[illegible]

Midterm

July 14, 2004 13.00-15.00

Good Luck!

Part A. Short Answer, answer the questions with one or two sentences (each 7 pt)

1 What are the two main purposes of an operating system?

2 What is multiprogramming?

3 What is deadlock?

Part B. Not Short Answer, you may answer the questions more than two sentences (each 9 pt)

4 An operating system runs in privileged mode, a hardware state where it has full access to machine resources. Why is such a mode needed, and why can t normal user processes and threads enter privileged mode?

5 What is a Critical Region? List and explain the four conditions that need to be satisfied to solve the critical-region problem?

6 What is meant by the term *context switch*? What might cause a context switch to occur?

7 Consider the following sets of processes, with the length of the CPU-burst time given in milliseconds. Arrival time is the time at which the process is added to the ready queue.

Process	**Burst Time**	**Arrival Time**
P1	20	0
P2	1	0
P3	6	0
P4	8	0
P5	4	8
P6	2	12

a Draw appropriate charts illustrating the execution of these processes using FCFS, SJF non-preemptive, and SJF preemptive.

b Calculate the wait times of each process for each strategy. Calculate the average wait times under each strategy.

Process	**FCFS**	**SJF-nonpre**	**SJF**
P1			
P2			
P3			
P4			
P5			
P6			
Average			

8 Describe the round-robin scheduling algorithm. Discuss what issues need to be consider to achieve good performance from this algorithm.

9 Describe four ways to *prevent* deadlock by attacking the conditions required for deadlock.

Part C.

10 i. Write a C program that creates processes and prints out whether child or parent process. (Hint: Use fork(), getpid().) (7 pts)

ii. Write a C program that creates threads and calculates factorial of N by using threads. (Hint: Use phtread_create(), pthread_join(), pthread_exit().) (8 pts)

11 The Readers and Writers Problem models access to a database with many competing processes wishing to read and write it. It is acceptable to have multiple processes reading the database at the same time, but if one process is updating (writing) the database, no other processes may have access to the database, not even readers. Write pseudo code to coordinate the readers and writers. Hints: Need a global counter, **rc**, for the number of processes reading or wanting to. Use 2 semaphores - one for controlling access to **rc**, one for controlling access to the database. (15 pts)