NAME: ____ (FIRST NAME FIRST) **SCORE**: _____

COSC 4330 FINAL EXAMINATION DECEMBER 12, 2006

THIS EXAM IS CLOSED BOOK. YOU CAN HAVE ONE PAGE OF NOTES. UH EXPELS CHEATERS.

1. Complete the following sentences: (4×5 points)

a) __Messages____

are an example of a consumable resource in a distributed computing system.

b) The major advantage of ***tickets*** over ***access control lists*** is that

checking a ticket is faster than consulting an access control list _______________.

c) The major disadvantage of the ***LRU page replacement policy*** is that

its overhead it so large it cannot be efficiently implemented __________________.

d) File descriptors ____

are one example of tickets in the UNIX file system.

2. A 32-bit Berkeley UNIX file system has a block size of 8 kilobytes.

a) What is the maximum number of ***blocks*** in a file ___________ 4GB/8K=512K **or 2^{19} blocks**

b) How many of these ***blocks*** can be accessed with one level of indirection? (5 points)

____ 8K/4 = 2K or 2^{11} **blocks**

3. Assume that you are asked to design a two-level page table organization for a computer having 32-bit addresses and a page size of 4 kilobytes. Given that you want each of your indexes to fit in a single page frame,

a) How many entries will your ***master index*** contain? (5 points) ___ 4K/4 = 2^{10} or 1,024 **entries**

b) How many entries will your each of your ***subindexe***s contain? (5 points) _ 2^{10} or 1,024 **entries**

Total _____/40

4. Which of the following statements are ***true*** or ***false*** (2 points) and ***why***? (3 points)

a) Monitor conditions can only have positive values.

FALSE, monitor conditions have no value.

b) The ***valid bit*** indicates which pages must be *written back to disk* when they are expelled.

FALSE, it indicates whether the page is in main memory or not.

c) It is easier to eliminate a *circular wait* condition than an *hold and wait* condition.

TRUE, we do not have to allocate all the resources at the same time.

d) The LRU policy always expels the page that has been in memory for the longest period of time.

FALSE, it always expels the page that has not been accessed for the longest period of time. (FIFO always expels the page that has been in memory for the longest period of time.)

e) A good page fault rate for a virtual memory system is one page fault every one thousand to two thousand references.

FALSE, it is much too high.

f) We can prevent deadlocks in message-passing systems by denying the mutual exclusion condition.

FALSE, that would imply that a message can be shared by its sender and its receiver, that is, being not yet sent and already sent at the same time.

5. Define ***internal fragmentation***. (5 points) What is its impact on the performance of (a) virtual memory systems and (b) file systems? (2×5 points) What does the Berkeley file system do to address this issue? (5 points)

Internal fragmentation occurs because virtual memory systems allocate to each process an integer number of ages and file modern file systems allocate to each file a fixed number of blocks. Assuming that all process sizes and all file sizes are equally likely, the average number of bytes lost to internal fragmentation is half a page per process and half a block per file.

Internal fragmentation has little impact on the performance of virtual memory systems because even the smallest processes comprise several pages.

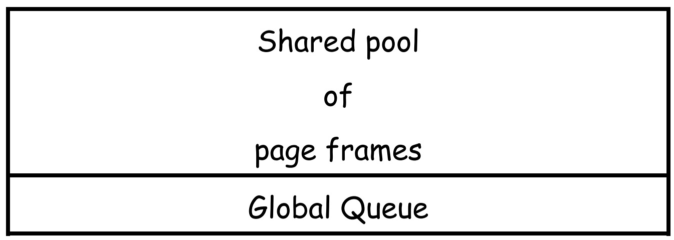
This is not true in file systems as many file systems include many files whose size is less than a block.

The Berkeley "fast" file system (FFS) addresses this issue by using block fragments (½ or ¼ block) to store small files and endings of larger files.

> Note that the answer has four short paragraphs and each paragraph corresponds to a specific point of the question.

6. Describe in some detail the Mach page replacement polic**y.** (10 points for an answer with the correct diagram).

Mach divides its main memory into a shared pool of page frames shared by all processes and one global queue from which pages can be reclaimed.



The pool of page frames is managed according to a Global FIFO policy but pages expelled from the pool by the FIFO policy are given what essentially amounts to a *second chance.* Instead of being immediately expelled from the main memory, they go at the end of the global queue and their *valid* bit is reset to *zero*.

Whenever a page fault occurs, the page fault handler looks first at the global queue to see if the missing page is cannot be found there. If this is the case, the page is returned to the FIFO pool. If another page needs to be expelled from the FIFO pool to make space for the returning page it will go at the end of the Global queue and another page put at the end of the global queue.