Operating Systems

Final Exam, Spring 2014

Date: May 16, 2014 **Marks:** 80 **Time: 3 hrs**

Question 1 (10+5+5 marks)

a) Show execution of the LRU Page replacement algorithm on the following page reference string. Assume only three frames are available, and these are initially empty. If the LRU algorithm is unable to guide you at some point then use FIFO.

1 2 3 1 4 5 6 7 5 8

This reference string says that a running process first needs page 1, then page 2, and so on.

b) Consider a paging system with a memory access time of 200 nano-seconds, and a TLB hit ratio of 90%. Calculate the effective access time.

c) Write down the steps in handling a page fault.

Question 2 (5+5+5+5)

a) How many processes would be created by the following code?

```
for (int i = 1; i <= 5; ++ i)
        fork();
```

b) Give all possible outputs of the following program:

```
int x = 0; // global
```

// Thread 1	// Thread 2	// Thread 3
x = x + 2;	x = x + 3;	cout << x;

c) Name the four necessary conditions for a deadlock.

d) Which of the following scheduling algorithms may suffer from starvation?

i. FIFO
ii. RR
iii. Priority Scheduling
iv. SJF
v. Multi-level Queue Scheduling

Question 3 (10+10 marks)

a) Consider a file stored in the following FAT with starting block 5:

3	6	0	7	8	9	2	4	eof	1
0	1	2	3	4	5	6	7	8	9

Now translate the following logical block numbers of the file into the physical block numbers:

(i) 0 (ii) 2 (iii) 5 (iv) 7 (v) 9

b) Consider a file system using Two-level Index with a block size of 4KB and a pointer size of 4 bytes. Write down a C/C++ function to compute the physical block number for a given logical block number of a particular file. Following is the function prototype:

```
int translate( int start, int logBlk )
```

The first parameter here points to the file index block and the second is the required logical block number. The function returns the corresponding physical block number after computation.

You may use the following system call to read a block: read(int blk, int* a). The first parameter here is the required physical block number and the second parameter is a pointer to the target array/buffer.

Question 4 (20 marks)

Consider a nursery where children eat sweets from a large pot. When the pot is empty the next interested child calls the chef, and the chef fills the pot again. This scenario can be modeled in a multi-threading environment as follows:

// child
while (true)
    getSweetFromPot
    eat ...
	// chef
	while (true)
	    relax ...
	    putSweetInPot

You need to synchronize the children and the chef using Semaphores. The "child" code is executed by multiple threads; on the other hand The "chef" code is executed by a single thread. Assume the chef places N sweets in the pot at a time, and each child takes only a single sweet at one time.