Please print in pen:

Waterloo Student ID Number:

WatIAM/Quest Login UserID:

UNIVERSITY OF WATERLOO

Final Exam - Spring 2026 - ECE 350

1. Before you begin, make certain that you have one **2-sided booklet with 16 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 write your interpretation and assumptions!**

3. All solutions must be placed in this booklet. If you need more space to complete an answer, you may be writing too much. If you need extra space, use the blank space on the last few pages of the exam, clearly label the question there, and note in the original question that you have done so.

Good Luck!

Question	Points Assigned	Points Obtained
1	56	
2	18	
3	12	
4	14	
Total	100	

Q1. (56 points) True/False with Explanation.

For each question:

- Circle True or False and write your explanation below it.
- Explanations must not exceed three sentences.
- One point is awarded for the correct True/False selection.
- One point is awarded for a correct explanation.
- No explanation point is awarded if the True/False selection is incorrect.

1. When a system call passes arguments through a buffer in user memory, the kernel copies them into kernel memory before validating them.

   True False

2. Within the same process, one thread cannot access another thread's stack.

   True False

3. An interrupt handler is a kernel thread that runs at the highest priority.

   True False

4. I/O requests cannot be completed without first being processed by a device driver.

   True False

5. For a fixed number of processes on a uniprocessor, improving throughput will improve the response time of at least one process.

   True False

6. On a multiprocessor, different threads of the same process may run simultaneously on different processors.

   True False

7. A Mesa monitor is easier to implement but harder to reason about than a Hoare monitor.

   True False

8. Round-robin (RR) scheduling can be implemented on a system that does not support interrupts.

   True False

9. If 99% of a program is perfectly parallelizable and there is no parallelization overhead, Amdahl's law predicts a speedup strictly greater than 9.2x on 10 cores.

   True False

10. In the MSI protocol, if a cache block is in the Modified state in one cache, it must be Invalid in all other caches.

True False

11. Assuming instructions are never written at run time (no self-modifying or dynamically generated code), cache coherence is not needed for instruction caches.

True False

12. Directory-based coherence protocols are more scalable than snooping protocols.

True False

13. In a perfectly load-balanced multiprocessor, work stealing is not required.

True False

14. With RCU locks, readers that begin reading the shared data during the grace period may or may not see the new data.

True False

15. For a workload with total utilization of 1, a rate-monotonic (RM) scheduler cannot produce a feasible schedule.

True False

16. With multi-segment address translation, the same physical address can be mapped to different virtual addresses in two different processes.

True False

17. Right after a process is forked, the same global variable in both parent and child has the same physical address if copy-on-write is implemented.

True False

18. Page-table address translation eliminates external fragmentation.

True False

19. It is possible to protect programs from one another without address translation.

True False

20. For a sparse virtual address space, a multi-level page table is generally more memory-efficient than a single-level page table containing an entry for every virtual page.

True False

21. Consider three periodic tasks, written as (period, execution time), with deadlines equal to periods: {T1: (4, 1), T2: (6, 3), T3: (8, 2)}. An EDF scheduler will generate a feasible schedule.

True False

22. A two-way set-associative cache can cache two addresses that map to the same cache index.

True False

23. Earliest-deadline-first scheduling can still meet all deadlines for any task set with utilization ≤ 1 when it operates non-preemptively.

True False

24. TLB entries generally include neither an access bit nor a dirty bit.

True False

25. The overhead of updating PTEs is the same in a multiprocessor and a uniprocessor.

True False

26. To reduce TLB hit time, TLBs are usually direct-mapped or have low associativity.

True False

27. Demand paging requires a process to be fully resident in physical memory before it can execute.

True False

28. When a keystroke occurs, the kernel schedules an interrupt handler to process it.

True False

Q2. (18 points) Multiple-Select Questions.

1. (2 points) Consider a thread that is executing a busy-wait loop while waiting for another thread. Assume a preemptive scheduler. While the thread is still logically in the busy-wait loop (i.e., it has not yet exited the loop), which of the following may be its scheduler state? Select all that apply.

   ☐ Running
   ☐ Waiting
   ☐ Ready
   ☐ Finished

2. (2 points) Which of the following can be produced by the program below? Select all that apply. Assume no errors occur.

```
void *helper(void *arg) {
    printf("b");
    return NULL;
}

int main() {
    pthread_t thread;
    pthread_create(&thread, NULL, &helper, NULL);
    pthread_yield();
    printf("a");
    return 0;
}
```

   ☐ ab
   ☐ ba
   ☐ b
   ☐ a

3. (2.5 points) Select all correct statements.

   ☐ In a paged address-translation system, the virtual and physical addresses use the same number of page-offset bits.
   ☐ Multi-segment address translation eliminates external fragmentation.
   ☐ Page-table entries in two different processes can map virtual pages to the same physical page frame.
   ☐ An inverted page table always contains fewer entries than the combined multi-level page tables for the same system.
   ☐ The virtual address space is always larger than the physical address space.

4. (2 points) Assume that the address space contains sufficient unused room and that the OS may update the relevant bound, segment-table entries, or page-table entries as the process grows. Which address-translation schemes can represent a process whose heap and stack grow over time? Select all that apply.

☐ Base-and-bound address translation.
☐ Multi-segment address translation.
☐ Single-level page-table address translation.
☐ Multi-level address translation.

5. (2.5 points) Select all correct statements.

☐ Allowing a system-call handler to be preempted at safe points can reduce worst-case scheduling latency in a real-time system.
☐ A real-time task can comprise multiple jobs.
☐ For independent periodic tasks with relative deadlines equal to periods on a fully preemptive uniprocessor, RM is optimal among fixed-priority assignments.
☐ If a polling server is scheduled by RM and all other parameters remain unchanged, reducing its period increases the server's fixed priority.
☐ For independent, fully preemptive jobs on one processor with zero scheduling overhead, EDF and LLF are feasibility-optimal online scheduling policies.

6. (2 points) With the MSI protocol, when is a cache controller forced to write back a cache line B? Select all that apply.

☐ B is in the S state, and another cache requests Get-M for B.
☐ B is in the M state, and another cache requests Get-M for B.
☐ B is in the S state, and it is evicted to make room for another block.
☐ B is in the M state, and it is evicted to make room for another block.

7. (3 points) Which of the following statements about the FAT file system are correct? Select all that apply.

☐ A quick format zeroes all data blocks and marks every FAT entry as free.
☐ FAT supports hard links, so the same file can be reached through several directory entries.
☐ FAT enforces per-user access rights and uses transactional updates to survive crashes.
☐ With 4 KiB blocks, a FAT volume can grow beyond 1 TiB because block addresses are encoded using 32 bits.
☐ FAT provides fast random access to blocks in large files, because each file's block addresses are held in a direct index.
☐ The FAT is stored on disk; the OS may cache FAT sectors in memory, and conventional FAT volumes commonly maintain a second on-disk FAT copy.

8. (2 points) Which of the following statements are **FALSE**? Select all that apply.

☐ Demand paging allows a process to be only partially resident in physical memory.
☐ Fixed-size pages allow the kernel to use any free physical page frame for any virtual page.
☐ Under memory overcommit, paging may avoid the need to swap entire processes out to disk.
☐ Pure segmentation allows the kernel to evict a single segment while keeping the rest of the process running.

Q3. (12 points) Multiple-Choice Calculations.

For each question:

- Select one answer.
- Show your work.
- Two points are awarded for the correct selection.
- Two points are awarded for correct work.
- No work points are awarded if the selection is incorrect.

Consider the following two-level page-table scheme, which is used for the next three questions. The virtual-address, physical-address, and page-table-entry formats are shown below.

Virtual address:

Virtual Page #1	Virtual Page #2	Offset
8 bits	12 bits	12 bits

Physical address:

Physical Page #	Offset
20 bits	12 bits

Page-table entry (PTE):

Physical Page #	OS-defined bits	Unused bits	Control & status bits
20 bits	3 bits	2 bits	7 bits

1. What is the size of a single page? **Show your work**.

☐ 1 KiB
☐ 2 KiB
☐ 4 KiB
☐ 8 KiB

2. What is the total maximum size of the page table (in bytes) for this scheme, i.e., the full multi-level structure that maps virtual to physical addresses? Here, "size" means the total number of bytes occupied by entries; ignore page-alignment and allocator padding. **Show your work.**

- ☐ 4194304
- ☐ 4198400
- ☐ 4195328
- ☐ 16777216

3. What is the maximum amount of physical memory addressable with this scheme? **Show your work.**

- ☐ 1 GiB
- ☐ 4 GiB
- ☐ 16 GiB
- ☐ 128 GiB

Q4. (14 Points) Polling Server, Deferrable Server, and Total Bandwidth Server

Consider the following periodic tasks. Each task has a relative deadline equal to its period and releases its first job at t = 0:

Task	Execution time	Period
T1	1	4
T2	2	8

The following aperiodic jobs each arrive once and have no hard deadline of their own:

Job	Arrival	Execution time
A1	1	1
A2	5	2
A3	12	1

Assumptions:

- Single processor.
- All periodic tasks and aperiodic jobs are fully preemptive.
- Preemption, context-switch, and dispatch costs are zero.
- All events occur at integer time instants.
- Scheduling decisions are made only at integer time instants.
- At any given instant, events are processed in the following order:
- (i) job completions, (ii) server budget replenishments, (iii) periodic-task releases and aperiodic-job arrivals, and (iv) the scheduling decision. Therefore, an aperiodic job arriving exactly at time t is considered pending at t.
- Multiple pending aperiodic jobs are served in FIFO order of arrival: A1 before A2 before A3.
- For the polling server (PS) and deferrable server (DS), the server budget is 2 and the server period is 6 (first budget refill at t = 0), and tasks are scheduled using rate monotonic (RM) policy.
- The total bandwidth server (TBS) uses bandwidth of 1/3.
- EDF tie-breaking rule: the ready entity with the earlier absolute deadline is selected. If two ready entities have the same absolute deadline, the one that became ready earlier is selected. If they are still tied, the periodic task is selected.

1. (2 Points) What are the deadlines assigned to each aperiodic job under TBS. Show your work.

Aperiodic job	Assigned deadline
A1	
A2	
A3	

2. (12 Points) What is the **execution schedule on [0, 16)?** For each unit interval write exactly one of: T1, T2, A1, A2, A3, or idle.

Interval	Polling Server system	Deferrable Server system	TBS system
[0,1)			
[1,2)			
[2,3)			
[3,4)			
[4,5)			
[5,6)			
[6,7)			
[7,8)			
[8,9)			
[9,10)			
[10,11)			
[11,12)			
[12,13)			
[13,14)			
[14,15)			
[15,16)			