Operating Systems, Spring 2009

Final

2:10pm ∼ 3:50pm, Tuesday, June 16, 2009

INSTRUCTIONS:

1. This is a *closed-book* exam.
2. Try to solve all of the problems.
3. Try to give short answers. (Hint: An answer need not always be longer than the question.)
4. No cheating.
5. Please hand in both the exam sheet and the answer sheet.
6. Please note that unless otherwise stated, all the line numbers for the program listings are for reference only.

1. (20%) Consider the two-dimensional array `A`:

   ```
   int A[][] = new int[200][200];
   ```

   where each integer occupies 4 bytes and `A[0][0]` is at location 200, in a paged system with pages of size 200 bytes. A small process is in page 0 (locations 0 to 199) for manipulating the matrix; thus, every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array initialization loops, using LRU replacement, and assuming page frame 1 has the process in it, and the other two are initially empty:

   (a)
   ```
   for (int i = 0; i < 200; i++)
       for (int j = 0; j < 200; j++)
           A[i][j] = 0;
   ```

   (b)
   ```
   for (int j = 0; j < 200; j++)
       for (int i = 0; i < 200; i++)
           A[i][j] = 0;
   ```

2. (20%) Suppose that a disk drive has 1000 cylinders, numbered from 0 to 999. The drive is currently serving a request at cylinder 200, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is

   50, 500, 250, 800, 350, 550, 400, 600, 100.

   Starting from the current head position, what is the *total distance* (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk scheduling algorithms?

   (a) SSTF
   (b) LOOK
   (c) C-LOOK
   (d) SCAN
   (e) C-SCAN

3. (20%) Given an i-node with eight direct blocks and three levels of indirect blocks and assuming that the sizes of a pointer and a block are, respectively, 8 bytes and 8 Kbytes, answer the following questions. *(Hint: you may assume all the meta-information for a file has been read into the main memory and forget about the case where some buffers may need to be written back to disk first.)*

(a) What would be the size of the smallest file allowed in bytes?

(b) What would be the size of the largest file allowed in bytes?

(c) In the worst case, how many disk I/Os are required to read data from a block?

(d) In the worst case, how many disk I/Os are required to write data to a block?

4. (20%) A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last Reference	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

(a) Which page will NRU replace?

(b) Which page will FIFO replace?

(c) Which page will LRU replace?

(d) Which page will second chance replace?

5. (20%) Consider the following program that uses the POSIX threads API. What would be the output of the program?

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <sys/types.h>
#include <sys/wait.h>

int value = 1;

int main(int argc, char **argv)
{
    pid_t pid = fork();
    if (pid > 0) {
        printf("A = %d\n", ++value);
    }
    else if (pid == 0) {
        pid_t pid = fork();
        if (pid > 0) {
            waitpid(-1, NULL, 0);
            printf("B = %d\n", ++value);
        }
        else if (pid == 0) {
            pid_t pid = fork();
            if (pid > 0) {
                waitpid(-1, NULL, 0);
                printf("C = %d\n", ++value);
            }
            else if (pid == 0) {
                printf("D = %d\n", ++value);
            }
            else {
                exit(1);
            }
        }
        else {
            exit(1);
        }
    }
    else {
        exit(1);
    }

    return 0;
}
```