Department of Computer Science
University of Saskatchewan
CMPT 332
Midterm Exam

October 31, 2011
Number of Pages: 5
Time: 50 minutes
Total: 50 marks.

Name:_____
Signature:_____
Student Number: _____
NSID: _____

CAUTION - Candidates suspected of any of the following, or similar, dishonest practices shall be dismissed from the examination and shall be liable to disciplinary action.

1. Having at the place of writing any communication devices, any books, papers or memoranda, calculators, audio or visual cassette players, or other memory aid devices.
2. Speaking or communicating with other candidates.
3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	TOTAL
5	5	4	12	10	6	4	4	50

1. (5 marks) Assume you have 640 KB of main memory. Processes A, B, C, D, E and F are presented to the medium-term scheduler for execution. How is memory allocated for the following sequence of operations for loading these 6 processes into memory using the FirstFit algorithm? Show your answer by providing a diagram of RAM after all the allocation and deallocation has been completed.

Process	Size (in KB)
A	180
B	100
C	144
D	98
E	240
F	70

Operations to perform:

- Allocate D,
- Allocate A,
- Allocate C,
- Free A,
- Allocate E,
- Allocate B,
- Free D,
- Allocate F.

IF the operations cannot be completed in the given order, indicate which operations would have to be aborted. Do not complete an aborted operation at a later time.

2. (5 marks). In class, we discussed 3 (three) types of information that is typically stored in a process control block. What are those 3 types of information? For each type, give one example PCB field and indicate **when** it is modified during process execution.

3. (5 marks) Consider the following proposed software solution to the critical section problem (process i is shown - process j is identical except i's and j's are interchanged):

   FLAGi and FLAGj are shared and both initially false

```
while(1) {
      FLAGi = TRUE;
      if (FLAGj == TRUE) {
           FLAGi = FALSE;
      }
      else  {
            DO CRITICAL SECTION;
            FLAGi = FALSE;
      }


      DO NON CRITICAL SECTION;
}
```

   Is this a correct solution to the critical section problem? If not, which characteristics of solutions to the Critical Section problem are met, and which are not? First state yes or no, then support your answer with a short argument.

4. (12 marks) Process Scheduling. Consider the following job mix, including relative arrival times.

Job	Arrival Time	CPU Burst Time
A	0	10
B	3	8
C	4	8
D	5	4
E	6	2

Show the order of execution of these jobs using Round-Robin with a quantum of 3 and Shortest Remaining Time First. Use a time-line like in the interactive examples and/or class showing which jobs execute during which time slots. For RR, jobs are assumed to arrive just **before** the time slot indicated and are placed at the **back** of the ready queue.

```
RR

-------------------------------------------------------------------------
|        |
-------------------------------------------------------------------------
 0      3  etc...

TAT:                                          EFF:
```

```
SRTF

-------------------------------------------------------------------------
|        |
-------------------------------------------------------------------------
 0

TAT:                                          EFF:
```

Indicate the avg. turnaround time (TAT) and the specific CPU efficiency for each algorithm on this set of tasks. For the efficiency, assume that each context switch takes .2 of a time unit. For the timeline, assume that each context switch takes 0 time. No calculator is necessary; leave the efficiency as a fraction.

5. (10 marks). Semaphores, Monitors, and Rendezvous IPC can all solve the same classes of problems, in that you can implement one facility with one of the others. Use the Monitor primitives MonEnter, MonLeave, MonWait, and MonSignal as defined in class, to provide Send/Receive/Reply IPC semantics. Provide detailed C-like pseudocode. Assume that each process has a field for incoming message and message length, and that this memory can be accessed by any process. Define the condition variables you require. Consider the very likely case that multiple senders may send to the same receiver simultaneously. I've got you started.

```
void *Send(
{
```

```
void *Receive(
  {
```

```
int Reply(
  {
```

6. (6 marks). You've been asked to design and implement a communication protocol. The protocol will be used to support communication for a server your company uses. You ask your boss

   "Does the server in this communication protocol provide an idempotent or a non-idempotent service?"

   Your boss gives you a look like you're nuts and asks you what you are talking about. Please define the meaning of the word idempotent in this context, and explain why it is important for you to know this given the task at hand.

7. (4 marks) What two advantages do multiple cores have when compared with multiple processors? What one major disadvantage do they have? List 1 (one) major motivation from the operating system view for having multiple execution cores on the same chip.

8. (4 marks) Briefly outline one approach for each of deadlock detection and deadlock avoidance. What is one advantage of one of your suggestions over the other?

——————- THE END——————