Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

Midterm Exam – CSI3131

Exam Duration: 60 minutes 28 June 2018

Professor: Ayman El Sawah

First Name: _____

Last Name: _____

Student Number: _____

Signature: _____

Closed book exam.

Please answer in the space provided (in this questionnaire).
No calculators are allowed.

Once you complete the exam, please exit quietly leaving
your booklet at the booth.

(Multiple Choice Questions – Use the Scantron Sheet)

Each multiple-choice question has only <u>one correct answer.</u>

Question 1 (1 point) – What is operating system?

a) OS is a collection of programs that manages hardware resources
b) OS is a system service provider to the application programs
c) OS is a link to interface the hardware and application programs
d) OS is all of the mentioned
e) OS is none of the mentioned

Question 2 (1 point) – To access the services of operating system, the interface is provided by:

a) System calls
b) API
c) Library
d) Assembly instructions
e) None of the above

Question 3 (1 point) – Which one of the following is **<u>not</u>** true?

a) kernel is the program that constitutes the central core of the operating system
b) kernel is the first part of operating system to load into memory during booting
c) kernel is made of various modules which cannot be loaded in running operating system
d) kernel remains in the memory during the entire computer session

Question 4 (1 point) – In operating system, each process has its own:

a) address space and global variables
b) open files
c) pending alarms, signals and signal handlers
d) all of the above mentioned
e) None of the above mentioned

Question 5 (1 point) – A process can be terminated due to:

a) normal exit
b) fatal error
c) killed by another process
d) all of the mentioned
e) None of the above

Question 6 (1 point) – Which of the following is **not** stored in stack:

a) Function parameters
b) Local variables
c) Return addresses
d) PID of child process
e) all of the mentioned

Question 7 (1 point) – Which system call returns the process identifier of a terminated child?

a) wait
b) exit
c) fork
d) get
e) exec

Question 8 (1 point) – The address of the next instruction to be executed by the current process is provided by the:

a) CPU registers
b) Program counter
c) Process stack
d) Pipe
e) None of the above

Question 9 (1 point) – The number of processes completed per unit time is known as _____

a) Output
b) Throughput
c) Efficiency
d) Capacity
e) None of the above

Question 10 (1 point) – The Process Control Block is:

a) Process type variable
b) Data Structure
c) A secondary storage section
d) A Block in memory
e) None of the above

Question 11 (1 point) – The degree of multi-programming is:

a) the number of processes executed per unit time
b) the number of processes in the ready queue
c) the number of processes in the I/O queue
d) the number of processes in memory
e) None of the above

Question 12 (1 point) – The parent process completes execution, but the child keeps executing, then the child process is known as:

a) Orphan
b) Zombie
c) Body
d) Dead
e) None of the above

Question 13 (1 point) – When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called:

a) dynamic condition
b) race condition
c) essential condition
d) critical condition
e) None of the above

Question 14 (1 point) – In Java, you can create a thread by:

a) Extending the Object class or using the intrinsic lock
b) Creating a class called Thread or creating an interface called Runnable
c) Implementing the Runnable interface or extending the Thread class
d) Creating immutable objects out of the Task class
e) Extending the Runnable class or implementing the Thread interface

Question 15 (1 point) – A monitor is a module that encapsulates:

a) shared data structures
b) procedures that operate on shared data structure
c) synchronization between concurrent procedure invocation
d) all of the mentioned
e) None of the above

Question 16 (1 point) – If all processes I/O bound, the ready queue will almost always be _____ and the Short term Scheduler will have a _____ to do.

a) full, little
b) full, lot
c) empty, little
d) empty, lot

Question 17 (1 point) – What is a medium-term scheduler?

a) It selects which process has to be brought into the ready queue
b) It selects which process has to be executed next and allocates CPU
c) It selects which process to remove from memory by swapping
d) None of the mentioned

Question 18 (1 point) – In a time-sharing operating system, when the time slot given to a process is completed, the process goes from the running state to the:

a) Blocked state
b) Ready state
c) Suspended state
d) Terminated state
e) New state

Question 19 (1 point) – Suppose that a process is in "Blocked" state waiting for some I/O service. When the service is completed, it goes to the:

a) Running state
b) Ready state
c) Suspended state
d) Terminated state
e) none of the mentioned

Question 20 (1 point) – What is a short-term scheduler?

a) It selects which process has to be brought into the ready queue
b) It selects which process has to be executed next and allocates CPU
c) It selects which process to remove from memory by swapping
d) All of the mentioned
e) none of the mentioned

Question 21 (1 point) – The interval from the time of submission of a process to the time of completion is termed as:

a) waiting time
b) turnaround time
c) response time
d) throughput
e) real time

Question 22 (1 point) – In priority scheduling algorithm, when a process arrives at the ready queue, its priority is compared with the priority of:

a) all processes
b) blocked processes
c) parent process
d) init process
e) currently running process

Question 23 (1 point) – Time quantum is defined in:

a) shortest job scheduling algorithm
b) round robin scheduling algorithm
c) priority scheduling algorithm
d) multilevel queue scheduling algorithm
e) none of the mentioned

Question 24 (1 point) – In multilevel feedback scheduling algorithm:

a) a process can move to a different classified ready queue
b) classification of ready queue is permanent
c) processes are not classified into groups
d) all of the mentioned
e) none of the mentioned

Question 25 (1 point) – Which of the following does not interrupt a running process?

a) A device
b) Timer
c) Scheduler process
d) Power failure
e) none of the mentioned

Question 26 (1 point) – Which one of the following cannot be scheduled by the kernel?

a) kernel level thread
b) user level thread
c) process
d) interrupt service routine
e) none of the mentioned

Question 27 (1 point) – An I/O bound program will typically have:

a) a few very short CPU bursts
b) many very short I/O bursts
c) many very short CPU bursts
d) a few very short I/O bursts
e) a few very long CPU bursts

Question 28 (1 point) – Bounded waiting implies that there exists a bound on the number of times a process is allowed to enter its critical section:

a) before a process has made a request to enter its critical section
b) when another process is in its critical section
c) after a process has made a request to enter its critical section and before the request is granted
d) all of the mentioned
e) none of the mentioned

Question 29 (1 point) – A minimum of _____ variable(s) is/are required to be shared between processes to properly solve the critical section problem.

a) zero
b) one
c) two
d) three
e) four

Question 30 (1 point) – To ensure data integrity in the readers – writers problem, _____ are given exclusive access to the shared object.

a) readers
b) writers
c) readers and writers
d) not readers nor writers
e) none of the mentioned

Question 31 (1 point) – All processes share a semaphore variable **mutex**, initialized to 1. Each process must execute **wait(mutex)** before entering the critical section and **signal(mutex)** afterward. Suppose a process executes in the following manner:

```
signal(mutex);
.....
critical section
.....
wait(mutex);
```

In this situation:

a) a deadlock will occur
b) processes will starve to enter critical section
c) several processes maybe executing in their critical section
d) all of the mentioned
e) none of the mentioned

Question 32 (1 point) – All processes share a semaphore variable **mutex**, initialized to 1. Each process must execute **wait(mutex)** before entering the critical section and **signal(mutex)** afterward. Suppose a process executes in the following manner:

```
wait(mutex);
.....
critical section
.....
wait(mutex);
```

In this situation:

a) a deadlock will occur
b) processes will starve to enter critical section
c) several processes maybe executing in their critical section
d) all of the mentioned
e) none of the mentioned

Question 33 (1 point) – Consider the methods used by processes P1 and P2 for accessing their critical sections whenever needed, as given below. The **initial values of shared Boolean variables S1 and S2 are randomly assigned**.

```
//Method used by P1:
while(S1==S2);
Critical section
S1 = S2;

//Method used by P2:
while(S1!=S2);
Critical section
S2 = not(S1);
```

In this situation, the following statements describes properties achieved:

a) Mutual exclusion but not progress
b) Progress but not mutual exclusion
c) Neither mutual exclusion nor progress
d) Both mutual exclusion and progress
e) none of the mentioned

Question 34 (1 point) – An un-interruptible unit is known as:

a) single
b) atomic
c) static
d) mutex
e) none of the mentioned

Question 35 (1 point) – Spinlocks are intended to provide _____ only.

a) Mutual Exclusion
b) Bounded Waiting
c) Aging
d) Progress
e) All of the mentioned

Question 36 (1 point) – The **TestAndSet** instruction is executed:

a) after a particular process
b) periodically
c) atomically
d) all of the mentioned
e) none of the mentioned

Question 37 (1 point) – A monitor is a type of:

a) semaphore
b) low level synchronization construct
c) high level synchronization construct
d) not a synchronization construct
e) mutex

Question 38 (1 point) – A monitor is characterized by:

a) a set of programmer defined operators
b) an identifier
c) the number of variables in it
d) all of the mentioned
e) none of the mentioned

Question 39 (3 point) – The following program consists of 3 concurrent processes and 3 binary semaphores. The **semaphores are initialized as S0 = 1, S1 = 0, S2 = 0**.

```
//Process P0:
while(true)
{
wait(S0);
print '0';
release(S1);
release(S2);
}

//Process P1:
wait(S1);
release(S0);

//Process P2:
wait(S2);
release(S0);
```

In this situation, P0 print '0':

a) At least twice
b) Exactly twice
c) Exactly three times
d) At least three times
e) none of the mentioned

Question 40 (3 point) – Each process Pi, I = 0,1,2,3,……,9 is coded as follows:

```
While (true) {
 wait(mutex)
 {Critical Section}
 signal(mutex)
}
```

The code for P10 is coded as follows:

```
While (true) {
 signal(mutex)
 {Critical Section}
 wait(mutex)
}
```

What is the largest number of processes that can be inside the critical section at any moment (the mutex being initialized to 1)?

a) 1
b) 2
c) 3
d) 4
e) none of the mentioned

Question 41 (1 point) – Two processes, P1 and P2, need to access a critical section of code. Consider the following synchronization construct used by the processes:

```
//Process P1 :
while(true)
{
  w1 = true;
  while(w2 == true);
  Critical section
  w1 = false;
}
Remainder Section

//Process P2 :
while(true)
{
  w2 = true;
  while(w1 == true);
  Critical section
  w2 = false;
}
Remainder Section
```

Here, w1 and w2 are shared variables, which are initialized to false. Which one of the following statements is TRUE about the above construct?

a) It does not ensure mutual exclusion
b) It does not ensure bounded waiting
c) It requires that processes enter the critical section in strict alternation
d) It does not prevent deadlocks, but ensures mutual exclusion
e) none of the mentioned

Question 42 (1 point) – Deadlock prevention is a set of methods:

a) to ensure that at least one of the necessary conditions cannot hold
b) to ensure that all of the necessary conditions do not hold
c) to decide if the requested resources for a process have to be given or not
d) to recover from a deadlock
e) none of the mentioned

Question 43 (1 point) – To ensure that the hold and wait condition never occurs in the system, it must be ensured that:

a) whenever a resource is requested by a process, it is not holding any other resources
b) each process must request and be allocated all its resources before it begins its execution
c) a process can request resources only when one is available
d) any of the mentioned
e) all of the mentioned

Question 44 (3 point) – The following pair of processes share a common variable X:

```
//Process P1 :
{
  int Y;
  A1: Y = X*2;
  A2: X = Y;
}

//Process P2 :
{
  int Z;
  B1: Z = X+1;
  B2: X = Z;
}
```

X is set to 5 before either process begins execution. As usual, statements within a process are executed sequentially, but statements in process A may interleave with statements in process B. How many different values of X are possible after both processes finish executing?

a) 1
b) 2
c) 3
d) 4
e) 8

Question 45 (1 points) – Which of the following statements is **true** about Shortest Job First (SJF).

ID	Statement
S1	It causes minimum average waiting time
S2	It can cause starvation

a) Only S1
b) Only S2
c) Both S1 and S2
d) Neither S1 nor S2

Question 46 (1 point) – The disadvantage of a process being allocated all its resources before beginning its execution is:

a) Low CPU utilization
b) Low resource utilization
c) Very high resource utilization
d) Very high CPU utilization
e) None of the mentioned

Question 47 (3 points) – Consider the following table of arrival time and burst time for three processes P0, P1 and P2.

Process	Arrival time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?

a) 4.0 ms
b) 4.33 ms
c) 5.0 ms
d) 5.33 ms
e) 6.33 ms

Question 48 (3 points) – The **enter_CS()** and **leave_CS()** functions to implement critical section of a process are realized using test-and-set instruction as follows:

```
void enter_CS(X){ while (test-and-set(&X)==false) ; }
void leave_CS(X){ X = 0; }

bool test-and-set(int *var){
  if (*var==0) {
    *var=1;
    return true;
  }
  else {
    return false;
  }
}
```

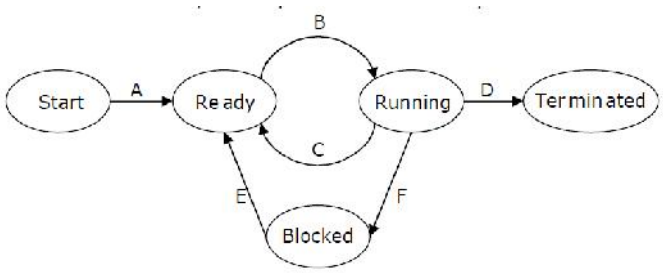
In the above solution, X is a memory location associated with the CS and is initialized to 0. Now consider the following statements:

ID	Statement
S1	The above solution to CS problem is deadlock-free
S2	The solution is starvation free
S3	The processes enter CS in FIFO order
S4	More than one process can enter CS at the same time

Which of the above statements is TRUE?

a) S1 only
b) S1 and S2
c) S1 and S3
d) S2 and S3
e) S4 only

Question 49 (3 points) – In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state:



Now consider the following statements:

ID	Statement
S1	If a process makes a transition D, it would result in another process making transition A immediately
S2	A process P2 in blocked state can make transition E while another process P1 is in running state
S3	The OS uses pre-emptive scheduling
S4	The OS uses non-pre-emptive scheduling

Which of the above statements are TRUE?

a) S1 only
b) S1 and S2
c) S1 and S3
d) S2 and S3
e) S2 and S4

Question 50 (1 point) – Given a priori information about the _____ number of resources of each type that maybe requested for each process, it is possible to construct an algorithm that ensures that the system will never enter a deadlock state.

a) minimum
b) average
c) maximum
d) approximate
e) None of the mentioned

Question 51 (1 point) – A state is safe, if:

a) the system does not crash due to deadlock occurrence
b) the system can allocate resources to each process in some order and still avoid a deadlock
c) the state keeps the system protected and safe
d) all of the mentioned
e) None of the mentioned

Question 52 (1 point) – All unsafe states are:

a) deadlocks
b) not deadlocks
c) fatal
d) all of the mentioned
e) None of the mentioned

Question 53 (3 point) – A system has 12 magnetic tape drives and 3 processes: P0, P1, and P2. Process P0 requires 10 tape drives, P1 requires 4 and P2 requires 9 tape drives.

Process	Maximum Need	Currently Hold
P0	10	5
P1	4	2
P2	9	2

Which of the following sequence is a safe sequence?

a) P0, P1, P2
b) P1, P2, P0
c) P2, P0, P1
d) P1, P0, P2
e) None of the mentioned

Question 54 (3 point) – A system with 5 processes P0 through P4 and three resource types A, B, C has A with 10 instances, B with 5 instances, and C with 7 instances. At time t0, the following snapshot has been taken:

Process	Allocated				Max Required		
	A	B	C		A	B	C
P0	0	1	0		7	5	3
P1	2	0	0		3	2	2
P2	3	0	2		9	0	2
P3	2	1	1		2	2	2
P4	0	0	2		4	3	3

	A	B	C
Available	3	3	2

The sequence < P1, P3, P4, P2, P0 > leads the system to:

a) an unsafe state
b) a safe state
c) a protected state
d) a deadlock
e) None of the mentioned