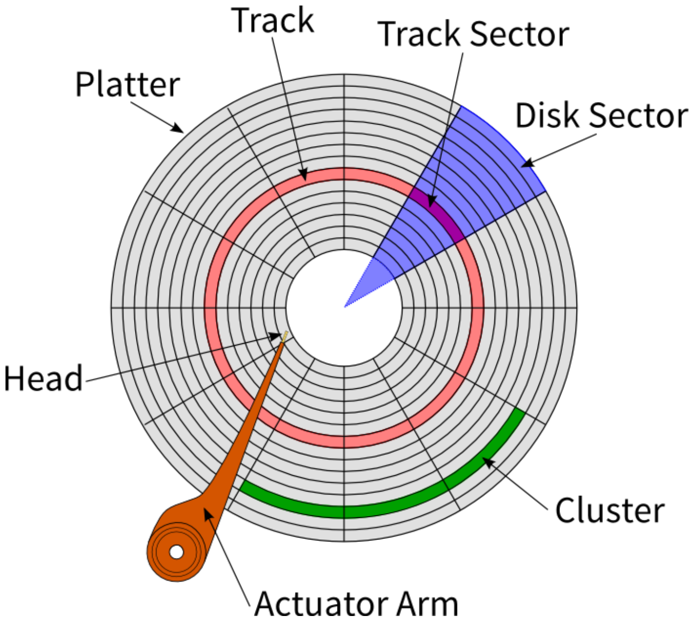
CPE 453
Individual work only. Any language that runs on unix machines is acceptable.

Disk Scheduling Simulator

In this lab you will be implementing a simplified disk scheduling simulator. For this program, we will assume the arms move at the same speed no matter the distance. This is not so in real hardware.

Given a sequence of disk track seek requests, your program must calculate and print the number of tracks that the head/arm will have to traverse in order to cover all the track requests in order and according to each algorithm.



All material discussed here is from the Silberschatz book.

Write a program that implements the following disk scheduling algorithms:

1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK

The program will implement a disk model with 5,000 cylinders (0-4999). The program takes one mandatory and one optional command line parameters.

The usage is as follows:

diskSim INITIAL_POSITION [ACCESS_SEQUENCE_FILE]

Where

- INITIAL_POSITION is an integer between -4999 and 4999 inclusive. Each number represents a disk cylinder position. The +/- sign represents the initial direction of travel which may be necessary in some algorithms. A negative sign means the arm is moving toward cylinder 0. A positive integer (‘+’ sign implied) means the arm is moving toward the outer edge or Cylinder 4999. For algorithms that don’t care about direction of travel, you may ignore the sign of the INITIAL_POSITION.
- ACCESS_SEQUENCE_FILE is the name of a file containing a number of integers, one per line, each between 0 and 4999 inclusive, representing disk cylinders to be seeked in order. This file is optional. If not specified, your program should generate 100 random tracks and use that sequence instead.

Output

The output of your program will be six lines, one for each of the disk-scheduling algorithms. Each line will consist of the name of the algorithm and the total distance traveled (in terms of number of tracks). A single space should appear in between the two fields. Example:

```
$ diskSim -42 mySequenceFile.txt
FCFS 445
SSTF 389
SCAN 377
C-SCAN 397
LOOK 425
C-LOOK 399
```

Deliverables

As usual, you will be submitting just one file, **diskSim.tar.gz** and it must contain:

- all your source code files (.c, .h, .cpp, .java, .py, etc.) that are necessary to make and run ***diskSim*** on the campus CSC unix machines.
- A Makefile - if necessary- that will
    - compile the entire project with “make”, producing an executable named **diskSim**
- A README file with your name and your partners name and an explanation of how functional your program is and what (if anything) you couldn’t get to work.