Faculty of Arts & Sciences
Department of Computer Science
CMPS 272—Operating Systems
Fall 2005–06
Monday, January 23, 2006

Final Exam
Version A

Name:_____ **Student Id:**_____

Signature:_____ **Section:**

Lect I	**11–12**
Lect II	**1–2**

Duration: 120 minutes

Instructions

- The exam is made of 12 pages. Make sure you have all of them.
- The exam is closed book, closed notes, and closed neighbor. Any communication with a neighbor will be interpreted as a cheating attempt and will be dealt with accordingly: at least a **zero** on the test!
- Your handwriting should be readable so it can be graded. Include all work or justification for partial credit.
- Use your time carefully. If you feel you are stuck, skip to the next question.

Problem 1	**15**		
Problem 2	**10**		
Problem 3	**10**		
3.a		5	
3.b		5	
Problem 4	**15**		
4.a		5	
4.b		10	
Problem 5	**15**		
5.a		5	
5.b		5	
5.c		5	
Problem 6	**10**		
6.a		5	
6.b		5	
Problem 7	**10**		
7.a		5	
7.b		5	
Problem 8	**10**		
Problem 9	**10**		
9.a		5	
9.b		5	
9.c		5	
9.d		5	
Total	**100**		

Problem 1. True/False Questions (15 = 15 × 1)

Mark your answer in the table below. Note that there is a "**–1**" **penalty** associated with each wrong answer!

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T															
F															

1. In paging system, page fault means a process page contains faulty data
2. Increasing block size will increase the disk space utilization
3. A thrashing process is a process that causes minimum number of page faults
4. Paging has to do with moving the entire process from main memory to the disk.
5. Compaction is a way to solve the internal fragmentation problem
6. Internal fragmentation in file blocks is reusable
7. Paging and swapping help in increasing the degree of multiprogramming
8. Compaction is a way to solve the external fragmentation problem in disk space allocations.
9. Aging is a technique used for solving the starvation problem in priority scheduling.
10. The Table Lookaside Buffer is a way to solve the large page table problem
11. Paging system is a one-linear address space system
12. The second chance page replacement algorithm gives every page a second chance
13. Internal fragmentation in paging is a waste of space.
14. Linked disk space allocation method suffers from external fragmentation.
15. Removing a symbolic link will remove the physical file

Problem 2. Processes (10 points)

How many processes does the following piece of code generate? Draw a graph showing the parent-child relations between processes. Assign process ids start at 100.

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

#define N 4

int i;
pid_t pid;

for (i = 0; i < N; i++) {
  if ((pid = fork()) == -1)
    break;
  }
fprintf(stderr, "This is process %ld with parent %ld\n",
        (long) getpid(), (long) getppid());
```

Problem 3. Scheduling (10 points = 5 + 5)

Measurements of a certain computer system have shown that the average CPU burst time is *T* msec. A process context switch requires *S* msec, which is effectively overhead (*i.e.*, wasted time). The efficiency of a scheduling algorithm is defined as the ratio of useful time (when instructions from the process are executing) to total time. As such, the efficiency is the percentage of time the CPU is doing useful work as opposed to switching contexts.

a. Which scheduling algorithm has the highest efficiency rating? Justify your answer.

b. Under what conditions, does the Round Robin scheduling algorithm match the efficiency of the algorithm identified above? Assume that the Round Robin algorithm has a time quantum *Q*. Justify your answer.

Problem 4. Synchronization I (15 points = 5 + 10)

Three concurrent processes need to coordinate their actions in order to achieve their task. Each process generates an event. However, it is critical that these events occur in one of two sequences that repeat themselves:

[(E1E2 | E2E1) E3 E3]+

Define semaphores and use the *P* & *V* primitives so the events always occur in the specified order.

a. Declare and initialize the needed semaphores.

b. Fill in the blanks in the table below.

P_1	P_2	P_3
…	…	…
do	**do**	**do**
…	…	…
E-1	*E*-2	*E*-3
…	…	…
while *true*	**while** *true*	**while** *true*

Problem 5. Memory Management (15 = 5 + 5 + 5)

a. In memory or disk space allocation, what is the *key* difference between internal and external fragmentation.

b. What does it mean when we say "most of the time, processes exhibit *locality of reference*"?

c. A computer has 48-bit virtual addresses with 8 KB pages and 32-bit physical addresses. How many entries are needed for the page table?

Problem 6. Memory Management II (20 = 5 + 5 + 5 + 5)

Assume that the list of holes in a variable partitions memory system contains the following entries (in the given order): m_1 = 95 KB, m_2 = 275 KB, m_3 = 110 KB, m_4 = 210 KB, m_5 = 325 KB, m_6 = 55 KB. Consider the following sequence of requests (in the given order): A =- 105 KB, B 215 KB, C = 50 KB, D = 210 KB. Determine which holes would be allocated to which request by each of the following schemes. Show you results in a separate table for each scheme.

a. First fit.

b. Best fit.

Problem 7. Virtual Memory (10 = 5 + 5)

Let *R* be the following reference string: 0, 3, 1, 4, 1, 5, 1, 6, 0, 5, 2, 6, 7, 5, 0, 0, 0, 6, 6, 6, 6.

a. Given a page frame allocation of 3 and assuming that the primary memory is initially unloaded, how many page faults will the given reference string incur under FIFO?

	0	3	1	4	1	5	1	6	0	5	2	6	7	5	0	0	0	6	6	6	6
f_0																					
f_1																					
f_2																					

b. Given a page frame allocation of 3 and assuming that the primary memory is initially unloaded, how many page faults will the given reference string incur under LRU?

	0	3	1	4	1	5	1	6	0	5	2	6	7	5	0	0	0	6	6	6	6
f_0																					
f_1																					
f_2																					

Problem 8. Consistency Models (10)

Consider two concurrent processes, p_1 & p_2, performing the following operations on two shared variables, x & y.

```
initial: x = 0; y = 0;
p1: { x = 1; a1 = y; b1 = x; }
p2: { y = 1; a2 = x; b2 = y; }
```

At the end of the execution, the value of each of the local variables a_1, b_1, a_2, & b_2 can be zero or one. For the 16 possible combinations of these values, determine which ones are valid under sequential consistency.

Problem 9. File Systems Project (10 + bonus = 4 × 5)

a. What are the block types in the file system project and what are they used for?

b. What are the implications of a change in the block size from 512 to 1024?

c. Cite at least one typo in the project's header files that produced a compilation error.

d. Why did you have to always use absolute paths for filenames? How can one get around this problem?