1. (30 Points) True/False Problems

Answer the following true/false (T/F) questions. Each T/F problem is worth 2 points. Each missing or wrong answer costs -2 point.

(a) Without "free" (deallocate) it is easy to write a "malloc" implementation that never fragments memory.

(b) DMA allows an I/O device to transfer data to and from memory without involving the CPU in the transfer.

(c) A context switch from one process to another can be accomplished without executing OS code in kernel mode.

(d) The dispatcher is responsible for setting thread priorities.

(e) Virtual addresses must be same size as physical addresses.

(f) Page offsets in virtual addresses must be the same size as page offsets in physical addresses.

(g) Monitors are more powerful than Semaphores because Monitors can implement solutions to synchronization problems that Semaphores cannot solve.

(h) A user-level process cannot modify its own page table entries.

(i) Shortest Remaining Time First is the best preemptive scheduling algorithm that can be implemented in an Operating System.

(j) Several threads can share the same address space.

(k) Paging leads to external fragmentation

(l) FIFO scheduling policy achieves lowest average response time for equal size jobs

(m) When designing a multithreaded application, you must use synchronization primitives to make sure that the threads do not overwrite each other's registers.

(n) A CPU scheduling algorithm cannot provide both fairness and minimum average response time.

(o) A correct application using the Banker's algorithm for all requests will never deadlock.

2. (12 points) Short Questions

(a) Is address translation (virtual memory) useful even if the total size of virtual memory as summed over all possible running programs is guaranteed to be smaller than a machine's total physical memory? Why? (3 points)

(b) What is the name of the following code: (2 points)
- Operating system code executed when an asynchronous device signals the CPU

(c) What is a TLB and what does it do? (3 points)

(d) What is an inverted page table? What are its benefits and disadvantages? (4 points)

3. (10 points) Most operating systems are designed for general-purpose computation. A proposal has been made for an OS that is optimized for running math-intensive programs. In MathOS, the kernel includes system calls for many useful mathematical operations, such as matrix arithmetic, Bessel functions, Euclidean distance, etc. These system calls are written in highly optimized assembly language for maximum performance. Is this concept for MathOS a good idea? Explain why or why not.

4. (10 points) Consider the following snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	

P3	0 6 3 2	0 6 5 2
P4	0 0 1 4	0 6 5 6

Answer the following questions using the banker's algorithm:

(a) Is the system in a safe state? Why?

(b) If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately?

5. (20 points) One day famed student Surfer Kim had an inspiration while hanging ten at Dck Pond (오리 연못). He observes that most programs have most of their data at the beginning of the address space. For his homegrown GeekOS, he decides that he is going to implement his page tables similar to the way Unix implements inodes. He calls this page table design *Inode Page Tables*. Inode Page Tables are essentially two-level page tables with the following twist: The first half of the page table entries in the master page table directly map physical pages, and the second half of the entries map to secondary page tables as normal. Call the first half the entries *fast*, and the second half *normal*.

   For the following questions, assume that addresses are 32 bits, the page size is 4 KB, and that the master and secondary page tables fit into a single page.

   (1) How many virtual pages are fast pages? (5 points)

   (2) How many virtual pages are normal pages? (5 points)

   (3) What is the maximum size of an address space in bytes (use exponential notation for convenience, e.g., 2^3)? (5 points)

   (4) Inode Page Tables reduce the lookup time for fast pages by one memory read operation. Do you think that this is an effective optimization? Briefly explain. (5 points)

6. (18 points) In the operating systems, there are several recurring themes (i.e., ideas that applied quite broadly across several topics).

   (1) One of these themes was **virtualization**. Identify **three** contexts in which virtualization was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of the virtualization approach to the problem. (6 points)

(2) A second theme was **hardware support**. Identify **three** contexts in which hardware support was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of a hardware-based approach to the problem. (6 points)

(3) A third theme was **caching**. Identify **three** contexts in which caching was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of caching as a solution. (6 points)