

CSC 262

Homework 2

Due Sep. 25, 2008

- 1. Checksums and Error Detection** Because disk drives are subject to transient errors, values can be mis-read. These errors have to be detected and corrected (most of this happens in drive circuitry now, thankfully). For these problems we'll employ the simple checksum algorithm described in class. Recall that the checksum byte is simply the sum of all the data bytes modulo 256 ($B_E = \sum_{i=0}^n B_i \text{ mod } 256$).
 - (a) What is the checksum byte for the values: 0x12, 0x45, 0xA, 0xFF, 0x19)?
 - (b) Suppose there was a read error on the first byte and it was read as 0x13 (rather than 0x12). Now suppose there was a read error on the second byte as well (and no other subsequent bytes). What value would the second mis-read have to produce in order to still pass the checksum (i.e. the checksum would not detect this error).
 - (c) If, in a stream of N bytes (where $N > 2$), there are exactly two read errors. What is the probability that those errors will go undetected?
- 2. Disk Scheduling** For this problem you will be performing different disk scheduling routines. Assume that the disk has 64 tracks (0-63), and that the read/write heads begin on track 0. The table below lists the tracks requested and the time at which they were requested. Assume that the request buffer starts empty. Also assume that all seeks take 12ms (this simplifies your task). You will emulate several of the algorithms discussed in class. If a request arrives before the current request is finished, then it is buffered. If more than one request are in the buffer then you must make scheduling decisions in accordance with the policy of the algorithm.

Track Requested	15	13	57	20	0	30	41	9	61	11
Time of Request (ms)	0	3	7	12	15	29	31	37	51	73

For each algorithm list the order of tracks serviced, the latency (the time between the request and when it was serviced), and the number of tracks traversed. *HINT: just work through each algorithm step-by-step. When done processing a request, figure out how long it took, and which requests are outstanding. Then use the policy to select the next request.*

- (a) **FIFO**
- (b) **SSTF**
- (c) **LOOK**
- (d) **C-LOOK, increasing track#**
- (e) **C-LOOK, decreasing track#**

(f) **F-LOOK**

- (g) Which algorithm(s) took the shortest time?
- (h) Which algorithm(s) had the lowest average latency?
- (i) Which algorithm(s) had the highest latency?
- (j) Which algorithm(s) traversed the most tracks?