Concordia University
Department of Computer Science & Software Engineering

Comp 346, Winter 2017
Instructor: Dr. Aiman Hanna
Time: 60 minutes

Mid-term Exam

Name: _____

ID. #: _____

- **Keep your answer very organized & clean.**
- **Exam will be marked out of 20. Exam has 7 pages.**

Question # 1 (3 marks)
Indicate whether each of the following statements is *true* or *false*. **Explain** your answer.

i) If a system is guaranteed to be used by a single-user, there is no real need for the *mode bit*.

OTrue OFalse

ii) The *Many-to-Many* user-kernel thread model results in the system being less scalable (in terms of supporting more/less user threads) than the *One-to-One* thread model.

OTrue OFalse

iii) Multitasking enables multiple users to run multiple programs simultaneously. This however cannot be achieved if the system is a uni-programming system.

OTrue OFalse

Question # 2 (2 marks)

A) Assume that you are a member of an OS design team. Assume also that all critical sections in the systems are guarded by semaphores, and hence all accesses to critical sections are protected correctly. You and your teammates developed a novel technique that allows the system to detect any fatal error by any running process and terminates the process. All members of the team argued that the developed technique must be applied to any/all processes at all times regardless of what section of the code the process is executing. The team leader however rejected that proposal and indicated that the technique should not be applied if a process is executing a critical section. However, the team leader did not provide any explanation that supports his position; rather he asked you if you can argue on his behalf!

Are there any valid arguments that you can raise to support your team leader? If *yes*, explain why the team leader has a valid point, and indicate what type of problems can occur if the technique is applied at all times. If *no*, explain why it makes more sense that the technique must be applied at all times (do not worry about risking your job; there is no problem in that regard!).

Answer: ☐ I **would** support the team leader ☐ I would **NOT** support the team leader

Explanation: (**Important**: do NOT use the entire space below; you do NOT need it!)

Question # 3 (2 marks)

Assume a system is capable of supporting interrupts, and is also capable of polling all devices. The initial configuration of the system utilized interrupts, however the system performance was low. One of the designers studied the system behavior and recommended the use of a combined approach where the system must first decide whether polling or interrupts should be used when there is a need. You have categorically rejected this proposal.

i) Explain the reasons behind your rejection. You need to explain why the mere use of interrupts would always result in better performance.

ii) While you have already rejected the proposal, you indicated that there is still some validity to it. Explain clearly under what conditions such approach may be valid (may actually result in better performance).

(**Important**: do NOT use the entire space below; you do NOT need it!)

Question # 3 (3 marks)

Assume a software has 3 major components: the first component performs an excessive amount of computations, the second one is expected to constantly perform read and write operations to/from some I/O devices, and the third component is a Graphical User Interface (GUI) that enable the user to easily use the software.

i) Assume that only one copy of the software is expected to run at a given time; will you implement that software as a single-thread (a process) or as a threaded system? Explain clearly the reasons behind your choice.

Answer: I would implement the system as a: ☐ Single-thread (process) ☐ Multi-threading

Explanation:

ii) Now, let us assume a change in the requirements, where multiple copies of the software are expected to run at the same time. Under these new conditions, would you change your implementation from #i above? Explain the reasons behind your answer.

Answer:: ☐ Yes; my implementation changes ☐ No; will not change implementation

Explanation:

Question # 4 (4 marks)

The following solution was given to solve the producer consumer problem, where the consumer should not be able to consume any items until at least two or more items are produced. Does this solution suffer any problems? Whether *yes* or *no*, explain your answer very clearly and in details.

Semaphore usedSlots=0;
Semaphore freeSlots=N;
Semaphore lock=2;

```
//Producer Code – May run many times
AddToBuffer( item ) {
        P( freeSlots );
        P( lock );
        V( lock );
                add_item;
        V( lock );
        V( usedSlots );
}
```

```
//Consumer Code – May run many times
RemoveFromBuffer( item ) {
        P( lock );
        P( lock );
        P( usedSlots );
        V( lock );
        P( lock );
                remove_item;
        V( freeSlots );
        V( lock );
}
```

Question # 5 (4 marks)

As a member of a system's design team, you investigated the use of *system calls* in the system and concluded that they are actually secure, with the exception of **6** particular calls that have the potential of producing serious security problems (improper privilege escalation to system-mode). In particular, an exploitation of those routines would allow an attacker to gain different controls (manipulations, use, changes, etc.) to some hardware components of the system. Since the fixing of these routines is estimated to take longer time that what can be accommodated by the clients, another team member recommended a temporary solution to the problem, where these 6 routines are actually permitted to be executed under user-mode instead of system-mode, hence eliminating the privilege escalation problem. Does this proposal have any validity from your point of view? If *yes*, indicate clearly the reasons behind your support to the proposal. If *no*, explain clearly why such proposal is incorrect.

Answer: ☐ Yes; there is validity to the proposal ☐ No; proposal has no validity

Explanation:

Question # 6 (4 marks)

Sometimes it is necessary to synchronize two or more processes in a group so that every process must finish its first phase of computation before any other process is allowed to start its second phase. This is called barrier synchronization; For example, for two processes P1 and P2, we can write:

Semaphores: s1 = 0, s2 = 0;

process P1	process P2
"phase I" V (s1) P (s2) "phase II"	"phase I" V (s2) P (s1) "phase II"

For this question, you are required to synchronize five processes P1, P2, P3, P4, and P5. All processes will run exactly once, with the exception of P2, which runs exactly 3 times. The new rules are: **1)** Process **P2** must run three times (that is, finishing both phase I and phase II, three times) before any of the other processes can access any of the phases, **2)** Process **P4** must finish phase I first, followed by process **P5**, which must finish Phase I before **P1** and **P3** can start that phase (there is no order for Phase I between **P1** and **P3**, but they cannot be both there at the same time), **3)** all phase I's must finish before any phase II starts (P2 is excluded from this rule as it has already finished!), and **4)** all phase II's must proceed in the order "**5143**" (that is **P5** must finish Phase II before **P1**, and **P1** must finish Phase II before **P4**, and finally **P4** must finish Phase II before **P3**). You are allowed to use any number of semaphores; however semaphores can **NOT** have any negative values. Initializing or setting any semaphore to a negative value will result in a total discard of the solution.

Answer:

Semaphores: _____

Process P1	Process P2	Process P3	Process P4	Process P5