CSE 30341: Home Work Project 4

Assigned: Mar 22, 2006

Due: Apr 7, 10:40AM

Late submissions will not be accepted

Group effort

File system simulator:

In this project, we will analyze the behavior of file systems in a similar fashion to the previous project. We will write a disk simulator which will indicate the disk access statistics. Your simulator will read a bunch of file requests, will write the corresponding directory and file structures to the simulated disk and then report the disk latency to run your simulation. Next we describe each of these steps in detail

1. **Step 1: Develop a disk simulator:**

   First we will simulate a simple disk. Your simulated disk will have the notion of cylinders, rotational latency, seek times etc. You will maintain the current disk head position at all times. So, when you issue a seek to a new disk location, you will calculate the time it takes for the head to be repositioned in order to carry out the next disk access operation. We will assume the following parameters for the disk: seek time in a track (rotational latency) – 1 ms (ave), seek time – 9 ms (ave), sectors per track – 500, total number of tracks – 1000, sector size – 512 kb. Thus, the total disk space is 500*1000*512 = 32 GB. The operations implemented by this step is “seek(sector)”, which moves the head from the current location to the new location.

2. **Step 2: Implement a disk scheduling algorithm:**

   On top of the disk described in Step 1, you will implement a disk head scheduling algorithm discussed in the text book (e.g. SCAN). Step 3 will basically issue a number of disk seek operations and the schedule algorithm will reschedule these operations efficiently. This step will basically implement an operation “read (sector)” and “write(sector)”. These will be reordered and result in various seek(sector) calls to Step 1.

3. **Step 3: Implement a file system**:

Next, you will implement a file system on top of this disk. Your file system will have the notion of file directory entries and file contents. You could use something like the combined scheme described in 11.4.3 (Figure 11.9) to layout the contents in the disk. You will keep track of free disk sectors and directory entries to figure out which sectors should be read to read the next bytes of the corresponding file.

4. **Step 4: Replay the trace through the system:**

You will replay the following access trace through your system. The traces are of the format <file> <size> <operation> . The <file> are denoted by a number, size is in kilobytes and operation can be Write (W), Read (R), or Delete (D). Write will write this file into the disk (don't worry about the contents), Read will read this particular file and Delete will delete the file. Both R and D have no need for the <size> field. So, "1 1024 W" means, write a new file named 1, of size 1024KB. You don't have to worry about name collisions, assume all W names are new. First, you will run your simulator through the following sample trace. I will email a more eloborate trace to the class mailing list later.

1 1024000 W
2 1024000 W
1 0 D
2 0 R
3 1024 W
2 0 R
4 1024 W
5 1024 W
6 1024 W
7 1024 W
8 40960000 W
4 0 D
5 0 D
7 0 R
7 0 D
5 40960000 W
5 0 R

You will report the simulated time to run the traces through your system..