations, e.g., a write buffer that is not flushed until a file is closed.

1.2. The Data to be Gathered

Determine, to the best of your abilities, the following characteristics of your machine. Except where noted, you may use any source of information, provided that you document the source.

1. The processor model and speed, the size of main memory, the size of the hard disk, and the operating system.
2. The size of a file block, i.e., the minimum amount of storage allocated on a disk for a file. The Unix system calls `stat` and `statfs` (Linux) might be helpful here.
3. The maximum rate at which your disk can perform seek operations. For this item, you should perform an experiment, being careful to avoid making any hits in the disk cache. Reasonable numbers (I think) will be between 50 and 200 seeks/second. Give the number both in terms of seeks/second and in CPU cycles/seek. (You may also be able to estimate this rate using a system command such as `iostat` (Solaris).) **Describe your experiment.**
4. The rate at which data can be read from or written to the disk using a random-access pattern. For this item, you should perform an experiment, avoiding any hits in the disk cache. You should read data from the disk in chunks equal to one file-block in length, avoiding sequential accesses to the disk. Give the rate in terms of KB/sec and in terms of CPU cycles/KB. Random accesses can be performed using the `lseek` system call. **Describe your experiment.**
5. The rate at which data can be read frm or written to the disk using a seqential access pattern. For this item you should perform an experiment. Initially, your disk cache should be clean. **Describe your experiment.**

For extra credit, obtain the following pieces of information.

1. The size of the disk cache.
2. The disk cache replacement policy.