CSCI 3753: Operating Systems
Homework Four

Please write your answers in the space provided.
Due date: Thursday, May 01 in class. No extensions will be given, except at the instructor's discretion in documented cases of extreme hardship or emergencies.

Problem 1. [40 Points] You are asked to allocate a file according to either a File Allocation Table (FAT) or multi-level indexed allocation (UNIX inode - triply indirect). Assume that the file is 132 MB long, there are 2 KB per disk block, each pointer in the FAT occupies 4 bytes, and each index block entry takes 4 bytes.

(a) How many bytes are used to lay out the file when using a FAT file system?

Total blocks needed = 132 MB/2 KB = $66 * 2^{10}$ = 67584

FAT: Need to allocate $66 * 2^{10}$ pointers for this file = $66 * 2^{12}$ bytes = 270336 bytes

(b) How many bytes are used to lay out the file when using a UNIX-style file system?

UNIX: First 12 blocks – 12 direct pointers = 48 bytes
Next 512 (2^9) blocks – single indirect pointer = $4 + 2^{11}$ bytes
Next 67060 blocks – double indirect pointer = $4 + 2^{11} + 131 * 2^{11}$ bytes
0 blocks – triple indirect pointer = 4 bytes
Total: $60 + 133 * 2^{11}$ bytes = 272,444 bytes

Now suppose that you wish to read the 66,000'th block of the file. Assume that each of the following counts as one search operation: moving from one element to the next in a linked list; indexing into an index block; moving from index block to the next.

(c) How many searches are needed to read block 66,000 when using the FAT file system?

Need to traverse 66,000 pointers in the linked list. So, 66,000 searches in total.

(d). How many searches are needed to read block 66,000 when using the UNIX-style file system?

This block is in the doubly indirect index.
Total searches = 1 into FCB + 1 to move to first level index block + 1 to index into the first level index block + 1 to move to second level index block + 1 to index into the second level index block = 5 searches in total

Problem 2. [30 Points] Suppose you are given a flash memory consisting of 4 KB pages, and there are 1024 pages. Assume that 300 pages are currently allocated.

(a) How many bytes of memory would the OS need to keep track of free space if a bitmap is used?

1024 bits = 128 bytes

(b) How many bytes of memory would the OS need to keep track of free space if a linked list is used? Assume 2 bytes/pointer.

Total 1024 – 300 = 724 pointers. So,1448 bytes. But only 2 bytes are reserved. Rest are obtained from free blocks themselves.

(c) Under what conditions would a linked list be more memory-efficient than a bitmap?

When there are fewer free blocks.

Problem 3. [**30 Points**] A disk has 1000 cylinders, 0 – 999. Disk access requests arrive in the following order of cylinder numbers: 30 80 200 3 702 80 96 89 100. Assume that the R/W head is on cylinder number 103 initially moving towards larger numbered cylinders. Calculate the total distance (Number of cylinders) travelled by the R/W head to service these requests if

(a) FCFS algorithm is used

103 → 30 → 80 → 200 → 3 → 702 → 80 → 96 → 89 → 100
distance travelled = 73 + 50 + 120 + 197 + 699 + 622 + 16 + 7 + 11 = 1795

(b) SSTF algorithm is used

103 → 100 → 96 → 89 → 80 → 80 → 30 → 3 → 200 → 702
distance travelled = 3 + 4 + 7 + 9 + 0 + 50 + 27 + 197 + 502 = 799

(c) SCAN algorithm is used

103 → 200 → 702 → 100 → 96 → 89 → 80 → 80 → 30 → 3
distance travelled = 97 + 502 + 602 + 4 + 7 + 9 + 0 + 50 + 27 = 1298