CptS 464/564 Homework #3

Given: Tuesday, April 15, 2003

Due: Thursday, May 2, 2003 at the beginning of class (due to the need to get the papers from TC for grading, I will not accept late work for this assignment under any conditions!)

Max points: 60 points for 464, 80 points for 564.

Weight: 5% of final grade

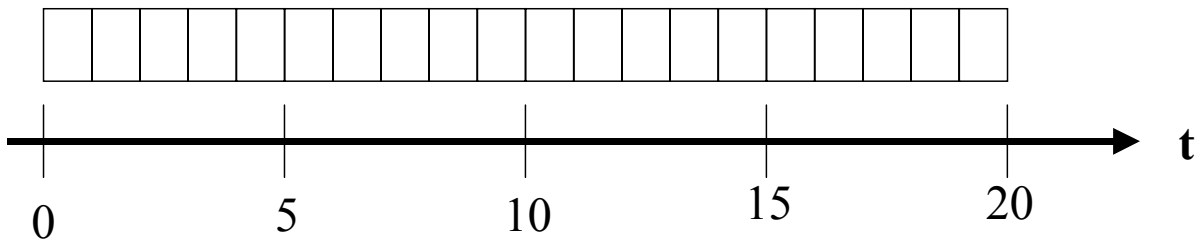
Note: This homework should not be handwritten, but must be done with a word processor.

1) [5 points] Rate Monotonic Scheduling

Fill in the schedule below with how a rate monotonic scheduler would allocate time slots to the following task table. Assume all tasks arrive at time 0.

Task	Period	T_{WCET}
1	2	1
2	10	2
3	20	6

Key: T1 T2 T3



2) [10 points] Security and Fault Tolerance

As mentioned in the text and in lecture, handling security is harder than fault tolerance, where the emphasis is generally on handling benign and accidental failures. The two reasons given were:

1. Direct causes of failures (attacks) are deliberate
2. Successful attacks often made possible by unintentional and dormant faults (vulnerabilities) made by designers

For each of these reasons, explain in a few brief but well-considered paragraphs why this makes handling security more difficult. To get full credit, an answer must show good insight.

3) [20 points 464; 30 points 564] Security Properties

The three main security properties discussed in class and the text were:

- **C**onfidentiality
- **I**ntegrity
- **A**vailability.

Devise an interesting and realistic distributed application scenario, using an existing application or making one up, where it goes through three plausible modes or phases of operation. In each phase it should have different CIA requirements that make sense, given the mode and application.

Write up a page or so total, explaining

- The overall application
- What each mode is, what its CIA requirements are for that mode, and, most importantly, why these requirements are appropriate for this mode. The requirements for a given mode and property (e.g., availability) should be from {High, Medium, Low}.

To get full credit, your writeup should have an innovative application (different from anyone else's) and should have cogent and compelling and interesting modes and reasons why the different CIA levels are required at each mode. Do your best writing: incorrect spelling and grammar will be a detriment to your grade.

564 students, do the same, but for two different applications (two separate pages).

4) (25 pts. 464; 35 pts. 564) Randomized algorithms

Consider a collection of 256 nodes, all except one initially holding the value 0. The exceptional node holds the value 1.

a) Suppose that the nodes individually execute the following protocol repeatedly:

> Choose a random node, call it r
>
> Set my value to max(my value, r's value)

Write a program that simulates this behavior for the 256 nodes. Do it as an ordinary sequential program. Simulate the behavior of all 256 nodes in each round before moving on to the next round. Results will be clearest if you compute *all* of the values of the $n+1^{st}$ round using the values of the n^{th} round, rather than modifying the data as you go.

At the end of each round record the number of nodes having value 1. Run the simulation until all the nodes have value 1. Graph the number of nodes holding value 1 versus the round number.

b) Suppose that the nodes individually execute the following protocol repeatedly:

> Choose a random node, call it r
>
> Set r's value to max(myvalue, r's value)

Modify your simulator so that nodes have this behavior. Again, run until all nodes have value 1 and graph the number of nodes holding value 1 versus the round number.

564 Only: Write two or three paragraphs comparing the results for a and b. Hint: consider especially the values for the first few rounds of part b and the last few rounds of part a. Try to explain what is happening.

c) Based on the above results, suggest a different behavior for the nodes that will more quickly distribute the 1 to all the nodes.

You may write the program in the language of your choice. (I suggest a scripting language such as python, perl, VB, etc. – go for maximum productivity on this one! My program was <80 lines of SML.) Turn in a program listing, your graphs and your explanations.