CS 198:416: Operating Systems Design

Fall 1996

MIDTERM EXAM I

Prof. B. R. Badrinath

Instructions:

1. Put your name and student number on the exam books NOW!
2. The exam is closed book.
3. You have 80 minutes to complete the exam. **Be a smart exam taker**— if you get stuck on one problem go on to another problem. Also, don't waste your time giving irrelevant (or not requested) details or material.
4. Show all your work. Partial credit is possible for an answer, but only if you show the intermediate steps in obtaining the answer.
5. Good Luck!

---

1. "Quickies" (20 points, 15 minutes)

1. Differentiate between multiprogramming and timesharing.
2. What are the problems with busy waiting solutions to the mutual exclusion problem?
3. Differentiate between direct IPC and indirect IPC.
4. Name at least three steps required in effecting a context switch of a process.
5. Which of the following system calls when executed may result in a) Descriptor table full b) Process table full: 1) fork() 2) exec(), 3) open().
6. differentiate between pipe and port for IPC
7. What happens if semaphore wait queues are implemented as stacks instead of FIFO queues.

2. "Concurrent Processes" (20 points, 15 minutes)

1. (10 points) Give a possible output from the following C program. Is this the only output possible? Explain the output.

```
main()
{int i,rc;
for (i= 0; i<= 1; i++) {
   if ( (rc= fork() ) == 0) {
     printf(``Child %d executing\n'', i);
  } /* end if */
} /* end for */
printf(``All children created\n'');
}
```

2. (10 points) Write a C program using pipes that allows Process A to communicate (send a message to) with Process B and Process B to communicate (send a message to) with Process C. Pseudo code should suffice, but make sure that proper system calls are used.

2. "Process Synchronization" (60 points, 50 minutes)

1. (30 points, 20 minutes)

   You've just been hired by a Rutgers Fraternity club to help them out with synchronization problem of getting a boy and a girl to dance. The trick is to get one boy() to team up with another girl() to dance on a dance floor. The dance floor can hold atmost N people (N is a even number). When the boy or girl enters the hall, each will call one of two procedures, *boyarrive* or *girlarrive*, depending on the sex of the process. The procedure must wait until there is an available process of the opposite sex and the number of people already on the dance floor is $< N$, and then one of the procedures must call the procedure *StartDance* (which just prints out a message that a pair has started dancing) and then each process calls a procedure leave(). The procedure must wait until another process of the opposite sex also is ready to leave. Men and women may enter and leave the hall in any order. It doesn't matter which man is paired up with which woman (Rutgers students aren't very choosy), as long as each pair contains one boy and one girl. At any point in time there cannot be more than N people on the dance floor. Use **semaphores** to implement the various procedures.

2. (30 points, 30 minutes)

   A particular river crossing is shared by both cannibals (bad guys) and missionaries (good guys). A boat is used to cross the river, but it only seats three people, and must always carry a full load. In order to guarantee the safety of the missionaries, you cannot put one missionary and two cannibals in the same boat (because the cannibals can gang up and eat the missionary), but all other combinations are legal. When Missionary and Cannibal processes arrive at the river bank, they call missionary_arrive and cannibal_arrive respectively. The respective procedures return after the missionary or the cannibal cross the river. For each crossing, only one of the procedures should print a statement Now.Crossing.River.

You can assume an infinite supply of boats. Hence, all you need to ensure is that the proper set of people are in the boat. Solve this problem using either semaphores or monitors (condition variables). Clearly state whether you are solving the problem using semaphores or monitors.