University of Toronto
Faculty of Applied Science and Engineering

ECE 443 - System Software

Final Examination, Fall 2001

Examiner:	T. Savor
Type:	Closed Book, Numeric Calculators Permitted.
Total Marks:	100
Allotted Time:	180 minutes
Instructions:	Write all answers in space provided. If additional space is required, use blank pages at end of booklet.

First Name: _____

Last Name: _____

Student Number: _____

Q1. _____

Q2. _____

Q3. _____

Total Mark [ ]

Question 1 – General

a) An operating system consumes CPU and memory resources just like an application program. Explain why operating systems are used in light of the fact that they consume hardware resources. [4 marks]

b) What is the purpose of implementing your code for lab 5 having two levels: a library level and a kernel level, even though it is not really necessary to do so? After all, executing a trap() is simply calling a C function [4 marks].

c) What is the benefit to the operating system of using a device driver to communicate with devices such as disks? [2 marks]

d) Why is it necessary to "flush" the TLB on every context switch? [2 marks]

e) What is memory fragmentation? Explain how it occurs in both static and dynamic memory allocation schemes. [4 marks]

f) Explain the notion of *blocking*. Explain why blocking is implemented by an operating system (3-4 sentences) [3 marks]

g) Several approaches to implementing mutual exclusion were discussed in class including Peterson's algorithm, hardware approaches, semaphores, critical regions and monitors. Can signals be used to implement mutual exclusion? Justify your answer (3-4 sentences total) [3 marks]

h) A *barrier* is a synchronization primitive that forces a set of cooperating processes to wait for one another until all of them reach the barrier. For example, none of the processes P1, P2 and P3 below will execute past the call to Barrier() until all of them have made this call in their respective codes. The barrier is created and initialized to the number of processes that are expected to "go through" it.

Write an implementation (in C or pseudocode) of a barrier primitive that uses busy wait to achieve barrier synchronization. Include your definition (e.g. struct in C) of what the barrier consists of. Show the implementation of the functions Barrier_init() and Barrier(). [10 marks]

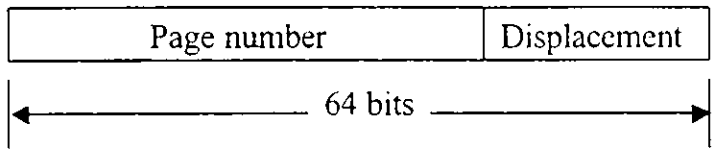
```
        P1                  P2                  P3

Barrier_init(3);    Barrier_init(3);    Barrier_init(3);
   :                   :                   :
   :                Barrier();             :
Barrier();             :                   :
   :                   :                Barrier();
   :                   :                   :
```

Question 2 – Memory Management

a) Consider a computer system with a processor that has a 30-bit address bus, 16 Mbytes of memory and a page size of 2 Kbytes. How many frames are there in memory, maximum? [2 marks]

b) What is the maximum number of pages a process can have? [2 marks]

c) Consider a machine with a 64-bit address space and a page size of 16K. Compute (1) the size of the page number (in bits) and (2) the size of the displacement (in bits). [2 marks]



d) Calculate the length of the page table for a 2Mb (2048K) process and a page size of 2K. [3 marks]

e) Modern operating systems (such as Solaris) use segments instead of a page table (as discussed in class). A segment maps process addresses (i.e. pages) to physical memory addresses (i.e. frames). Unlike an entry in the page table, a segment can translate an arbitrarily long contiguous segment of memory. Explain the principal advantage of the segment approach over the page-table approach. [4 marks]

f) Explain what is meant by a segmentation fault. [2 marks]

g) Explain how the *dirty bit* aids in improving the performance of an operating system. [3 marks]

h) What happens to the contents of the translation lookaside buffer (TLB) after a context switch [2 marks]

i) In a demand-driven paging system, when a process is first initiated typically none of its pages are loaded from disk into memory. Explain why this is the case. [2 marks]

A portion of the page table for a process running on a 32-bit CPU with 256 Mbytes of RAM and a page size of 8 Kbytes. What is the physical address corresponding to the logical address hexadecimal 00006714? [3 marks]

0	3
1	4
2	1
3	0

Consider the following page reference sequence:

1, 2, 3, 4, 2, 6, 6, 3, 2, 4, 2, 3, 7, 6, 3, 4

How many page faults would occur for the following replacement algorithms assuming 3 frames are availabie for allocation? Assume that all frames are initially empty, so your first unique pages will all cost one page fault each.

i) LRU replacement [4 marks]

j) Optimal [4 marks]

k) Consider a machine that makes use of a demand paging system (i.e. a TLB, a page table and the ability to page to/from disk). Assume that the CPU issues a request for a memory address that has been moved to disk and hence does not reside in the TLB or main memory. List the steps that the demand paging sub-system goes through to get access to the address [6 marks]

l) In a TLB, pages and their associated frames are stored in random locations. A linear search through the TLB would be too slow. Explain how addresses are accessed in a TLB to optimize performance. [3 marks]

m) A computer has a cache, main memory and a disk used for virtual memory. If a referenced word is in cache, 20ns are required to access it. If it is in main memory but not in the cache, 60ns are needed to load it into the cache and then the reference is started again (i.e. the word is then loaded by the processor from the cache). If the word is not in main memory, 8 milliseconds are required to fetch the word from disk, followed by 60 ns to copy it to the cache and then the reference is started again. The main-memory hit ratio is 0.6. The desired average access time of the memory sub-system is 100,000ns (100us). What is the required cache hit ratio? [10 marks]

Question 3 – Information Storage on Disk

Consider a physical disk that has (and/or is formatted to have) 512 byte blocks, 64 blocks/sector, 1024 tracks/surface and 16 surfaces. The disk is formatted with an I-node-based file system. I-nodes are created statically as part of the format and consume 1 block each. The I-node structure is as described in class. Assume the file system contains 100,000 I-nodes. Free blocks are kept track of by having one I-node link to all free blocks. As files are written to disk, free blocks are taken and allocated to an I-node representing a file.

a) What is the physical capacity of the disk (in bytes)? [2 marks]

b) What is the formatted capacity of the disk (i.e. the space available to the user)? [2 marks]

c) How many individual files can be stored on the disk? [2 marks]

d) List the major parts of an I-node as discussed in class (i.e. show as a data structure). [4 marks]

e) What governs the maximum size of disk this file system can support? [1 mark]

f) What is the maximum size of disk that this file system can support? [2 marks]

g) What governs the largest file this file system can support? [1 mark]

h) What is the largest file this file system can support? [2 marks]

ANSWER OVERFLOWS (1/2)

ANSWER OVERFLOWS (2/2)