CMSC 412 Homework Three

Name: _____

Read Chapters 4 and 5.
Due Tuesday Sept 23.
Type, please.
As always, if you use sources on-line (e.g., wikipedia, pages found by google), please cite.

1. How many simultaneous pthreads can a single process run on linuxlab (heaving.csic.umd.edu)? Write and turn in the code, it's maybe 50 lines. `pthread_create` and `pthread_join` are needed. To compile, `gcc -Wall -g t.c -o t -lpthread.`

2. How long does it take to start, execute, and join that many threads if the thread body is empty? (i.e., the function returns without executing any logic.) Note: you may need to put some code in the thread body to ensure that it sticks around for question 1; be sure to remove it for this benchmark.

3. 4.10 Which of the following components of program state are shared acorss threads in a multithreaded process? (a) register values; (b) heap memory; (c) global variables; (d) stack memory.

4. 4.11 Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single-processor system? Explain.

5. 5.2 Explain the difference between preemtive and nonpreemptive scheduling.

6. 5.9 Why is it important tfor the scheduler to distinguish I/O-bound programs from CPU-bound programs?

7. 5.12 Consider the following set of processes:

Process	Burst time (ms)	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

   The processes are assumed to arrive in order $P_1, P_2, \ldots P_5$ at time 0.

   (a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemtpive priority (smaller priority number is higher) and RR (quantum = 1ms).

   (b) What is the turnaround time of each process for eachof the scheduling algorithms in part a?

   (c) What is the waiting time of each process for each of theses scheduling algorithms?

   (d) Which of the algorithms results in the minimum average waiting time (over all processes)?

8. 5.16 Consider a system implementing multilevel quueu scheduling. What strategy can a computer user employ to maximize the amount of CPU time allocated to the user's process.