Please print in pen:

Waterloo Student ID Number:

WatIAM/Quest Login UserID:

UNIVERSITY OF WATERLOO

Midterm Exam - Spring 2026 - ECE 350

1. Before you begin, make certain that you have one **2-sided booklet with 12 pages**. You have **90 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. 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	**70**	
2	**30**	
Total	**100**	

Q1. (70 points) True-False with explanation.

For each question:

- Circle your answer and write your explanation below each question.
- Explanations should not exceed 3 sentences.
- One point for correct true-false.
- One point for correct explanation.
- No points for any explanation if true-false is incorrect.

1. For every system call, the kernel validates the user-provided arguments before copying them from user-space memory into kernel memory.

   True False

2. Within the same process, one thread can access another thread's stack.

   True False

3. Within the same process, different threads share the same heap.

   True False

4. Context switching between threads that belong to different processes has higher overhead than context switching between threads of the same process.

   True False

5. System calls allow a user process to directly access the kernel's memory.

   True False

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

   True False

7. Hyperthreading is the functionality that lets multiple threads, each with its own registers and program counter, share the functional units of a single pipeline.

   True False

8. An I/O request can be completed without ever being processed by a device driver.

   True False

9. With POSIX threads, if the main thread calls exit(), all threads in the process terminate.

   True False

10. A thread's state can change directly from “ready” to “finished.”

    True False

11. The maximum number of virtual addresses depends on the processor/ISA's address width (the width of the memory bus).

    True False

12. During the boot process the kernel is loaded into memory by the bootloader.

    True False

13. The command-line arguments passed when a program is invoked are stored at lower addresses in the virtual address space than any variable allocated on the heap.

    True False

14. All implementations of a mutex on a uniprocessor involve busy waiting.

    True False

15. A simple test-and-set spinlock on a multiprocessor involves busy waiting.

    True False

16. Immediately after a thread calls thread_yield(), another thread always starts running.

    True False

17. A thread can always enforce mutual exclusion by disabling interrupts on the processor it is running on.

    True False

18. Without atomic read-modify-write instructions, mutual exclusion cannot be implemented on multiprocessors.

    True False

19. A thread created by the thread_create() system call is guaranteed to run.

    True False

20. A user-space thread running inside a mutex-protected critical section cannot be context-switched.

    True False

21. On a uniprocessor, first-come-first-served (FCFS) scheduling minimizes average response time when all tasks have equal length.

    True False

22. For a workload on a uniprocessor, scheduler A achieves lower average response time than scheduler B. Therefore, scheduler A must have higher throughput than scheduler B on that workload.

    True False

23. Interrupt handlers can use spinlocks.

    True False

24. A thread can wait on multiple condition variables simultaneously.

    True False

25. For a set of tasks with fixed execution times on a uniprocessor with zero context-switch overhead, the throughput of a work-conserving scheduler could be worse than that of a non-work-conserving scheduler.

    True False

26. On a uniprocessor, a user-space library can implement its own mutex so that locking the mutex does not require a mode switch into the kernel.

    True False

27. Assuming there is no context-switching overhead, under round-robin (RR) scheduling, changing the time quantum does not affect the response time of the longest task.

    True False

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

    True False

29. Lottery scheduling becomes RR when all tasks have the same number of tickets.

    True False

30. To improve average response time with lottery scheduling, we can give each job a number of tickets proportional to its runtime.

    True False

31. Assuming there is no context-switching overhead, first-come-first-served (FCFS) scheduling is always better than RR for average response time.

    True False

32. In stride scheduling, a task with a lower stride is given more CPU time than a task with a higher stride.

    True False

33. In lottery scheduling, the ratio of the response times of two tasks with the same number of tickets and the same length, arriving at the same time, is close to one.

    True False

34. Given a fixed tie-breaking rule, weighted max-min-fair scheduling and stride scheduling produce the same schedule when tickets are assigned in proportion to weights and the stride-scheduling time quantum equals the target latency in weighted max-min fairness.

    True False

35. Lottery scheduling can prevent CPU starvation by giving at least one ticket to each task.

    True False

Q2 (30 points) Multiple Select

1. (3 points) Select the thread attributes that require hardware support to be saved safely during a user-to-kernel mode switch in x86 systems.

    - ☐ Program counter (PC)
    - ☐ EFLAGS register
    - ☐ File descriptors
    - ☐ General-purpose registers
    - ☐ Memory map
    - ☐ Stack pointer (SP)

2. (2 points) Which of the following may bring the processor from kernel mode to user mode? (select all that apply)

    - ☐ A new process or thread begins executing
    - ☐ A user-level upcall
    - ☐ An interrupt, exception, or system call
    - ☐ A process/thread context switch that resumes a user thread

3. (3 points) Select all correct statements about the virtual address-space layout of a typical C program.

- [ ] The stack grows from higher addresses toward lower addresses.
- [ ] The heap starts at address zero.
- [ ] Arrays are always stored on the heap.
- [ ] The stack and heap grow in opposite directions.
- [ ] The space between the stack and heap is always kept empty.
- [ ] Local variables of a function are stored on the stack.

4. (2 points) A thread that calls which of the following could continue running without being context-switched? (select all that apply)

- [ ] pthread_create()
- [ ] pthread_yield()
- [ ] pthread_exit()
- [ ] pthread_join()

5. (2 points) On a processor that does not support interrupts, which of the following cannot happen or work properly? (select all that apply)

- [ ] Programmed I/O
- [ ] Direct memory access (DMA)
- [ ] Voluntary context switch
- [ ] Involuntary context switch

6. (2 points) What are the disadvantages of protecting a critical section by disabling interrupts? (select all that apply)

- [ ] User-level code cannot use it.
- [ ] It can delay the handling of time-critical events.
- [ ] It wastes CPU cycles.
- [ ] It does not work well on multiprocessor systems.

7. (2 points) Which of the following statements about semaphores are INCORRECT? (select all that apply)

- [ ] A semaphore can be initialized to any value between -2^{31} and $2^{31} - 1$.
- [ ] Semaphores cannot be used by interrupt handlers.
- [ ] A condition variable can be implemented with a single semaphore.
- [ ] A mutex can be implemented with a single semaphore.

8. (3 points) Which of the following statements about Mesa monitors are correct? (select all that apply)

☐ A thread that calls signal() keeps the mutex locked and does not yield the processor.
☐ A thread should call wait() inside an if statement.
☐ A thread that wakes up inside wait() is suspended again before returning if another thread holds the mutex.
☐ The thread that returns from wait() holds the mutex.
☐ A thread placed on the ready queue by a signal always runs immediately after the signaler is context-switched.
☐ The mutex is released after the thread calling wait() is placed on the waiting queue.

9. (3 points) Which of the following are advantages of FCFS over RR? (assume nonzero context-switching overhead; select all that apply)

☐ It achieves better or equal throughput.
☐ It achieves lower or equal context-switching overhead.
☐ It achieves better cache reuse by running each job to completion.

10. (2 points) Two threads run simultaneously with the program orders shown below. Which of the following global execution orders are consistent with sequential consistency? (select all that apply)

Thread A (CPU 1)	Thread B (CPU 2)
Load1	Load4
Load2	Store1
Load3	Load5

☐ Load1, Load2, Load3, Load4, Store1, Load5
☐ Load1, Load4, Load2, Store1, Load3, Load5
☐ Load4, Store1, Load5, Load1, Load2, Load3
☐ Load1, Store1, Load2, Load3, Load4, Load5

11. (6 points) Consider the single-threaded processes in the table below. Select all correct options. (For RR, at the end of each quantum the running process is placed at the back of the ready queue first, and only then are any newly arrived processes appended.)

Process	Arrival time (s)	Burst time (s)
P1	0	2
P2	1	6
P3	4	1
P4	7	4
P5	8	3

- ☐ Average response time of FCFS = 5.2.
- ☐ Average response time of SRTF = 4.8.
- ☐ Average response time of RR with quantum 1s = 6.4.