Computer Organization Final Project – Hard Disk Scheduler

Due: 23:55, June 6, 2019

Hard disk is a mechanical storage system, which stores data on the surface of a magnetic platter, to provide a mass non-volatile storage environment in a computer system. A program and its data will be loaded from a hard disk to dynamic random access memory (DRAM) before being accessed by a microprocessor. However, the speed of a mechanical system is much slower than the speed of a microprocessor. The overall performance of a computer system could be strongly related to the disk access time. The disk access time consists of the seek time, the rotation delay and the data transfer delay. Empirically, the disk seek time contributes a significant part among the above three major factors. The disk seek time is related to the seek distance from the original position of disk read/write head to the new position of requested disk address. To minimize the disk seek time, we often exploit the benefit of locality by scheduling neighboring requested addresses together. However, the requested addresses should be scheduled within a certain of tolerance timing window to prevent from starvation condition. This is so-called the quality of service (QoS) constraint. For more details, please read Chapter 6.

In this project, you are going to invent a minimal seek distance disk scheduler under a given QoS constraint t. We formulate this problem as a one-to-one mapping problem below. Assume that the total number of disk references is N. There are two sequences, the requested sequence, $\{A_1, A_2, \ldots, A_N\}$, and the scheduled sequence, $\{R_1, R_2, \ldots, R_N\}$. In the requested sequence, the $i\text{-}th$ element A_i represents the disk address of the $i\text{-}th$ requested access. In the scheduled sequence, the $j\text{-}th$ element R_j represents that the $j\text{-}th$ scheduled access is the $R_j\text{-}th$ requested access in the requested sequence. According to the given QoS constraint t, both inequalities, $1 \le R_j \le N$ and $j - t \le R_j \le j + t$, must be satisfied for all $1 \le j \le N$. Therefore, the objective function of the minimal seek distance disk scheduler is:

$$\min \sum_{j=1}^{N} \left| A_{R_j} - A_{R_{j-1}} \right|, \text{assume } R_0 = 0, A_{R_0} = A_0 = 0$$

There are two inputs in your program, the requested sequence file (ex: "filename.in") and the QoS constraint t. There are also two outputs in your program, the scheduled sequence file ("access.out") and the total seek distance.

Please design your own disk scheduler according to the following rules:

1- The maximum number of disk accesses is 10,000 and the maximum disk address is 10,000.
2- Your program must be executed on a Linux workstation.
3- Upload your source code tarball (*.tgz) to moodle (including "Makefile" and "Readme.txt" to indicate how to build your executable).
4- The filename is your student ID (e.g., B12345678.tgz).
5- We are going to have a volunteer contest on June 1 (Saturday morning).

SYNOPSIS

```
%> hds access.in t
```

(NOTE: The input file name is given by user while the output file name is "access.out".)

Run-time Example:

```
%> cat access.in
1:50
2:30
3:70
4:20
5:50
%> hds access.in 2
The total seek distance is 90.
%> cat access.out
1:2
2:4
3:1
4:5
5:3
```

The following two results are for your references.

```
%> hds access.in 2
The total seek distance is 190.
%> cat access.out
1:1
2:2
3:3
4:4
5:5
```

```
%> hds access.in 2
The total seek distance is 120.
%> cat access.out
1:2
2:1
3:3
4:5
5:4
```