Department of Computer Science and Engineering
National Sun Yat-sen University

First Semester of 2004 PhD Qualifying Exam   Operating Systems

1. (15%) Consider the following program:

```
boolean blocked[2];
int turn;
void P(int id) {
    while (true) {
        blocked[id] = true;
        while (turn != id) {
            while (blocked[1-id]) {
                turn = id;
            }
        }
        /* critical section */
        blocked[id] = false;
        /* remainder */
    }
}

void main()
{
    blocked[0] = false;
    blocked[1] = false;
    turn = 0;
    parbegin(P(0), P(1)); /* begin P(0) and P(1) in parallel */
}
```

Find a counterexample that demonstrates that this solution is incorrect. That is, show an *execution sequence* of the two processes that either violates mutual exclusion or leads to deadlock. **For full credit, justify your answer.**

2. (10%) Consider the interprocess-communication scheme where mailboxes are used. Suppose a process P wants to wait for two messages, one from mailbox A and one from mailbox B. What sequence of `send` and `receive` should it execute so that the messages can be received in any order?

3. Consider a system with four processes P_1 through P_4 and five allocatable resources R_1 through R_5. The current allocation and maximum needs are as follows:

	$Allocated$	$Maximum$	$Available$
P_1	1 0 2 1 1	1 1 5 1 2	0 0 x 1 1
P_2	2 0 1 1 0	5 2 2 1 0	
P_3	1 1 0 1 0	3 1 3 1 0	
P_4	1 1 1 1 0	1 1 2 2 1	

Answer each of the following questions.

(a) (5%) What is the smallest value of x for which this is a safe state?

(b) (5%) What is the safe sequence?

4. Suppose that a scheduler has k ready processes at time 0, and that no new processes are created after time 0. Process i ($0 < i \leq k$) requires i units of computing time. Answer each of the following questions.

   (a) (5%) For a preemptive, round-robin scheduler with a scheduling quantum of one time unit, what is the mean turnaround time for these processes, assuming that process k is at the front of the ready queue and that other processes appear in decreasing order of required computing time?

   (b) (5%) For a non-preemptive, shortest-job-first scheduler, what is the mean turnaround time for these processes?

5. Assume a page reference string for a process with m frames (initially all empty). The page reference string has length p with n distinct page numbers occurring in it. For any page-replacement algorithms,

   (a) (5%) What is a lower bound on the number of page faults? **For full credit, justify your answer.**

   (b) (5%) What is an upper bound on the number of page faults? **For full credit, justify your answer.**

6. Consider a paging system with the page table stored in memory.

   (a) (5%) If a memory reference takes 100 nanoseconds, how long does a paged memory reference take? **For full credit, justify your answer.**

   (b) (5%) If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.) **For full credit, justify your answer.**

7. Consider the two-dimensional array `A`:

```
int A[][] = new int[100][100];
```

   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) (5%)

```
for (int i = 0; i < 100; i++)
    for (int j = 0; j < 100; j++)
        A[i][j] = 0;
```

   (b) (5%)

```
for (int j = 0; j < 100; j++)
    for (int i = 0; i < 100; i++)
        A[i][j] = 0;
```

8. Consider a file currently consisting of 200 blocks. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous allocation case, assume that there is no room to grow in the beginning, but there is room to grow in the end. Assume that the block information to be added is stored in memory.

   (a) (3%) The block is added at the beginning.

(b) (3%) The block is added in the middle.
(c) (3%) The block is added at the end.
(d) (3%) The block is removed from the beginning.
(e) (3%) The block is removed from the end.

9. 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) (2%) SSTF
   (b) (2%) SCAN
   (c) (2%) LOOK
   (d) (2%) C-SCAN
   (e) (2%) C-LOOK