CMPUT 379 – Operating System Concepts
Practice Final Exam - Fall 2019
Department of Computing Science, University of Alberta
Instructor: Omid Ardakanian

Full Name: _____

Student ID Number: _____

Instructions

- Print your full name and ID clearly above.
- You have 120 minutes to complete the exam.
- There should be 6 questions and 8 pages in this exam booklet. You are responsible for checking that your exam booklet is complete.
- It is a closed-book exam. You are not allowed to bring printed or handwritten notes. Electronic devices such as laptops, calculators, phones, etc. are strictly prohibited.
- Place answers in the spaces provided on the question pages. Keep your answers brief. Think about each question a bit before answering. If required, you may use the backside of a sheet to provide an answer but clearly indicate when you have done so.
- This exam is worth 100 points and counts 35% toward your final grade in this course. The weight of each question is indicated in square brackets by the question number.

Question	1	2	3	4	5	6	Total
Out of	15	10	30	15	10	20	100

Question 1 [15 points]: Operating System Concepts

Choose either True or False for the questions below. You do not need to provide justifications.

(1) An operating system provides a specific set of services to user programs through system calls regardless of the programming language they are written in.

☐ True ☐ False

(2) Named pipe cannot be used by two processes that do not have a parent-child relationship.

☐ True ☐ False

(3) Round Robin scheduling may lead to starvation.

☐ True ☐ False

(4) A thread pool can prevent an overly popular web site from crashing the hosting web server.

☐ True ☐ False

(5) Deadlock will eventually happen when a system is in an unsafe state.

☐ True ☐ False

(6) Increasing the page size will increase external fragmentation.

☐ True ☐ False

(7) The number of page faults will certainly decrease as we increase the number of page frames.

☐ True ☐ False

(8) A Translation Lookaside Buffer (TLB) caches frequently accessed page table entries.

☐ True ☐ False

(9) Thrashing occurs when the sum of the working sets of all processes exceeds the available memory.

☐ True ☐ False

(10) A disk can have multiple partitions, each having a different file system.

☐ True ☐ False

(11) A file descriptor is an index into the system-wide open file table.

☐ True ☐ False

(12) Indexed allocation has a higher block-pointer overhead than linked allocation for small files.

☐ True ☐ False

(13) Random-access I/O is faster on NVM drives than HDDs.

☐ True ☐ False

(14) The SSTF algorithm for disk scheduling yields the minimum total length of disk seeks needed to service a given set of I/O requests.

☐ True ☐ False

(15) The storage overhead of RAID 1 is higher than RAID 5.

☐ True ☐ False

Question 2 [10 points]: Multiple Choice Questions
You must select the best answer from the choices given.

Part 2.1 [2 points]: Which scheduling algorithm makes the most sense for NVM devices?

- ○ SCAN.
- ○ Shortest Seek Time First (SSTF).
- ○ C-SCAN.
- ○ First Come, First Served (FCFS).

Part 2.2 [2 points]: Which memory allocation strategy picks the largest hole?

- ○ First fit.
- ○ Best fit.
- ○ Worst fit.
- ○ None of the above.

Part 2.3 [2 points]: Which of the following page replacement algorithms approximates the Least Recently Used (LRU) page replacement algorithm more closely?

- ○ First-In First-Out.
- ○ Most Frequently Used.
- ○ Second Chance.
- ○ Additional-Reference-Bits.

Part 2.4 [2 points]: Which of the following scheduling algorithms is non-preemptive?

- ○ First-Come First-Served.
- ○ Shortest Remaining Time First.
- ○ Round Robin.
- ○ Multilevel Feedback Queue.

Part 2.5 [2 points]: How does the effective memory access time on a TLB miss of a system with a 4-level page table compare with that of a system with a 2-level page table?

- ○ 5/3x slower.
- ○ 2x slower.
- ○ 5/3x faster.
- ○ 2x faster.

Question 3 [30 points]: Short-Answer Questions

Part 3.1 [10 points]: Suppose n periodic tasks are admitted to a hard real-time system. A new task (indexed by $n + 1$) is submitted to this system. How should we decide whether to reject this task or not? **Hint:** recall that a periodic task i is characterized by its deadline d_i, its fixed processing time t_i, and its period p_i.

Part 3.2 [5 points]: Why could adding more memory to a computer make its processes run faster?

Part 3.3 [5 points]: What are the difference between segmentation and paging? Give two differences.

Part 3.4 [10 points]: Consider a file system which has 32-bit file pointers and blocks of size 512 bytes. Each inode contains 16 direct blocks and 2 single indirect blocks. What is the maximum possible file size that can be supported?

Question 4 [15 points]: Process Synchronization

Part 4.1 [10 points]: Build a mutex lock using a POSIX semaphore by implementing the three functions defined blow.
Hint: Recall that a POSIX semaphore must be initialized using `sem_init` before it can be operated on using `sem_post` and `sem_wait`.

```
struct lock_t {
 sem_t asem;
};

void lock_init(lock_t *lock) {




}

void acquire(lock_t *lock) {




}

void release(lock_t *lock) {




}
```

Part 4.2 [5 points]: Briefly explain the difference in behaviour of `sem_post` and `pthread_cond_signal` when no threads are waiting in the corresponding semaphore or condition variable.

Question 5 [10 points]: Page Replacement Policies

Consider the following page reference string in a demand-paged virtual memory system with 3 page frames that are initially empty:

```
12, 8, 3, 7, 8, 3, 9, 10, 3, 5, 7, 10, 3, 7, 5, 7, 12, 10, 7, 5, 12
```

Part 5.1 [5 points]: How many times will page fault occur under the LRU policy? Use the following table to keep track of the pages that are loaded in the page frames in each step. If there is a tie, break it by replacing the oldest page.

Request	12	8	3	7	8	3	9	10	3	5	7	10	3	7	5	7	12	10	7	5	12
Frame 1																					
Frame 2																					
Frame 3																					

Part 5.2 [5 points]: How many times will page fault occur under the optimal policy? Use the following table to keep track of the pages that are loaded in the page frames in each step. If there is a tie, break it by replacing the oldest page.

Request	12	8	3	7	8	3	9	10	3	5	7	10	3	7	5	7	12	10	7	5	12
Frame 1																					
Frame 2																					
Frame 3																					

Question 6 [20 points]: File System and Disk

Part 6.1 [10 points]: Consider a file that is allocated 100 data blocks on disk which are organized as a linked list, i.e., the pointer to the next file block is stored at the end of each block. Suppose a data block is appended to the end of the file which causes the file size to grow to 101 blocks. How many disk read and write operations are required to update the file and the file control block?
You should assume that each operation reads or writes a single disk block, both the file and the file control block are initially on disk, and the file buffer cache is empty.

Part 6.2 [10 points]: Answer the same question this time for the case that an index block contains all block pointers rather than storing them at the end of data blocks (i.e., indexed allocation is used instead of linked allocation).