

Please print in pen:

Waterloo Student ID Number:

--	--	--	--	--	--	--	--

WatIAM/Quest Login UserID:

--	--	--	--	--	--	--	--

UNIVERSITY OF
WATERLOO



Final Exam - Winter 2019 - SE 350

1. Before you begin, make certain that you have one **2-sided booklet with 12 pages**. You have **150 minutes** to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points for that question.
2. Please read all of the questions before starting the exam, as some of the questions are substantially more time consuming. Read each question carefully. Make your answers as concise as possible. **If there is something in a question that you believe is open to interpretation, then please ask us about it!**
3. All solutions must be placed in this booklet. If you need more space to complete an answer, you may be writing too much. However, if you need extra space, use the blank space on the last page of the exam clearly labeling the question and indicate that you have done so in the original question.

Good Luck!

Question	Points Assigned	Points Obtained
1	34	
2	18	
3	17	
4	12	
5	19	
Total	100	

1. (X points) True-False and Why? For each question:

- CIRCLE YOUR ANSWER
- One point for correct true-false.
- One point for correct explanation.
- No points for any explanation if true-false is not correct.
- No points for an explanation that exceeds 3 sentences.

1.a. If each waiting thread in a system is waiting for a resource held by another waiting thread, then the system is in deadlock.

True False Why?

1.b. For the Communal Dining Politicians problem where there are N chopsticks in the middle of a table and N politicians who can grab one chopstick at a time, there are N unsafe states.

True False Why?

1.c. In a uniprocessor, for a workload of tasks with fixed sizes, work-conserving schedulers achieve better throughput than non-work conserving schedulers.

True False Why?

1.d. Assuming zero overhead for Round Robin time slicing, FIFO is always better than Round Robin for average response time.

True False Why?

1.e. The execution time of a parallel job could be more than 9.2 times faster on ten cores than on one core if only 99% of the job's execution time can be parallelized.

True False Why?

1.f. With virtual memory, the memory content of a user process is hidden from other user processes.

True False Why?

1.g. In a segmented memory, different processes can access the same physical address using different virtual addresses.

True False Why?

1.h. Using TLBs could increase the cost of address translation for some processes and decrease it for others.

True False Why?

1.i. On a context switch, TLB does not have to be flushed.

True False Why?

1.j. Any modification to page table entries in a multiprocessor necessarily requires a TLB shutdown.

True False Why?

1.k. In a system with *base* and *bound* address translation, virtually addressed caches do not suffer from the aliasing problem.

True False Why?

1.l. If no new task arrives, SJF scheduler never preempts currently running task.

True False Why?

1.m. Threads within the same process can share data with one another by passing pointers to objects on their stacks.

True False Why?

1.n. With copy-on-write, immediately after a process has been forked, the same variable in both the parent and the child will have the same virtual address but different physical addresses.

True False Why?

1.o. Disabling interrupts on any computer system that supports it guarantees atomicity.

True False Why?

1.p. Switching the order of two P() semaphore primitives can lead to deadlock.

True False Why?

1.q. The function `thread_create()` will always ensure that another thread starts running.

True False Why?

2. (18 points) Uniprocessor Scheduling.

2.a. (9 points). Given the following mix of tasks, task lengths, and arrival times, compute the completion time for each task for the FIFO, RR, and SJF algorithms. Assume a zero-cost time slicing of 10 milliseconds and that all times are in milliseconds. For RR, arriving tasks has lower priority than tasks that are already waiting. If a time slice completes at the same time that a job arrives, assume the arriving job has lower priority than tasks that are already waiting, but higher priority than the task completing its quantum.

Task	Length	Arrival Time	FIFO	RR	SJF
0	85	0			
1	30	10			
2	35	15			
3	20	80			
4	50	85			

Show your work here:

2.b (9 points) Consider the following preemptive priority-scheduling algorithm based on dynamically changing priorities. Larger numbers imply higher priority. Tasks are preempted whenever there is a higher priority task. Assume that time is divided into 1 millisecond time quanta and tasks do not arrive in the middle of a time quantum. When a task is waiting for CPU (in the ready queue, but not running), its priority changes at rate a : $P(t) = P(t - 1) + a$, and when it is running, the task's priority changes at rate b : $P(t) = P(t - 1) + b$. Suppose that $P(t) = 0$ for all $t \leq t_0$, where t_0 is the time at which the task joins the ready queue.

2.b.1 (3 points). What is the algorithm that results from $b > a > 0$?

2.b.2 (3 points). What is the algorithm that results from $a < b < 0$?

2.b.3 (3 points). Suppose that tasks retain their priority when they are preempted. What happens if two tasks arrive at nearly the same time and $a > 0 > b$?

3. (17 points) Magnetic Disk. Consider a magnetic disk with the following spec.

Seek time from middle to outer track	10.3 ms
Seek time from middle to inner track	10.1 ms
Seek time from inner to outer track	19 ms
Average seek time	10.5 ms
Rotation time	8.3 ms
Transfer rate	54 - 128 MB/s
Bytes per sector	512

3.a. (6 points) SPTF is not optimal. Assume that the disk's head is on the middle track. Suppose that there are two sets of pending requests. The first set is 1000 requests to read each of the 1000 sectors on the inner track of the disk; the second set is 2000 requests to read each of the 2000 sectors on the outer track of the disk. Compare the average response time per request (i.e., the time for a request to complete, from when a request arrives until it is done, **excluding the transfer time**) for the SPTF schedule (first read the "nearby" inner track and then read the outer track) and the alternative of reading the outer track first and then the inner track. Write your final answer in the boxes.

SPTF:

Outer-track-first:

Show your work here:

3.b. (3 points) Sequential Access. Now suppose that the head of the disk is on a random track. Consider 1000 read requests for sequential sectors on the outer track. How long does serving these requests take on average (considering the transfer time and out of order read)? Write your answer in the box.

Average sequential access time:

Show your work here:

3.c. (2 points). Effective Bandwidth. In 3.b. what fraction of the disk's bandwidth is realized? Write your answer in the box.

Effective bandwidth:

Show your work here:

3.d. (6 points) Linked List vs. Tree. Consider two different file systems FAT and FFS. For FFS, there are twelve direct pointers, one indirect pointer, one double indirect pointer, and one triple indirect pointer. Suppose that data blocks are 4KB in both file systems and suppose that FFS has 4-byte block pointers. Suppose that we create a new file, write 4 KB at offset 0, seek to block offset 2^{17} , and write another 4 KB. How many blocks does each file system use to store the file? Write your answer in the boxes.

FAT:

FFS:

Show your work here:

4. (12 points) Address Translation. Consider a system with the following parameters.

Variable	Measurement	Value
P_{TLB}	Probability of TLB hit	0.95
P_{L1}	Probability of a first-level cache hit for all accesses	0.99
P_F	Probability of a page fault when a TLB miss occurs on user pages (assume page faults do not occur on page tables).	0.001
T_{TLB}	Time to access TLB	1ns
T_{L1}	Time to access L1 cache	10 ns
T_M	Time to access DRAM	250 ns
T_D	Time to transfer a page to/from disk	10^7 ns

Suppose that the system has a 3-level page table that is stored in DRAM and are cached like other accesses. Also assume that the costs of the page replacement algorithm and updates to the page table are included in the T_D measurement. Suppose that pages mapped on a page fault are not cached and hardware automatically fills TLB on a miss.

4.a. (4 points). How long does it take for a user program to do one memory reference if the address translation is cached in TLB? Write your answer in the box.

Show your work:

Time:

4.b. (4 points). How long does it take for a user program to do one memory reference if the address translation is not cached in TLB? Write your answer in the box.

Show your work:

Time:

4.b. (4 points). How long does it take for a user program to do one memory reference? Write your answer in the box.

Show your work:

Time:

5. (19 points) Locks and Deadlocks.

5.a. (5 points) Spinlock Using Swap. We want to implement locks using the swap() primitive. swap() has the following semantics, and is executed atomically:

```
void swap(int *a, int *b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```


b) During the second phase, any lock may be released, and lock E may be downgraded from write mode to read mode; releases and downgrades can happen in any order; by the end of part 2, all locks must be released; and no locks may be acquired or upgraded.

These rules ensure freedom from deadlock.

True False Why?

5.c. (10 points) Banker's Algorithm. Suppose there are three jobs A, B, and C running in a multi-core processor. Each job requires 4 resources (CPUs, last-level cache, memory capacity, and memory bandwidth) to run. Suppose that the multi-core processor has 16 cores, 12 MB last-level cache, 32 GB memory, and 70 GB/s memory bandwidth. For each job, the maximum possible usage for each resource is specified in the table below.

Job	CPU	Cache (MB)	Mem. Capacity (GB)	Mem. Bandwidth (GB/s)
A	4	8	20	16
B	6	6	10	32
C	10	4	14	36

5.c.1. (5 points). Consider the following allocation. Is the system in a safe state? If so, give an example guaranteed safe execution. If not, give an example of requests that could deadlock the system. Circle your answer and show your work.

Job	CPU	Cache (MB)	Mem. Capacity (GB)	Mem. Bandwidth (GB/s)
A	2	2	10	8
B	3	5	8	30
C	8	2	12	20

Yes No Why?

5.c.2. (5 points). Consider the following safe allocation. Is it safe to allocate to C four more cores? If so, give an example guaranteed safe execution. If not, give an example of additional requests that could deadlock the system. Show your work.

Job	CPU	Cache (MB)	Mem. Capacity (GB)	Mem. Bandwidth (GB/s)
A	3	7	18	16
B	4	3	6	20
C	5	1	6	34

Yes

No

Why?