Name, SURNAME and ID ⇒

Middle East Technical University
Department of Computer Engineering

CENG 334

Section 1 and 2
Spring '2007-2008

Final

- **Duration: 120** minutes.
- **Exam:**
  - This is a **closed book**, **closed notes** exam. No attempts of cheating will be tolerated. In case such attempts are observed, the students who took part in the act will be prosecuted. The legal code states that students who are found guilty of cheating shall be expelled from the university for **a minimum of one semester**!
- **About the exam questions:**
  - The total number of points in this exam is 110.
  - The points assigned for each question are shown in parenthesis next to the question.
  - Due to the penalty policy stated for the first question, you may get a negative point.
  - Whereever available, use the boxes to write down your answers.
- **This booklet consists of *8* pages including this page. Check that you have them all!**
- **GOOD LUCK !**

Question 1		Question 2	
Question 3		Question 4	
Question 5		Question 6	
		Question 7	
		Total ⇒	

1 (21 pts)

For each of the statements given below, indicte whether it's true or false and explain why in **one** sentence. Wrong answers will be penalized with -3 points.

(a) (3 pts) A preemptive round robin scheduling policy will result in better CPU utilization than a nonpreemptive round robin scheduler.

(b) (3 pts) Threads belonging to a single process have separate register sets but the same address spaces.

(c) (3 pts) During a context switch, it is important to remember to set the return register (e.g., `%eax`) value to zero to indicate that no error occurred during the switch.

(d) (3 pts) A virtual memory system that uses paging can suffer from internal fragmentation.

(e) (3 pts) I/O devices can signal the completion of an asynchronous I/O operation (e.g., a disk block read) by having their device driver execute a trap instruction.

(f) (3 pts) DMA is used to offload from the operating system (processor) the overhead of copying data between I/O devices and main memory.

(g) (3 pts) In Unix-derived filesystems, soft links are implemented using *link counts* in the file's inode.

2 (15 pts)

Consider the following page reference string:

A, B, A, C, D, B, E, C, D, E, A, C, A, B, D, C, A, B, D, A, C

How many page faults would occur for the following replacement algorithms, assuming that there are three page frames in the machine? Show your work using the templates below (i.e., show what page goes in what frame and mark where a page fault occurs)! Assume that none of the process's pages are loaded in memory when the above trace starts.

(a) (5 pts) FIFO

Access	A	B	A	C	D	B	E	C	D	E	A	C	A	B	D	C	A	B	D	A	C
Frame 1																					
Frame 2																					
Frame 3																					
Fault?																					

Total number of faults:

(b) (5 pts) LRU

Access	A	B	A	C	D	B	E	C	D	E	A	C	A	B	D	C	A	B	D	A	C
Frame 1																					
Frame 2																					
Frame 3																					
Fault?																					

Total number of faults:

(c) (5 pts) OPT

Access	A	B	A	C	D	B	E	C	D	E	A	C	A	B	D	C	A	B	D	A	C
Frame 1																					
Frame 2																					
Frame 3																					
Fault?																					

Total number of faults:

3 (15 pts)

Implement a new synchronization primitive called a `Lightswitch`. This primitive should provide synchronization among threads in analogy with the problem where the first person into a room turns on the light and the last one out turns it off. Implement the methods of this primitive using semaphores using the skeleton given below:

```
class Lightswitch{
    init(){ // declarations and initializations
        counter = 0;
        mutex = semaphore(1);
    }
    lock(semaphore switch){









    }
    unlock(semaphore switch){









    }
}
```

The `init` method initializes the class.

The `lock` method takes one parameter, a semaphore called `switch`, that it will check and possibly *turn it on*.

When the last thread calls `unlock`, the switch is *turned off*.

Exemplary use: The problem mentioned as an analogy above can be solved as:

```
// declarations and initializations
roomLightswitch = Lightswitch();
roomEmpty = semaphore(1);

// person code
use_room{
    roomLightswitch.lock(roomEmpty);
    do_read();
    roomLightswitch.unlock(roomEmpty);
}
```

4 (10 pts)

Old Unix-like systems did not provide any primitives intended for synchronization of user processes, but they did provide pipes.

(a) (5 pts) Explain how to implement a semaphore (discuss how would implement up() and down()) using a pipe.

(b) (5 pts) A semaphore implemented as a pipe may deadlock in a situation where a conventionally implemented semaphore would not. Describe the situation.

5 (10 pts)

Disk requests come into the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. Assume that the disk has 100 cylinders. A seek takes 6msec per cylinder moved. Assuming that the arm is initially at cylinder 20, write down the access sequence, and compute the average seek time for the request sequence given above for the following policies:

(4) (4 pts) Shortest Seek Time First (SSTF)

Access sequence:
Average seek time:

(3) (3 pts) SCAN

Access sequence:
Average seek time:

(c) (3 pts) C-SCAN

Access sequence:
Average seek time:

6 (25 pts)

Answer the following questions regarding filesystems:

(i) (10 pts) Consider the organization of a FAT-like filesystem called FAT8, which used 8 bits for disk addressing. In a system with 1-KB block size, what's (a) the size of the file allocation table, (b) the size of the largest disk partition it can support, and (c) the size of the largest file it can support? (You can write the results as a sum of multiplicative terms.)

The size of the file allocation table:
The size of the largest disk partition it can support:
The size of the largest file it can support:

(ii) (15 pts) Consider the organization of a UNIX-like filesystem that employs a multi-level indexed file inode structure. Assume that there are 12 direct block addresses, and one single-, one double-, and one triple-indirect block addresses in the inode. Assume that disk addresses are 32-bits and that the disk has been formatted to use 512-byte blocks for this filesystem. Assume that all directories fit in a single disk block.

(a) (5 pts) What is the maximum size file that can be supported by the system? (You can write the result as a sum of multiplicative terms.)

(b) (5 pts) Write down the sequence of disk operations needed to fetch the i-node of the file */home/erol/final.pdf* in this system? Assume that the i-node for the root directory is in memory but nothing else in the path.

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

...

(c) (5 pts) Assuming that no information other than the file i-node is present in main memory at the time of the operation, how many disk blocks must be accessed to access 4-kilobytes of a 2-gigabyte file starting at offset 1MB? Show your work.

7 (14 pts)

(a) (4 pts) In public-key cryptography each person can generate a public and a private key to encrypt and decrypt messages. Consider the following case: Leyla and Mecnun have previously exchanged public keys. What are the operations required for Leyla to send her confidential message to Mecnun, and for Mecnun to read it. Cross the options (written as `this/that`) in the following sentences:

(b) (5 pts) Assume that you have a multi-core machine. Each of the cores has its own private L1 and L2 caches, but use the same bus to access the shared memory. You set out to port your favorite OS that was developed for a uniprocessor machine onto this machine to support symmetric multi processing(SMP). In the original version of your OS, you realized that there were three different mutex implementations based on 1)disabling interrupts, 2)spinlocking, and 3)test-and-set instruction. Which of these implementations would you choose to use, and what support would you look for at the hardware level. Hint: There are more than one answers to this question.

(c) (5 pts) Support for virtualization can be implemented within the host OS as a type 2 hypervisor running in user mode. In such a system, when the guest OS (which is running as a user-level program in the host OS) executes a privileged instruction such as I/O, what does the hypervisor do?