Final Exam – Graduate Operating Systems
CSE 60641 – Fall 2018

NAME: _____

Reminders

1. Make sure to look through entire exam from front to back.
2. Please be concise in your answers if possible. A dissertation response is not needed for most questions.
3. Illustrations can be helpful to convey your point.
4. When in doubt, ask a question.
5. Try to at least have a partial answer for each exam item.
6. Many answers depend on context. Clearly define the context if necessary for your answer.
7. The exam is closed book, closed note, closed computer, closed electronic device!
8. Write your answers legibly.

Question 1: Thrashing occurs when more memory is being actively utilized than the system contains. When talking about thrashing, one often refers to the **working set** of a process.

a.) Define the **working set** of a process.

b.) If a system is thrashing, how can we try to reduce thrashing within the OS (i.e., how would we change the OS)?

Question 2: Consider the following code for inserting into a hash table that will be concurrently used by multiple threads. Some details of the code are omitted for clarity. The hash table is implemented as an array of N linked lists, and each node in these linked lists contains an integer hash key (key) and its associated object (obj).

```
void hashInsert(int key, void *obj) {

        int listNum = key % N; // Determine which list the item belongs to

        listInsert(hashArray[listNum], key, obj);

}

void listInsert(node *head, int key, void *obj) {

        // Create a new node (nn) that contains key and obj

        nn->next = head;

        head = nn;

}
```

a.) Show a problem that can occur if two threads try to insert items into the hash table simultaneously.

b.) Use a single semaphore to fix the problem (you can put the required code for the semaphore directly into the code above).

Question 3: Consider a system with four processes P1..P4 with the following address space sizes: 64KB for P1, 44KB for P2, 16KB for P3, and 60KB for P4. The page size is 4KB. Each process has a working set that is 1/2 of its address space. How many frames should the system provide to these four processes (each) to avoid thrashing? Clearly show each step of your analysis.

Question 4: The second chance algorithm is an approximation of LRU based on using one "use bit" per page (when a page is used/accessed, the "use bit" is set to 1). The algorithm also uses a pointer to the next potential victim, which goes through the frames in a round robin fashion whenever a victim frame is needed. The table below shows a sequence of page accesses and four (initially empty) frames F0-F3. Assume that the victim pointer points initially to F0. After each page request, show the contents of the four frames in the table. When a page fault occurs, clearly mark (e.g., using a circle) which frame is the victim. Hint: clearly mark for each entry if the use bit is set or not (e.g., by making a dot in the corresponding cell in the table) and indicate which frame the pointer is pointing to after each round. Also assume that when there are empty frames, pages are loaded into the lowest available frame.

Page	**0**	**1**	**3**	**6**	**2**	**4**	**5**	**2**	**5**	**0**	**3**	**4**	**2**	**0**	**5**
F0															
F1															
F2															
F3															

Question 5: The TLB can either be dedicated to the currently running process only or it can be shared among all processes in a system.

a.) Name an advantage of a dedicated TLB.

b.) Name an advantage of a shared TLB.

Question 6: You are using a buddy algorithm that allocates storage from 16-byte blocks up to 1024-byte blocks. What percentage (approximately) of allocated memory is wasted due to internal fragmentation when satisfying requests for allocating 130-byte chunks? Justify your answer.

Question 7:

a.) What kind of fragmentation do we have with fixed-sized partitions? Given N processes each in size M memory partitions, how much fragmentation do we have?

b.) What kind of fragmentation do we have with variable-sized partitions? Explain this type of fragmentation and propose a technique/algorithm that can reduce such fragmentation.

Question 8:

a.) What does it mean for an architecture to be virtualizable? (Hint: explain this using the terms "sensitive" and "privileged")

b.) Clearly explain the difference between a "host OS" and a "guest OS" in a virtual machine system.

c.) How does paravirtualization fix the problem of the non-virtualizability of x86 computers?

Question 9: For a periodic real-time task set consisting of tasks T1(2,10), T2(2,8), T3(1,5), and T4(1,5), where the first number in parenthesis is the worst-case execution time and the second number is the period. Assume that the deadline is the end of the period. Answer the following questions:

a.) What is the total CPU utilization?

b.) In a system that uses EDF as the scheduling policy, is this task set schedulable? (Justify your answer!)

c.) In a system that uses rate-monotonic scheduling, can we guarantee that the tasks will always meet their deadlines? (Justify your answer!)

d.) In a system that uses rate-monotonic scheduling, is task T1 or task T3 more likely to miss deadlines? (Justify your answer)

Question 10 (Potpourri #1): Clearly mark the correct answer for each question below (only one correct answer per question!).

a.) Consider a machine that requires all addresses be mapped through its 128-entry TLB. What is the maximum amount of memory that can be accessed without incurring any TLB faults, if the page size is 4KB?

a. 128KB
b. 128MB
c. 512KB
d. 512MB
e. 4GB

b.) Monitoring page fault frequency of a process allows us to:

a. Manage page frame allocation per process.
b. Adjust the size of the TLB for optimum performance.
c. Determine if the process is I/O bound or CPU intensive.
d. Identify the number of illegal instructions and invalid memory accesses within a program.

c.) The Berkeley Fast File System did NOT improve the Unix File System by adding:

a. Cylinder groups.
b. Bitmap allocation for keeping track of free and used blocks.
c. Extent-based allocation.
d. Prefetching of blocks.

d.) The copy-on-write mechanism provides:

a. An efficient way to create new processes.
b. A clever way to share virtual memory pages (at least temporarily).
c. A way to avoid unnecessary page copying.
d. All of the above.
e. None of the above.

e.) What is a hint that a page might be good to swap out?

a. It hasn't been used for a while.
b. It is currently loaded into a core's TLB.
c. It belongs to an operating system process.
d. It is shared by multiple processes.

f.) Which cannot be a valid page size?

a. 32 bytes
b. 1024 bytes
c. 3072 bytes
d. 1,048,576

g.) Which of the following is NOT a useful approach to improving system performance?

a. Carefully choosing an appropriate benchmark.
b. Improving the parts of your code that you just know are slow.
c. Analyzing data from experiments to identify bottlenecks.
d. Developing a new simulator to improve reproducibility.

h.) Which of the following is NOT a reason that virtualization became popular?

a. Difficulty migrating software setups from one machine to another.
b. Useful hardware virtualization features.
c. Ability to re-provision hardware resources as needed.
d. Lack of true application isolation provided by traditional operating systems.

i.) Which of the following makes it easier to virtualize the x86 architecture?

a. Hardware page tables.
b. Instructions that are not classically virtualizable.
c. Multiple privilege levels.
d. Real-time scheduling algorithms.

j.) Which of the memory allocation schemes are subject to internal fragmentation?

a. Multiple Contiguous Fixed Partitions.
b. Segmentation.
c. Multiple Contiguous Variable Partitions.
d. None of the above.

Question 11 (Potpourri #2):

___T ___F A clock page replacement algorithm tries to approximate a Least Recently Used (LRU) algorithm.

___T ___F Using a multi-level page table will increase the TLB hit time.

___T ___F The main reason to have a TLB is to speed up address translation.

___T ___F Page offsets in virtual addresses must be the same size as page offsets in physical addresses.

___T ___F For every new level in a multi-level page table, an additional memory reference will be needed.

___T ___F Inverted page tables are smaller than regular page tables.

___T ___F The working set of a process is statically determined using compile-time analysis.

___T ___F The goal of paravirtualization (e.g., Xen) is to make the guest operating system completely unaware that it is running on a virtual machine.

___T ___F Priority inheritance is a scheme to circumvent priority inversion where the higher priority process temporarily assumes a lower priority.

___T ___F Two processes reading from the same physical address will access the same contents.

___T ___F TLBs are more beneficial with multi-level page tables than with linear (single-level) page tables.

___T ___F Virtual machine monitors that use dynamic binary translation typically have better performance than virtual machine monitors that use hardware support for virtualization.

___T ___F The balloon process in VMware's ESX server allows the guest OS to request additional memory from the host OS.

___T ___F The kernel on a multiprocessor can use the local disabling of interrupts (within one CPU) to ensure that only one process enters a critical section.

___T ___F A thread can be blocked on multiple condition variables simultaneously.

___T ___F Rate-monotonic scheduling uses the worst-case execution time of a process to determine the process' priority.

___T ___F A tradeoff in using DVFS in real-time systems is that schedulability cannot be guaranteed anymore.

___T ___F The trap & emulate approach can only be used if a system is fully virtualizable.

___T ___F Scheduler activations provide a communication mechanism between different guest operating systems in a hosted VM environment.

___T ___F In a real-time system, the quantum given to a process by the round-robin scheduler corresponds to the process' priority level.

___T ___F The reason for using a multilevel page table is to make it easier to find unused page frames in the system.

___T ___F “Dirty bit” and “reference bit” are different names for the same thing.

___T ___F In virtual memory systems, evicting a dirty page is faster than evicting a clean page.

___T ___F A TLB caches translations from full virtual addresses to full physical addresses.