Douglas Mejia

Operating Systems

Project 4 write up

**PART 1**

The aim of this project was to emulate the following disk scheduling algorithms: First, come, first serve; Shortest seeking-time first; Scan; Look; C-Scan; and C-Look. First come, first serve handles all each request as in the order they arrive. Shortest seeking time looks for the the request closest to the current head. Scan and C-scan move in a certain direction and handle request on the way. Look and C-look act in the same manner as scan and c-scan but they don’t go till the ends of the disk, only to the farthest request. First come, first serve was typically the slowest. Shortest Seeking time usually had results better than first come first serve. Scan and Looks, efficiency depended on the starting position and where usually pretty close to each other.

I had the most difficulty coding c-scan and shortest seeking time first, but I was able to get around it by implementing a flag array that would keep track of which request had already been taken care of. Implementing this array made the other scheduling algorithms easier to keep track of.

*To run:*  In the terminal

gcc -o schedule Disk\_Scheduling.c

./schedule “starting position”

\*replace “starting position” with a number between 0 – 4999;

Example output:

