Name _____

Student Number _____

Instructor: S. Qiao

COMP SCI 3MH3/6MH3

Day Class
Duration of examination: two hours
McMaster University Final Examination December 2007

This examination paper includes **6** pages and **11** questions. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator.

SPECIAL INSTRUCTIONS: This paper must be returned with your answers. Use of McMaster standard (Casio-FX991) calculator only is allowed.

1. (2 marks) Which of the following eliminates external fragmentation? Justify your answer.
   (a) pure paging
   (b) segmentation
   (c) paged segmentation

   Answer: (a). No fragmentation between pages.

2. (2 marks) In a demand-paging system, which of the following statements is true?
   (a) A TLB miss is always followed by a page fault.
   (b) A page fault is always preceded by a TLB miss.
   (c) A TLB hit is always followed by a page fault.
   If all are false, you may answer none.

   Answer: (b)

3. (2 marks) Which of the following statements is false?
   A file system with more structures for files
   (a) provides more support for users
   (b) is more flexible
   (c) is more complicated to implement
   If all are true, you may answer none.

   Answer: (b)

4. (2 marks) Which of the following is not stored on disk?

   (a) file header (inode)
   (b) bit map for sectors
   (c) OpenFile
   If all of them are stored on disk, you may answer none.

   Answer: (c)

5. (2 marks) In network protocol layers, delivery (reliability) is guaranteed at

   (a) transport layer
   (b) datalink layer
   (c) network layer

   Answer: (a)

6. (8 marks) Assuming a pure paging system on a 16-bit machine with page size of 256 bytes and the following process page table of size 4:

valid	used	dirty	readOnly	virtual page number	physical page number
T	T	F	T	0_{16}	A_{16}
T	T	F	T	1_{16}	B_{16}
F	F	F	F	2_{16}	0_{16}
T	T	T	F	3_{16}	12_{16}

Translate the following virtual addresses into physical addresses and indicate exceptions if there is any. All numbers are hexadecimal.

(a) A_{16} read

Answer: Virtual page number 0, physical address A0A.

(b) 123_{16} write

Answer: Virtual page number 1, read only violation.

(c) 234_{16} write

Answer: Virtual page number 2, invalid, page fault.

(d) 5678_{16} read

Answer: Out of virtual page number bound, bus error.

7. (6 marks) Given the following inverted page map table

use	physical page number	virtual page number	space
T	0	0	1234
T	1	3	1234
F	2	14	5678
F	3	0	5678

Suppose that my process (`mySpace = 1234`) is running and a memory reference to virtual page 5 has caused a page fault and the physical memory is full. The clock algorithm is applied to the above inverted page table for page replacement. Assuming the current position of the clock hand is at the first entry, show the content of the inverted page table after the clock algorithm is applied for page replacement.

Answer:

use	physical page number	virtual page number	space
F	0	0	1234
F	1	3	1234
T	2	5	1234
F	3	0	5678

8. (6 marks) Give the structure of the global open file table entry in a multithreaded file system.

Answer:

```
bool  valid;
Lock* fileLock;
int   sector;
bool  isRemoved;
int   count;
FileHeader* hdr;
```

9. (6 marks) Suppose that Nachos file header structure uses one single indirect, one double indirect, and rest for direct data blocks, as the following:

```
class FileHeader {
  public:

  private:
    int numBytes;
    int numSectors;
    int dataSectors[NumDirect];
    int singleIndirect;
    int doubleIndirect;

};
```

What is the maximal file size (in bytes) in this system? Recall that sizeof(int) is 4, the sector size is 128 bytes and the file header fits in exactly one sector. Explain and show your calculations.

Answer:
NumDirect $= (128 - 4 \times 4)/4 = 28$.
The single indirect sector has $128/4 = 32$ pointers.
The double indirect sector has $32 \times 32 = 1024$ pointers.
Max size: $(28 + 32 + 1024) \times 128 = 138,752$ bytes.

10. (8 marks) Describe the steps taken by the file system function

```
OpenFile* fileSystem::Open(fileName)
```

in a multithreaded file system.

```
lock directory
search directory for header number using fileName
lock global open file table
search open file table using header number
if entry found
    count++
else
    create a new entry for the file
end
get the pointer to the entry
unlock open file table
unlock directory
construct openFile
return openFile*
```

The following question is for CS6MH3.

11. (8 marks) Describe the steps taken by the file system in a single thread system to create a file, given the file name (a string) and the initial size (an integer):

```
Create(char *name, int initialSize)
```

Assuming the current directory is already open.

Answer:

1. Load in directory file, make sure the file doesn't already exist;
2. Load in bit map file, allocate a sector for the file header;
3. Allocate disk sectors for the data blocks for the file;
4. Add the name to the director;
5. Flush the new file header to disk;
6. Flush the changes to the directory and bitmap back to disk;
7. Clean memory.