University of Waterloo
Midterm Examination

Spring, 2008

Student Name:	_____
Student ID Number:	_____
Section:	_____

Course Abbreviation and Number	CS350
Course Title	Operating Systems
Section	01 (13:30)
Instructor	J. Selby

Date of Exam	June 24, 2008
Time Period	19:00-21:00
Duration of Exam	120 minutes
Number of Exam Pages (including this cover sheet)	11 pages
Exam Type	Closed Book
Additional Materials Allowed	None

Question 1: (16 marks)	Question 2: (15 marks)	Question 3: (11 marks)
Question 4: (8 marks)	Question 5: (14 marks)	Question 6: (6 marks)
Total: (70 marks)		

1. (16 total marks)

a. (3 marks)
What are the elements of a thread's context?

- *GPR + PC + SP*
- *stack*
- *one of (privalege, priority, etc.)*

b. (4 marks)
Parallelization of a task can be accomplished by decomposing the task into multiple threads or multiple processes. Give two benefits of each method.

- *Threads:*
  - *cheap and easy communication since they share Addr. space*
  - *thread creation and handling is lightweight*
- *Procs:*
  - *Procs are isolated from each other → fault tolerance*
  - *Different addr space makes programming easier → less race conditions.*

c. (3 marks)
What are the three different mechanisms we discussed that transfers control from a user process to the kernel?

- *Interrputs, exceptions and system calls*

d. (2 marks)

What are the key differences between *local* and *global* page replacement policies?

- *A local policy will only examine the pages of the process that caused the page fault while a global policy examines all pages currently in memory.*

e. (4 marks)

Define *temporal* and *spatial* locality and provide an example of each.

- *Temoporal - when an address (page) is referenced it is likely to be referenced again relatively soon in the future.*
- *Spatial - when an address (page) is referenced the addresses around it are likely to be requested in the future.*

2. (15 total marks)

a. (2 marks)

There are two kinds of semaphores - binary and counting. If binary semaphores are sufficient to ensure mutual exclusion describe the usefulness of counting semaphores.

- *Counting semaphores can be used to* **allocate** *and* **guard** *access to a limited number of resources.*

b. (2 marks)

Why are spinlocks (busy waiting) the preferred approach to guarding critical sections inside of the operating system kernel when executing on a multiprocessor?

- *Assumming that the CS is small, it is less expensive to have one thread executing on a processor inside of the CS while the others spin a short while. Otherwise, you must disable interrupts on ALL other processors which is very expensive.*

c. (6 marks)

Discuss three ways in which deadlock can be prevented.

List and describe the following:

- *No hold and wait,*
- *Preemption,*
- *Resource ordering*

d. (5 marks)

Some architectures provide a *swap* instruction that can be used to atomically swap the values of two variables. Suppose that inside of OS/161 we are going to simulate this functionality using the following `Swap` function:

```
/* Atomically swap the values of lock and key. */
void Swap(int* lock, int* key)
{
  int spl;
  int temp;

  spl = splhigh();
  temp = *lock;
  *lock = *key;
  *key = temp;
  splx(spl);
}
```

Using C or pseudo-code sketch a general solution to enforcing mutual exclusion using the `Swap()` synchronization primitive.

```
int lock, key;

lock = 0;
key = 1;


while (key == 1)
   Swap(&lock, &key)

|----------------------|
|   CRITICAL SECTION   |
|----------------------|

Swap(&lock, &key)
// instead of 2nd Swap could set
// lock = 0; key = 1;

|----------------------|
|   REMAINDER SECTION  |
|----------------------|
```

3. (11 total marks)

Suppose you are given the following *demand* (D_i), *allocated* (A_i) and *unallocated* (U) resource vectors:

$D_1 = (0, 0, 0, 1, 0)$
$D_2 = (0, 0, 0, 0, 1)$
$D_3 = (1, 1, 0, 0, 0)$
$D_4 = (0, 1, 0, 0, 0)$

$A_1 = (1, 0, 0, 0, 0)$
$A_2 = (0, 1, 0, 0, 0)$
$A_3 = (0, 0, 0, 1, 0)$
$A_4 = (0, 0, 1, 0, 1)$

$U = (0, 1, 0, 0, 0)$

a. (5 marks)

Draw the resource allocation graph defined by the given vectors.

b. (6 marks)

Using the deadlock detection algorithm discussed in class identify if the processes are in a state of deadlock. Show your work (i.e. For each step identify which (if any) process's resource request can be granted and the resulting update to the unallocated resource pool.). If the processes are deadlocked, report the processes and resources that are the cause of the deadlock.

4. (8 marks)

Beginning at the point when a user process executes a `syscall` (system call) instruction provide a detailed list of the sequence of events that occur as control is transfered to the kernel, the system call is handled and then control is transfered back to the user process.

-
-
-
-
-
-
-
-

5. (14 total marks)

Consider the contents of the following TLB and Page Table (snippet):

TLB

Page Number	Frame Number	Valid Bit
0x4019d	0x21f	1
0x4212f	0x026	0
0x0811d	0x11d	0
0x40015	0xf3e	1
0x0842f	0x20e	0
0x4002a	0x72d	0
0x40016	0x154	0
0x404ec	0x84b	1

Page Table (Snippet)

Page Number	Frame Number	Valid Bit
...	...	...
0x40013	0x542	1
0x40014	0xb6c	1
0x40015	0xf3e	1
0x40016	0x532	1
0x40017	0x68b	0
0x40018	0xf58	1
0x40019	0xc14	1
0x40020	0x49c	0
...	...	...

Given a page size of 4K (2^{12}), translate each of the following virtual addresses into physical addresses. For each, report whether it is a TLB hit or miss, a page table hit or miss and the corresponding physical address or identify if the translation results in a page fault.

a. (2 marks)
0x40015b0c

- *TLB Hit → 0xf3eb0c*

b. (3 marks)
0x40016c70

TLB Miss (invalid), PT Hit → 0x532c70

c. (3 marks)
0x4001775a

TLB Miss, PT Miss (invalid) → Page Fault

d. (3 marks)
0x40019b40

TLB Miss (invalid), PT Hit → 0xc14b40

e. (3 marks)

What is an inverted page table? Under what circumstances are they typically used?

- *A table indexed by frames which record the virtual page mapped to the frame and the process ID from which the page belongs to.*
- *When dealing with very large virtual address spaces single- and multi-level page tables become very large and slow memory accesses hence it may help to invert the table since VMem >> PMem.*

6. (6 total marks)

Although processes usually do not share (virtual) memory, in a segmented or paged (or both) virtual memory, sharing parts of the virtual address space is possible.

a. (2 marks)

The shared memory must consist of a set of segments or pages, not an arbitrary section of memory. Why is that the case?

b. (2 marks)

If shared memory is implemented in a paged (not segmented) environment, two or more page-table entries will point to the same page. Explain the complications this causes when performing page-in and page-out operations.

c. (2 marks)

If shared memory is implemented in a segmented environment (with or without paging), explain why the problems discussed in part (b) will be largely or completely eliminated.