6. Suppose that you wish to design a virtual memory system with the following characteristics:

   - The size of a page table entry is 4 bytes.
   - Each page table must fit into a single physical frame.
   - The system must be able to support virtual address spaces as large as $2^{38}$ bytes (256 GB).

   (a) (2 marks) Suppose that you decide to use a multi-level paging scheme with no more than two levels of page tables. What is the minimum page size that your system must have?

   (h) (2 marks) Draw a diagram indicating how the bits of a virtual address will be interpreted by the address translation mechanism. Indicate which bits (and how many) are used to index the page tables at each level, and which bits form the page offset. Draw neatly.

(c) (2 marks) Suppose instead that you are willing to use a three-level paging scheme. What is the minimum page size that your system must have in this case?

(d) (4 marks) Draw a diagram indicating how the bits of each virtual address will be interpreted by the address translation mechanism. Indicate which bits (and how many) are used to index the page tables at each level, and which bits form the page offset. Draw neatly.

7. (4 marks) Suppose that an operating system supports concurrent threads. Suppose further that a process is running, and that it contains two threads, called ThreadX and ThreadY. These are system threads, scheduled by the operating system. Thread X is executing the ThreadXFunction, shown below. Thread Y is executing the ThreadYFunction. In these functions, ReadInput, WriteOuput and Exit are system calls. The ReadInput call reads a character from the input device into tbe specified buffer. It is a blocking call - the caller will block for a short time while an input character is obtained. The WriteOuput call writes a character from the specified buffer to the output device. It is a non-blocking call. The Exit system call terminates the calling process. For the purposes of this question, you should assume that the operating system uses a scheduling quantum large enough that a running thread will never use its entire quantum.
Code for Thread X

```
void ThreadXFunction() {
 char buf;
for(i = 0;i < 5;i + +) {
 ReadInput(&buf);
WriteOutput(&buf);
WriteOutput(X);
}Exit(0);
}
```

Code for Thread Y

```
void ThreadYFunction() {
 char buf;
for(i = 0;i < 5;i + +) {
ReadInput(&buf);
WriteOutput(&buf);
WriteOutput(Y);
}Exit(0);
}
```

Assume tbat tbe following sequence of input characters is typed on the input console:
a b c d e f g h i j
Show a sequence of characters that could be generated on the output console by this process.

Entry Number: ______________ Name : ______________________________

8. (10 marks) Consider the following synchronization problem. A boat delivers bananas to a desert island which is inhabited only by monkeys. The boat delivers one load of bananas at a time, but the load has a variable number of bananas in it; when it arrives, it calls the *deliver(int nbananas)* function, where *nbananas* specifies how many are delivered (at least one is always delivered). Bananas are placed in a crate on the beach. When a monkey is hungry, it goes to the crate. If there are enough bananas in the crate, it takes what it needs; otherwise, it joins a queue of monkeys waiting for the boat. The monkeys are greedy and are not always satisfied with just one banana. A monkey wanting to get *nbananas* bananas calls *monkey(int nbananas)*. Calling monkey should cause it to wait in the queue until it gets *nbananas* bananas, then leave. A monkey will to stay at the head of the queue until it has all the bananas it wants, rather than, for instance, grabbing 3 out of 4 bananas and then going to the back of the queue to wait for its final banana. Devise a solution for tbe monkeys-and-bananas problem which follows the rules given above, and incorporates correct concurrency control using only semaphores.

Blank Page

Blank Page

Major Exam (2 May 08) CSL373:Operating Systems Time : 1300-1500

Answer the questions in the spaces provided on the question sheets. This is a closed book and closed notes exam. I donot answer any technical questions.

Name : ______________________________

| Question: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total |
|---|---|---|---|---|---|---|---|---|---|
| Points: | 19 | 6 | 6 | 5 | 10 | 10 | 4 | 10 | 70 |
| Score: | | | | | | | | | |

1. (a) (2 marks) In most file systems, an update to a block of a file is written back to the same block on disk. Some modern high-performance file systems just write the updated block of data into a nearby available disk block. Briefly describe one advantage and one disadvantage of this strategy.

   (b) (2 marks) Gridlock is a term describing a traffic situation in which there are so many cars in the streets and the intersections that essentially no car can move any direction because other cars are in the way. Explain how this is the same as deadlock in an operating system by showing how each of the four conditions for deadlock hold in this situation.

Entry Number: ____________________ Name : ______________________________

(c) (2 marks) What are the differences between traps and interrupts? What is the use of each?

(d) (2 marks) In the linux kernel, the input and output consoles are asynchronous devices. Explain briefly what it means for a device to be asynchronous.

(e) (2 marks) To create a new process, the exec system call is used in linux. To implement Exee, the operating system must determine the size of the new process's virtual address space. Explain, briefly, how this is accomplished.

(f) (2 marks) Briefly explain the difference between the user mode and the priviledged mode of a CPU. Why is that difference important for the goals of an operating system?

(g) (2 marks) What is the difference between FIFO and round-robin schcduling?

(h) (3 marks) Briefly explain the usage and benefits of a translation lookasidc buffer (TLB) for paging

(i) (2 marks) An operating system uses a variable resident set policy with a global page replacement algoritlım. Why may it be necessary to keep track of both a global and a per-process page fault frequency?

2. (6 marks) Suppose that you have a UNIX file system where the disk block size is 1kB, and an inode takes 128 bytes. Disk addresses take 32 bits, and the inode contains space for 64 bytes of data (a recent optimization), 8 direct addresses, one indirect, one double-indirect and one triple- indirect (the rest of the space in the inode is taken up with other information such as ownership and proteetion). An index block is the same size as a disk block. How much space (including overhead) do files that are: a) one (1) byte long, b) 1025 bytes long, c) 65536 (64KB) bytes long, and d) 1048576 (1MB) bytes long require?

Entry Number: ______________________ Name : ______________________________

3. (6 marks) Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time we have added an associative memory that reduces access time to one memory reference if the page table entry is in the associative mcmory. Assume that 80 percent of the accesses are in the associative memory and that of the remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time.

Entry Number: ________________ Name : ________________________________

4. (5 marks) Suppose that an operating system uses a preemptive multi-level feedback scheduler with (for simplicity) an infinite number of levels. New processes start at level 0. A process that uses its entire quantum at level i gets demoted to level i + 1. A process that does not use its entire quantum moves back to level 0, regardless of the level that it had previously been at. The scheduling quantum for level i is 2i time units. Initially, there is one process in the system. It requires a total of 9 time units of running time before it exits. After every two units of running time, this initial process will create a new process. These new processes require only 1 time unit of running time each before they exit. (Note: process creation does not cause preemption of the creating process.) None of these processes perform any blocking system calls - once started, they run until they exit or until they are preempted. What will be the turnaround time of the initial process? That is, how long will it take for the initial process to finish? Justify your answer, preferably with a diagram.

5. Two processes, Process A and Process B, are running in a system. Their page tables are shown below. Each page table entry includes a frame number and use (U), dirty (D), and valid (V) bits. An entry with V = 1 is a valid entry. For simplicity, assume the page size is 100 bytes.

| Page Table for Process A | | | | | Page Table for Process B | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Page | Frame | U | D | V | Page | Frame | U | D | V |
| 0 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 4 | 1 | 1 | 1 | 1 | 7 | 1 | 0 | 1 |
| 2 | 5 | 1 | 0 | 1 | 2 | 8 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 | 1 | 4 | 0 | 0 | 0 | 0 |
| 5 | 9 | 1 | 0 | 1 | 5 | 3 | 1 | 1 | 1 |

(a) (2 marks) To which physical address does virtual address 410 of Process A map? If it does not map to a physical address, write "does not map"

(b) (2 marks) To which physical address does virtual address 64 of Process B map? If it does not map to a physical address, write "does not map".

(c) (2 marks) To whieh physical address does virtual address 410 of Process B map? If it does not map to a physical address, write "does not map".

(d) (2 marks) To which virtual address, in which process, does physical address 90 map? If it docs not map to a virtual address, write "does not map".

(e) (2 marks) To which virtual address, in which process, does physical address 890 map? If it does not map to a virtual address, write "does not map".