Answer 100 points worth of questions; if you answer more than that, I will throw away scale down your worst answers.

**Problem 1:** (15 pts)
In your opinion, what are the three most important concepts you learned in this class. Justify your choices (explain why you think they are important). This problem will be graded based on both the importance and explanation of your choices.

**Problem 2:** (30 pts)
Provide short explanations to any 10 of the following questions
1. What is the difference between exceptions and interrupts? Give an example of each.
2. What are the two flavors of monitor semantics we discussed? Is there a difference to the programmer?
3. What is the difference between a working set and a resident set?
4. Priority Inversion is a phenomenon where a low priority process prevents a high priority process from running. Explain how this could happen.
5. In a uniprocessor, do you need an atomic instruction in the hardware in order to implement semaphores?
6. Is each of the following associated with a process or a thread? page table, stack, global variables, program counter.
7. LRU is suitable for disk cache replacement but not for page replacement. Explain.
8. Buffer memory should be pinned all the time for all possible OS I/O configurations; true or false? Explain.
9. A new magnetic disk device with two independently controllable read/write heads has been introduced. Suggest a disk scheduling algorithm for it. What if only one head can write?
10. Given a RAID disk array, are there additional considerations when you are defragmenting?
11. Can two concurrent activities that access the same critical region but do not synchronize deadlock?

**Problem 3:** (5 pts) Evaluate the following scheduling algorithm. Start with multiple level feedback. However, the time quanta for each successive queue is reduced.

**Problem 4:** (10 pts) A new memory/storage device has been invented that offers stable storage at the speed and performance levels of memory. So, there is no need for a separate disk and memory, all storage can be implemented using this new device. Is virtual memory needed? Should memory management and file/disk management be kept separate? if not suggest an integrated “storage management” approach.

**Problem 5:** (10 points) Consider a byte-addressable computer that supports segmentation. There are 26 bits in a virtual address, of which 5 are the segment number. The computer can access at most 1 GB of physical memory. Answer the following and explain your reasoning. (a) How many bits in a physical address? (b) How many entries in the segment table? (c) What is the maximum size of a segment? (d) How many bits in a base register? (e) How many bits in a bound (segment length) register? (f) How many processes will fit into the computer’s memory simultaneously, if each process is using all of its segments and each segment is its maximum size? (g) Suppose you decide to increase the maximum segment size. How big can you make it for this computer?

**Problem 6:** (10 pts)
- (a) In file allocation techniques, we discussed the unix I-node scheme. If you recall, in this scheme, the I-node for a given file contains a small number of indexed block entries (so far, identical to indexed allocation), a number of two level entries (each entry pointing to another index block), and a number of three level entries. This is analogous to the multi-level page table scheme used in memory management. Does it make sense to use an inverted page-table like approach for file allocation tables? Explain.
• (b) Free list management can be done either using a bitmap or an allocation table of the free blocks/frames. Discuss tradeoffs between the two approaches. Consider space and efficiency; consider memory and disk management.

**Problem 7: (10 pts)**
This problem refers to the Figure of Manhattan traffic above.
(a) Show that the four ingredients for deadlock exist in this situation.
(b) For three of the deadlock management techniques discussed in class, what does the policy translate to in terms of this problem? Which of these techniques are practical (can be done)?
(c) You have just been elected mayor of new york. Unfortunately, your term might be cut short because drivers in Manhattan are not happy with your traffic management policies (too many red lights!) Drawing on your OS experience, you realize that you can avoid deadlock by rearranging the traffic pattern. Demonstrate this (note that the solution is not acceptable if you make traffic flow in a single direction).

**Problem 8: (15 pts)**
You are designing an airline reservation system. A centralized database is concurrently checked and updated by all authorized travel agents. Agents check planes for vacancies and reserve empty seats if there are any.
(a) Describe some wrong behavior that may occur if concurrency is not managed in this system. Is disabling interrupts while an agent accesses the database a valid solution for this situation?
(b) Mutual exclusion can be enforced at the database level, at the flight level, or at the individual seat level; what is the tradeoff involved?
(c) The CPU for the system that will run your airline software has a test-and-increment atomic instruction that can be used to build locks. Implement an the acquire lock and release lock procedures. Make sure that you correctly handle overflow.

**Problem 9: (20 pts)**
A monkey community lives on an island that has only one coconut tree. The monkeys take turn going up the tree (which can hold only 3 monkeys at a time). After each monkey grabs a coconut, it climbs down the tree to go and eat it. Monkeys can be climbing up, or down the tree, but never both ways.
(a) Build the procedure implementing this synchronization problem; each monkey is an independent thread.
(b) The dominant monkey does not like to wait. When it arrives, it climbs up the tree ahead of any monkeys that got there before it (there still can only be three monkeys up the tree). Implement the synchronization for this problem (show the procedure for the dominant monkey thread and extend the procedure for the regular monkeys). Is starvation possible?