Systems and Networks	**Homework 5: Deadlocks/Memory**
CS3600 Fall 2014	**July 28, 2014**

This homework is due at the beginning of class on October 22, 2014 and is worth 1.5% of your grade.

Name: _____

CCIS Username: _____

Problem	Possible	Score
1	15	
2	15	
3	25	
4	20	
5	25	
Total	100	

1. Suppose we have the following set of processes, and we have a computer with 11 instances of resource type A. Also suppose that each process has specified, up front, the maximum number of each resource type it will ever request at once (Requested). We also know how many instances of each resource type each process is currently using (Used).

	Resource type A	
Process	**Requested**	**Used**
P_1	10	6
P_2	3	1
P_3	8	1

1a. Is is the system in a safe state? If so, why, and if not, why not? (5 pts)

1b. Suppose that P_2 requests the use of another instance. Would the be in a safe state? If so, why, and if not, why not? (5 pts)

1c. Suppose that, instead, P_3 requests the use of another instance. Would the be in a safe state? If so, why, and if not, why not? (5 pts)

2a. Why are page sizes typically implemented as powers of 2? (10 pts)

2b. Does it make sense for a paging system to use multiple page sizes? What would be the advantages and disadvantages of such a scheme? You must state at least one advantage and disadvantage. (5 pts)

3a. Explain the tradeoffs in choosing a page size in a paged main memory (i.e., non-virtual) management system. You must state at least three different concerns. (10 pts)

3b. In a pure paging system, is internal fragmentation possible? What about external fragmentation? You must explain each of your answers. (5 pts)

3c. What is the effect of allowing entries in different page tables to point to the same page frame in memory? Explain how this effect could be used to decrease the amount of memory needed to create multiple processes that run the same executable program or use the same libraries. What would be the effect if a process modified a page "shared" in this way? Can you think of a way to overcome this problem? (10 pts)

4a. Consider a 32-bit system running with a three-level page table (e.g., a 2nd outer page number, and outer page number, and an inner page number). Suppose the memory page size is 1 KB. How many bits are in the page offset? (5 pts)

4b. Let us suppose that we want our outer pages and inner pages to each hold 256 entries. How big (in bytes) will each the outer page and inner pages be? (5 pts)

4c. In this scenario, how many entries will our 2nd outer page table have? How big will it be in memory? (5 pts)

4d. In the worst case, in this scenario, how many memory loads may be necessary to look up a memory address? (5 pts)

5a. Consider a paging system with the page table stored in memory with no TLB. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take? (5 pts)

5b. Suppose we add a TLB, and 75% of all page-table references are found in the TLB. What is the effective memory reference time? Assume that a lookup in the associative registers takes 40 nanoseconds. (10 pts)

5c. Suppose you had the choice between increasing the size of the TLB (thereby increasing the hit rate to 85%) or increasing the lookup speed of the TLB to 25 nanoseconds. Which would you pick, and why? (10 pts)