Carleton UNIVERSITY

Sample Solution to MIDTERM
EXAMINATION
Winter 2018

DURATION: 1.5 HOURS **No. Of Students: 47**

Department Name & Course Number: Systems and Computer Engineering SYSC 4001B

Course Instructor: Thomas Kunz

AUTHORIZED MEMORANDA:

William Stallings, *Operating Systems: Internals and Design Principles*, 9th edition, Pearson 2018, ISBN-9780134670959 (as physical book, no ebook) or earlier versions of that same book.

Students MUST count the number of pages in this examination question paper before beginning to write, and report any discrepancy to a proctor. This question paper has 5 pages + cover page = 6 pages in all.

This examination question paper MAY NOT be taken from the examination room.

In addition to this question paper, students require: an examination booklet: NO
Scantron Sheet: NO

Name: _____

Student Number: _____

Question 1: _____ /10

Question 2: _____ /10

Question 3: _____ /10

Question 4: _____ /10

Total: _____/40

Question 1: Process Description and Control (10 marks)

1. Consider a computer with *N* processors in a multiprocessor configuration.
   a. How many processes can be in each of the `Ready`, `Running`, and `Blocked` states at one time?
   b. What is the minimum number of processes that can be in each of the `Ready`, `Running`, and `Blocked` states at one time?

Answer (5 marks):

a. There is no limit in principle to the number of processes that can be in the `Ready` and `Blocked` states. The OS may impose a limit based on resource constraints related to the amount of memory devoted to the process state for each process. At most, *N* processes can be in the `Running` state, one for each processor.
b. Zero is the minimum number of processes in each state. It is possible for all processes to be in either the `Running` state (up to N processes) or the `Ready` state, with no process blocked. It is possible for all processes to be blocked waiting for I/O operations, with zero processes in the `Ready` and `Running` states.

2. Consider a system in which the following states are defined for processes: `Execute` (running), `Active` (ready), `Blocked`, and `Suspend`. A process is blocked if it is waiting for permission to use a resource, and it is suspended if it is waiting for an operation to be completed on a resource it has already acquired. In many operating systems, these two states are lumped together as the blocked state, and the suspended state has the definition we have used in class/the textbook. Compare the relative merits of the two sets of definitions.

Answer (5 marks):

One textbook had the following description to motivate the two states as follows:

> Suppose a process has been executing for a while and needs an additional magnetic tape drive so that it can write out a temporary file. Before it can initiate a write to tape, it must be given permission to use one of the drives. When it makes its request, a tape drive may not be available, and if that is the case, the process will be placed in the blocked state. At some point, we assume the system will allocate the tape drive to the process; at that time the process will be moved back to the active state. When the process is placed into the execute state again it will request a write operation to its newly acquired tape drive. At this point, the process will be move to the suspend state, where it waits for the completion of the current write on the tape drive that it now owns.

So it is only a matter of convenience. The distinction made between two different reasons for waiting for a device could be useful to the operating system in organizing its work. However, it is no substitute for a knowledge of which processes are swapped out and which processes are swapped in. This latter distinction is a necessity and must be reflected in some fashion in the process state.

Question 2: Concurrency: Mutual Exclusion and Synchronization (10 marks)

1) A situation in which a runnable process is overlooked indefinitely by the scheduler, although it is able to proceed, is _____ .

A) mutual exclusion B) deadlock

C) starvation D) livelock

2) The requirement that when one process is in a critical section that accesses shared resources, no other process may be in a critical section that accesses any of those shared resources is _____ .

A) critical section B) livelock

C) mutual exclusion D) atomic operation

3) A means for two processes to exchange information is with the use of _____ .

A) spinlocks B) event flags

C) condition variables **D) messages**

4) A semaphore that does not specify the order in which processes are removed from the queue is a _____ semaphore.

A) weak B) general

C) strong D) binary

5) A _____ occurs when multiple processes or threads read and write data items so that the final result depends on the order of execution of instructions in the multiple processes.

A) atomic operation **B) race condition**

C) livelock D) deadlock

6) The term _____ refers to a technique in which a process can do nothing until it gets permission to enter its critical section but continues to execute an instruction or set of instructions that tests the appropriate variable to gain entrance.

A) spin waiting B) general semaphore

C) critical resource D) message passing

7) A _____ is a data type that is used to block a process or thread until a particular condition is true.

A) deadlock B) general semaphore

C) condition variable D) mutex

8) A semaphore whose definition includes the policy that the process that has been blocked the longest is released from the queue first is called a _____ semaphore.

A) general **B) strong**

C) weak D) counting

9) The _____ is a programming language construct that provides equivalent functionality to that of semaphores and is easier to control.

A) atomic operation B) coroutine

C) critical section **D) monitor**

10) Probably the most useful combination, _____ allows a process to send one or more messages to a variety of destinations as quickly as possible.

A) blocking send, blocking receive

B) nonblocking send, blocking receive

C) nonblocking send, nonblocking receive

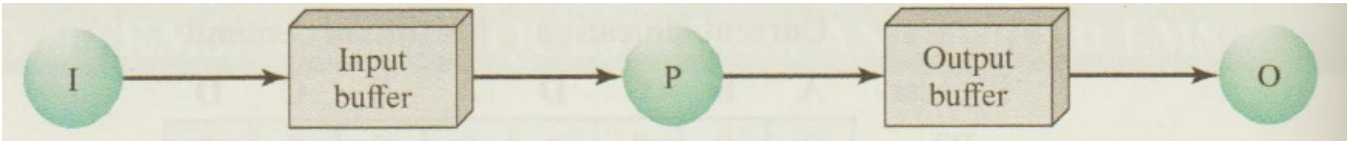
D) blocking send, nonblocking receive

Question 3. Deadlocks (10 marks)

1. A spooling system (see picture below) consists of an input process I, a user process P, and an output process O, connected by two buffers. The processes exchange data in blocks of equal size. These blocks are buffered on a disk using a floating boundary between the input and the output buffers, depending on the speed of the processes. The communication primitives used ensure that the following resource constraint is satisfied:

   `i + o <= max`

   `max` = maximum number of blocks on disk
   `i` = number of input blocks on disk
   `o` = number of output blocks on disk



   The following is known about the processes:

   - As long as the environment supplies data, process I will eventually input it to the disk (provided disk space becomes available).
   - As long as input is available on the disk, process P will eventually consume it and output a finite amount of data on the disk for each block input (provided disk space becomes available).
   - As long as output is available on the disk, process O will eventually consume it.

   Show that this system can become deadlocked.

   Answer (5 marks):
   A deadlock occurs when process I has filled the disk with input (i = max) and process I is waiting to transfer more input to the disk, while process P is waiting to transfer more output to the disk and process O is waiting to transfer more output from the disk.

2. Suggest an additional resource constraint that will prevent the deadlock in the first part, but still permit the boundary between input and output buffers to vary in accordance with the present needs of the processes.

   Answer (5 marks):
   Reserve a minimum number of blocks (called `reso`) permanently for output buffering, but permit the number of output blocks to exceed this limit when disk space is available. The resource constraints now become:

   `i + o ≤ max`
   `i ≤ max – reso`
   where
   `0 < reso < max`

   If process P is waiting to deliver output to the disk, process O will eventually consume all previous output and make at least `reso` pages available for further output, thus enabling P to continue. So P cannot be delayed indefinitely by O. Process I can be delayed if the disk is full of I/O; but sooner or later, all previous input will be consumed by P and the corresponding output will be consumed by O, thus enabling I to continue.

Question 4. Memory Management (10 marks)

1. Consider the following segment table:

Segment	Base	Length
0	1219	600
1	3300	14
2	90	100
3	2327	580
4	1952	96

What are the physical addresses for the following logical addresses?
a. 0, 430
b. 1, 15
c. 2, 50
d. 3, 400
e. 4, 112

Answer (5 marks):
a. 1219 + 430 = 1649
b. illegal
c. 90 + 50 = 140
d. 2327 + 400 = 2727
e. illegal

2. A system uses (contiguous) dynamic partition memory management, with 110K of memory for user processes. The current memory allocation table (all numbers are in units of K) is shown below:
3.

Job	Base Address	Length
A	20	10
B	46	18
C	90	20

A new job, D, arrives needing 15K of memory. Show the memory allocation table entry for job D for each of the following memory allocation strategies: first fit, best fit, worst fit.

Answer (3 marks):

Strategy	Base Address	Length
First fit	0	15
Best fit	30	15
Worst fit	64	15

4. Consider a logical address space of eight pages of 1024 bytes each, mapped onto a physical memory of 32 frames.
a. How many bits are there in the logical address?
b. How many bits are there in the physical address?

Answer (2 marks):
a. $8 \times 1024 = 8192 = 2^{13}$
13 bits needed for the logical address
b. $32 \times 1024 = 32{,}768 = 2^{15}$
15 bits needed for the physical address