University of Toronto Mississauga

February 2009 Midterm Test
Course: CSC369H5S
Instructor: Andrew Petersen
Duration: 50 minutes
Aids allowed: none

Last Name:	______________________
Given Name:	______________________
Student Number:	______________________

This midterm test consists of 5 questions on 10 pages (including this one and two scrap pages). When you receive the signal to start, please make sure that your copy of the test is complete.

If you need more space for one of your solutions, use the last pages of the test and indicate clearly the part of your work that should be marked.

In written answer, be as specific as possible and explain your reasoning. Clear, concise answers will be given higher marks than vague, wordy answers. Marks will be deducted for any incorrect statements in an answer. Please make your handwriting legible!

Marking Guide

# 1:	______/ 6
# 2:	______/12
# 3:	______/ 8
# 4:	______/ 5
# 5:	______/14
TOTAL:	______/45

continued on page 2

6 marks **Question 1.** Very Short Answer

Each of the questions below should be answered with a couple of words or one sentence.

1 mark **Part (a)** What is the fundamental difference between a condition variable and a semaphore?

1 mark **Part (b)** Name one advantage and one disadvantage of using processes rather than threads.

1 mark **Part (c)** Consider the following code snippet used to implement critical sections in a system with two threads with `id` 0 and 1 respectively. It uses a shared variable `int flag[2]` that is used to indicate whether a thread intends to enter the critical section.

```
flag[id] = true;
while (flag[1-id]);
/* critical section, access protected resource */
flag[id] = false
```

Does this code correctly implement critical sections? Explain your answer.

CONTINUED ON PAGE 3

1 mark **Part (d)** Explain why context switch time is especially important when implementing the round robin scheduling heuristic.

1 mark **Part (e)** What is the difference between an error code and a return value? How are error codes returned to the user in OS/161?

1 mark **Part (f)** Provide one example of an OS *mechanism* and provide an example of a *policy* that uses it.

continued on page 4

12 marks **Question 2.** Short Answer

Each of the questions below should be answered with a paragraph or a diagram.

3 marks **Part (a)** Draw a diagram illustrating the address space for a process with two threads. Label each section in your diagram.

3 marks **Part (b)** Describe the events that occur during a system call. You may refer to functions in OS/161 if you wish.

CONTINUED ON PAGE 5

3 marks **Part (c)** In the hockeyrink problem, how does your implementation handle the situation when a skater leaves the rink and no other skaters are currently skating? Be sure to mention any synchronization that is involved.

3 marks **Part (d)** Discuss when and why child threads must be detached when their parent exits.

CONTINUED ON PAGE 6

8 marks **Question 3.** Deadlock and Starvation

2 marks **Part (a)** Briefly describe how deadlock and starvation are related and why they are not synonyms.

3 marks **Part (b)** Compare deadlock prevention and deadlock avoidance. Name a weakness of both strategies.

1 mark **Part (c)** Why is it reasonable to ignore deadlock created by resource contention in most modern operating systems?

The following table describes a system that has four active processes that are sharing a non-unique resource; use it for the following two questions. The "maximum" column refers to the maximum amount of the resource that a process may request, and "current" refers to the amount of that resource currently held by a process.

Process	**Current**	**Maximum**
A	2	4
B	4	8
C	0	1
D	1	5

1 mark **Part (d)** If 12 units of the resource are available, construct an example in which the system is deadlocked.

1 mark **Part (e)** How many units of the resource are required to guarantee deadlock freedom?

CONTINUED ON PAGE 7

5 marks **Question 4.** Address Translation

The following questions test your understanding of paging and address translation. The memory system in question has **22 bit virtual addresses** with **65536 (2^{16}) byte pages**.

Here is a fragment of the page table:

Page Number	**Frame Number**
0	0xA
1	0x2C
2	0x7
3	0xF0
...	...

1 mark **Part (a)** What is the maximum number of pages a process may have?

1 mark **Part (b)** What physical address does the virtual address 0x03BEEF correspond to?

1 mark **Part (c)** What virtual address does the physical address 0x2CF070 correspond to?

1 mark **Part (d)** If physical addresses are actually 32 bits, what is the maximum number of frames the OS can support?

1 mark **Part (e)** State one disadvantage of using paging.

CONTINUED ON PAGE 8

14 marks **Question 5.** CPU Scheduling

12 marks **Part (a)** An OS uses a multi-level feedback scheduler with 2 levels. New and returning processes start at level 0, which uses round robin scheduling with a quantum of 2. A process that uses its entire quantum at level 0 gets moved to level 1, which uses first-come-first-served scheduling. The scheduler always chooses a process from the lowest-numbered non-empty level. Admission of a new process or a process returning from IO does not force a scheduling decision; those processes are placed on the level 0 queue to wait until a scheduling decision is to be made.

Initially, there is one process, $P0$. New processes $P1$ and $P6$ are created at times 1 and 6 respectively. Those are the only three processes you need to schedule. Each process has a CPU burst of 5 time units, an IO burst of 3 time units, and a CPU burst of 1 time unit. During an IO burst, a process is blocked (on a wait queue); it does not require the processor.

Assume context switches take no time. Fill in the timeline below for each process. Note when a process is *New*, when it has the *CPU*, when it loses the CPU due to a *Preempt*, what queue it is on (*L0, L1, IO*), and when it *Exits*. The first 5 time units have been filled in for you.

Time	0	1	2	3	4	5	6
P0	CPU	CPU	Preempt, L1	L1	CPU		
P1		New, L0	CPU	CPU	Preempt, L1		
P6							

Time	7	8	9	10	11	12	13
P0							
P1							
P6							

Time	14	15	16	17	18	19	20
P0							
P1							
P6							

1 mark **Part (b)** For **your solution above**, what is the total wait time (time spent ready but not executing) for all processes?

1 mark **Part (c)** For **your solution above**, what is the average response time (time spent waiting for the processes's first time slice)?

Total Marks = 45

CONTINUED ON PAGE 9

[Use the space below for rough work. This page will ***not*** *be marked, unless you clearly indicate the part of your work that you want us to mark.]*

continued on page 10

[Use the space below for rough work. This page will ***not*** *be marked, unless you clearly indicate the part of your work that you want us to mark.]*

End of Examination