CSIE 3310, Spring 2022: Final Exam (Phase 2) Solution Sketch

Due at 12:20noon; Marked as Last Submission after 12:20noon; Gradescope Closed at 12:25noon

0 Gradescope Submission (2pts)

When you submit your answers, select the corresponding page(s) of each question.

1 Hard Disk Drive (HDD) Scheduling (13pts)

There are 100 cylinders indexed from 0 to 99. The disk head is initially at Cylinder 30.

- If the last digit of your student ID number is 0 or 5, the disk queue with requests for I/O to blocks on cylinders is:
$$98, 16, 47, 28.$$
- If the last digit of your student ID number is 1 or 6, the disk queue with requests for I/O to blocks on cylinders is:
$$22, 90, 45, 86.$$
- If the last digit of your student ID number is 2 or 7, the disk queue with requests for I/O to blocks on cylinders is:
$$11, 85, 54, 92.$$
- If the last digit of your student ID number is 3 or 8, the disk queue with requests for I/O to blocks on cylinders is:
$$91, 42, 18, 84.$$
- If the last digit of your student ID number is 4 or 9, the disk queue with requests for I/O to blocks on cylinders is:
$$3, 96, 70, 16.$$

Use the following scheduling algorithms and answer the total movement between cylinders.

1. (1pt) Write down your student ID number.
2. (4pts) Use the the FCFS algorithm.
3. (4pts) Use the SCAN algorithm. Assume that the disk arm is moving toward 0 first, and it will reach one end before reversing the direction.
4. (4pts) Use the C-SCAN algorithm. Assume that the disk arm is moving toward 99 first, and it will reach one end before moving to the other end.

Answer:

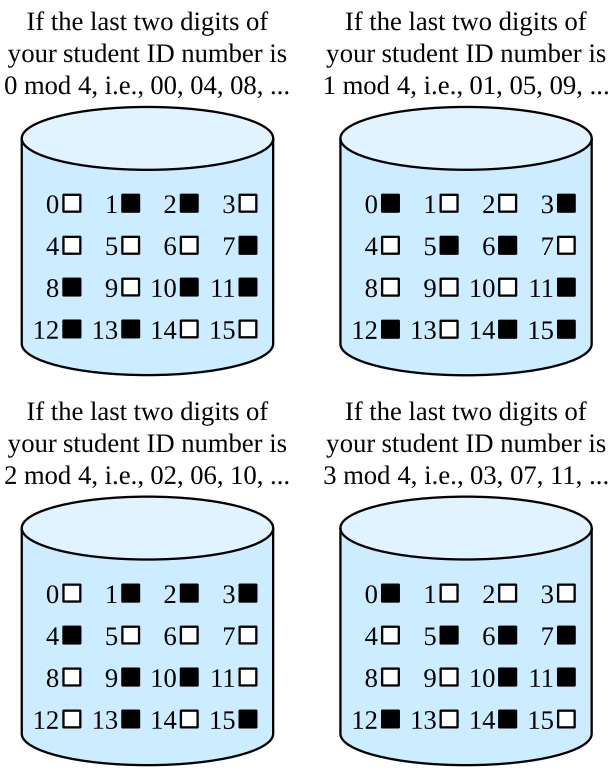
1. Omitted,

2. 200, 162, 162, 200, 200.
3. 128, 120, 122, 121, 126 from $30 +$ (the maximum number).
4. 196, 190, 179, 186, 184 from $69 + 99 +$ (the maximum number under 30).

2 File Allocation (23pts)

A block size is 64 bytes, and a pointer to a block is 8 bytes. You are allocating a 220-byte file in the following disk space, where "□" means that the block is available, and "■" means that the block is not available. Use the following allocation methods to allocate the file, where you can pick available blocks as you want, and you should assume that all allocation methods start from the same disk space as below.

For each allocation method, list the numbers of blocks that you allocate for the file and explain where the corresponding pointers are, if any. For example, you can answer: use Blocks 0, 1, 2 with the pointers in Blocks 0, 1.



1. (1pt) Write down your student ID number.
2. (3pts) Use the contiguous allocation.
3. (3pts) Use the linked allocation without clustering (do not use a file allocation table).
4. (3pts) Use the linked allocation with clustering and 2 blocks as a cluster.
5. (3pts) Use the indexed allocation.
6. (10pts) Allocate another N-byte file with an unlimited disk space.
   - The contiguous allocation allocates B_C blocks for the file.
   - The linked allocation without clustering (do not use a file allocation table) allocates $B_{L,1}$ blocks for the file and the corresponding pointers.
   - The linked allocation with clustering and 2 blocks as a cluster allocates $B_{L,2}$ blocks for the file and the corresponding pointers.
   - The indexed allocation allocates B_I blocks for the file and the corresponding pointers.

   When N is approaching infinity, list the ratio $B_C : B_{L,1} : B_{L,2} : B_I$ and explain your answer.

Answer:

1. Omitted,
2. 4 contiguous blocks.
3. 4 blocks, and 3 pointers in the first 3 blocks (or 4 pointers in the 4 blocks).
4. 2 and 2 contiguous blocks, and 1 pointer in the second block (or 2 pointers in the second and fourth blocks).
5. 4 blocks, and 4 pointers in 1 additional block.
6. $1 : \frac{8}{7} : \frac{16}{15} : \frac{8}{7}$.

3 Directory Structure (12pts)

A general-graph directory structure is modeled as a directed graph G, where a directory or a file is modeled as a vertex in G.

1. (4pts) Assume that all vertices are labeled as "NONE" initially. Write down "CONDITION1" in Line 4 to prevent an infinite search loop. Explain your answer. Requirement: all vertices which are reachable from v should still be searched (traversed).

```
1  Depth-First-Search-1(G, v){
2      label v as "DOING";
3      for each directed edge from v to u in G
4          if CONDITION1
5              Depth-First-Search-1(G,u);
6      label v as "DONE";
7  }
```

2. (4pts) Assume that all vertices are labeled as "NONE" initially. Keep your "CONDITION1" (from Question 1) in Line 6 here and write down "CONDITION2" in Line 4 to replace edges which cause cycles. Explain your answer. Requirement: the number of replaced edges should be minimized.

```
1  Depth-First-Search-2(G, v){
2      label v as "DOING";
3      for each directed edge from v to u in G
4          if CONDITION2
5              replace (v, u) by (v, v'), where v' is a new vertex labeled as "NONE";
6          else if CONDITION1
7              Depth-First-Search-2(G,u);
8      label v as "DONE";
9  }
```

3. (4pts) How do you use Depth-First-Search-2(G, v) to remove all cycles in G so that you can turn a general-graph directory structure into an acyclic-graph directory structure? If you need other data structures or functions, explain them clearly.

Answer:

1. u is "NONE" with sufficient explanation.
2. u is "DOING" with sufficient explanation.
3. If G has the information of all vertices, then, for each "NONE" vertex v in G, execute Depth-First-Search-2(G, v). One can also perform garbage collection to remove non-referable directories and files and then perform Depth-First-Search-2(G, v) for vertices at root.