CS460 Assignment #6

Date assigned: 4/25/14
Date due: 5/5/14
Points: 35

This assignment is optional (not extra credit). If you have time and need the points, do it. If you do turn in this assignment it will be graded and count toward your final grade.

You are responsible for the reading and content of pages 323-330.

1) (5 pts) Consider the following resource information:

A total of 10 instances of a single resource (e.g. hard disk drives) exist and we know the following information:

Process	Currently Has	Maximum Needs
P0	3	9
P1	2	4
P2	2	7

a) Show that the system is currently in a "safe state?"

b) Suppose that P0 requests and gets another resource. Is the system currently in a "safe state?" If so, show how the system satisfies the safety condition. If not, show how the system does not satisfy the safety condition.

2) (5 pts) Consider RR scheduling with a quantum of 2.

Given the following processes with a length of the CPU burst in milliseconds:

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

a) Draw the Gantt chart to illustrate the execution of these processes.

b) What is the average waiting time? Show your work.

c) What is the average turnaround time? Show your work.

4) (25 pts) Write a C program that visualizes the Dining Philosophers problem. Your program is to implement the philosophers as concurrently running threads where thinking and eating are randomly generated as are the times for thinking and eating. Thinking and eating times must be in the less than 10 seconds range. Tackle the critical section issues using semaphores. Your implementation for this problem is to be in C on zeus. Here is sample output for the command:

```
 > ./CS460_Dine 5
```

```
.
.
Philosopher 4: Now I'm Thinking.
Philosopher 5: Now I'm Thinking.
Philosopher 4: Now I'm Hungry.
Philosopher 4: Grabbed  chopstick 4.
Philosopher 4: Waiting for chopstick 5.
Philosopher 4: Grabbed chopstick 5.
Philosopher 4: Now I'm Eating.
Philosopher 1: Now I'm Hungry.
Philosopher 1: Grabbed  chopstick 2.
Philosopher 1: Waiting for chopstick 1.
Philosopher 1: Grabbed chopstick 1.
Philosopher 1: Now I'm Eating.
Philosopher 3: Now I'm Hungry.
Philosopher 3: Waiting for chopstick 4.
Philosopher 4: Now I'm Thinking.
Philosopher 3: Grabbed  chopstick 4.
Philosopher 3: Waiting for chopstick 3.
Philosopher 3: Grabbed chopstick 3
.
.
```

Note: While eating, thinking, and their associated times are random, do not let a Philosopher starve. Therefore, implement some kind of aging solution after a certain period of time to keep a philosopher from starving.

On the day this assignment is due, you are to turn in a typed copy containing your answers to the above questions in order.

Execution Format:

```
./CS460_Dine NumberOfPhilosophers
```

Makefile Targets:

```
CS460_Dine: build the executable CS460_Dine at the root of the project

clean:

valgrind: Run the following:
valgrind -v --leak-check=yes ./CS460_Dine 5
```

Testing:

I will test your program on zeus.

Notes:

- Your project needs to be well documented and broken into well-defined functions.
- Minimize global variables!
- **Remember all of your solutions are to be original and in your own words.**

THERE IS NO LATE GRACE PERIOD OF TIME FOR THIS LAST SOLUTION!!!!