CSci 402 - Operating Systems

Alternate Final Exam

Spring 2021

(7:00:00pm - 7:40:00pm, Friday, May 7)

Instructor: Bill Cheng

Teaching Assistant: Shashank Saurabh

(This exam is open book and open notes.
Remember what you have promised when you signed your
Academic Integrity Honor Code Pledge.)

Time: 40 minutes

_____-
Name (please print)

Total: 38 points

_____-
Signature

Instructions

1. This is the first page of your exam. The previous page is a title page and does not have a page number. Since this is a take-home exam, no need to sign above since you won't submit this file.
2. Read problem descriptions carefully. You may not receive any credit if you answer the wrong question. Furthermore, if a problem says *"in N words or less"*, use that as a hint that N words or less are expected in the answer (your answer can be longer if you want). Please note that points may get *deducted* if you put in wrong stuff in your answer.
3. If a question doesn't say `weenix`, please do not give `weenix`-specific answers.
4. Write answers to all problems in the **answers text file**.
5. For non-multiple-choice and non-fill-in-the blank questions, please show all work (if applicable and appropriate). If you cannot finish a problem, your written work may help us to give you partial credit. We may not give full credit for answers only (i.e., for answers that do not show any work). Grading can only be based on what you wrote and cannot be based on what's on your mind when you wrote your answers.
6. Please do *not* just draw pictures to answer questions (unless you are specifically asked to draw pictures). Pictures will not be considered for grading unless they are clearly explained with words, equations, and/or formulas. It's very difficult to draw pictures in a text file and you are not permitted to submit additional files other than the answers text file.
7. For problems that have multiple parts, please clearly *label* which part you are providing answers for.
8. Please ignore minor spelling and grammatical errors. They do not make an answer invalid or incorrect.
9. During the exam, please only ask questions to *clarify* problems. Questions such as "would it be okay if I answer it this way" will not be answered (unless it can be answered to the whole class). Also, you are suppose to know the definitions and abbreviations/acronyms of *all technical terms*. We cannot "clarify" them for you. We also will **not** answer any clarification-type question for multiple choice problems since that would often give answers away.
10. Unless otherwise specified and stated explicitly, multiple choice questions have one or more correct answers. You will get points for selecting correct ones and you will lose points for selecting wrong ones.
11. When we grade your exam, we must assume that you wrote what you meant and you meant what you wrote. So, please write your answers accordingly.

(Q1) (2 points) What are the **OS design approaches** to fix the **reliability** problem of a **monolithic kernel** to reduce kernel crashes?

(1) run the entire system (kernel+applications) inside a virtual machine
(2) shrink the size of the kernel code that must run in the privileged mode
(3) never release a kernel unless it's completely bug-free
(4) use dynamically loaded kernel modules so that unreliable kernel module can be unloaded while the rest of the kernel is running
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q2) (2 points) Which of the following statements are correct about the **inverted page table** scheme?

(1) the main advantage of using a inverted page table scheme is its lookup speed
(2) in an inverted page table scheme, all processes can share one inverted page table
(3) in an inverted page table scheme, a hash function is used, whose inputs are thread ID and virtual page number
(4) in an inverted page table scheme, the page table is said to be "indexed by physical page number"
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q3) (3 points) Let's say that you have four threads A, B, C, and D and you are using the basic **round-robin (RR) / time-slicing** scheduler with a very small time slice. At time zero, all four threads are in the run queue and their processing times are shown in the table below. Assuming that there are no future arrivals into the run queue, please complete the table below with the "waiting time" of all four threads and the "average waiting time" (AWT) of these four threads and write the results on your answer sheet. Please make it very clear which waiting time is for which thread and which one is the AWT. For non-integer answers, you can use fractions or decimals with two digits after the decimal point. Your answer must not contain plus or multiplication symbols. You must use the definition of "waiting time" given in lectures.

	A	B	C	D	AWT
T (hrs)	5	7	2	5	-
wt (hrs)					

(Q4) (3 points) Let's say that you have four threads A, B, C, and D and you are using **stride scheduling**. You have decided to give thread A 4 ticket, thread B 2 tickets, thread C 3 tickets, and thread D 4 tickets. The initial pass values that **you must used** for the four threads are shown below along with the "winner" of the iteration 1. Please run **stride scheduling** to fill out all the entries (pass values) in the table and keep track of the "winner" in each round. For **iterations 2 through 7**, please write on your answer sheet the "winner" and the winning pass value of that iteration. (For example, you would write "A:5" for iteration 1 since A is the "winner" of iteration 1 and the winning pass value is 5.) You must use the **smallest possible integer stride values** when calculating all the pass values. If you get the stride values wrong, you will not get any partial credit for this problem.

itr	A	B	C	D
1	(5)	7	10	11
2				
3				
4				
5				
6				
7				

(Q5) (2 points) Which of the following statements are correct about using the **multi-level feedback queue** to schedule both interactive and non-interactive jobs?

(1) if a thread gives up the CPU voluntarily before using up a full time slice, you should decrease its priority if you can
(2) if a thread uses a full time slice, you should increase its priority if you can
(3) if a thread gives up the CPU voluntarily before using up a full time slice, you should increase its priority if you can
(4) if a thread uses a full time slice, you should decrease its priority if you can
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q6) (2 points) Which of the following statements are correct about **paravirtualization**?

(1) VMware is well-known for its paravirtualization patent
(2) inside a paravirtualized OS/kernel, usually there are no device drivers
(3) a paravirtualized OS/kernel cannot run on the hardware that the real OS was designed to run on
(4) one way to implement paravirtualization is to to fix the hardware so that an OS/kernel can run inside a virtual machine without modification
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q7) (2 points) Which of the following statements are correct about **vfork()**?

(1) **vfork()** is usually faster than **fork()**
(2) if **vfork()** is used, the parent process and the child process cannot run "concurrently"
(3) **vfork()** is best used by expert system programmers because it's tricky to use it correctly
(4) **fork()** and **vfork()** are equivalent
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q8) (2 points) For a terminal device, it is possible that characters can arrive from the keyboard even though there isn't a waiting read request from an application. It is also possible that characters generation are too fast for the terminal to display characters. What **kernel data structures** are used to **solve these two problems**?

(1) a memoroy management unit (MMU)
(2) a B+ tree and hash map
(3) a translation lookaside buffer
(4) a pair of input and output buffers
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q9) (2 points) Which of the following statements are correct about **extents** used in file systems in Windows?

(1) it's a good idea to "defrag" the disk to improve performance even for Linux systems that do not use extents
(2) extents has both external fragmentation and internal fragmentation
(3) extents is optimized for random access
(4) NTFS uses disk maps instead of extents to improve performance over FAT32
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q10) (2 points) Which of the following statements are correct about the **NOR** vs. **NAND** flash memory technologies?

(1) for writing, a NAND flash is not page-erasable but it is block-erasable
(2) for a NAND flash, the smallest addressable unit for reading is a block
(3) a NOR flash is byte-addressable
(4) a NAND flash is byte-addressable
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q11) (2 points) Which of the following statements are **correct** about **executing sensitive instructions** in an IBM 360 **virtual machine**?

(1) when a sensitive instruction is executed in the **VMM**, it should causes an additional trap into the VMM itself
(2) when a sensitive instruction is executed in the **privileged mode** of the **real machine**, it should cause a trap into the VMM
(3) when a sensitive instruction is executed in the **virtual privileged mode** (inside a **virtual machine**), it should cause a trap into the VMM
(4) when a sensitive instruction is executed in the **virtual user mode** of the **virtual machine**, it should cause a trap into the VMM
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q12) (2 points) Which of the following statements are correct about using **base and bounds** registers in a **segmented virtual memory** scheme?

(1) the bounds register contains a virtual address
(2) the base register contains a physical address
(3) the bounds register contains a physical address
(4) the base register contains a virtual address
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q13) (2 points) A correct implementation of **straight-threads** (i.e., no interrupt) **thread switching** on a **single CPU** is shown here (assuming that the run queue is never empty):

```
void thread_switch() {
  thread_t NextThread, OldCurrent;

  NextThread = dequeue(RunQueue);
  OldCurrent = CurrentThread;
  CurrentThread = NextThread;
  swapcontext(&OldCurrent->context, &NextThread->context);
}
```

Which of the following statements are correct about using the above code in a **multiple-CPU** system?

(1) cannot use the code as-is because accessing the **RunQueue** needs to be synchronized across multiple CPUs if a single **RunQueue** is used
(2) cannot use the code as-is because **CurrentThread** needs to be an array since we have multiple CPUs
(3) cannot use the code as-is because multiple CPUs must have multiple **RunQueues**
(4) cannot use the code as-is because **swapcontext()** must include an argument to specify which CPU to use for context swapping
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q14) (2 points) Which of the following statements are correct about **terminal device drivers** vs. **network device drivers**?

(1) one major difference between the two is the data rate they must deal with
(2) in network communication, data needs to be passed from one kernel module to another without copying to achieve acceptable performance
(3) network drivers deals with binary data while terminal drivers only deals with non-binary data
(4) most of the work performed by a terminal device driver is done as "deferred processing"
(5) their abstractions to applications are completely different

Answer (just give numbers): _____

(Q15) (2 points) Which of the following statements are correct about thread implementation strategies?

(1) one problem with the M x N (two-level) model is priority inversion
(2) the scheduler activations model is a variation on the one-level model
(3) one problem with the N x 1 (two-level) model is priority inversion
(4) one problem with the 1 x 1 (one-level) model is that it's slow
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q16) (2 points) Which of the following statements are correct about using the **aging** mechanism for a **multi-level feedback queue**?

(1) if aging is used, the scheduler may decrease a thread's priority if the thread has not run for a long time
(2) the main goal of aging is to increase throughput for this scheduler
(3) aging is not necessary because threads can never starve if you use this scheduler
(4) aging can be used to address the thread starvation problem for this scheduler
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q17) (2 points) The following code can be used to lock a mutex in a **straight-threads** (i.e., no interrupt) mutex implementation on a **single CPU**:

```
if (!m->locked) {
  m->locked = 1;
}
```

Which of the following statements are correct about using the above code in a **multiple CPU** system?

(1) cannot use the code as-is because reading the value of m->locked is not an atomic operation
(2) cannot use the code as-is because setting the value of m->locked is not an atomic operation
(3) cannot use the code as-is because multiple threads may all think that they own the same mutex
(4) cannot use the code as-is because multiple CPUs must not be allowed to read from the same memory location at the same time
(5) cannot use the code as-is because multiple CPUs must not be allowed to write to the same memory location at the same time

Answer (just give numbers): _____

(Q18) (2 points) A correct implementation of **straight-threads** (i.e., no interrupt) **synchronization** on a single CPU is shown here:

```
void mutex_lock(mutex_t *m)
{
  if (m->locked) {
    enqueue(m->queue,
        CurrentThread);
    thread_switch();
  } else
    m->locked = 1;
}
```

```
void mutex_unlock(mutex_t *m)
{
  if (queue_empty(m->queue))
    m->locked = 0;
  else
    enqueue(runqueue,
        dequeue(m->queue));
}
```

Let's say that thread X owns mutex **m**. Which of the following statements are correct about what would happen if another thread (thread Y) calls **mutex_lock(m)**?

(1) thread X will enter **mutex_unlock()** immediately and thread Y will return from mutex_lock() immediately with the mutex locked
(2) thread X will enter **thread_switch()** and thread Y will return from **thread_switch()** immediately
(3) thread Y will enter **thread_switch()** and thread X will return from **thread_switch()** immediately
(4) thread Y will go to sleep in **m**'s mutex queue
(5) thread Y will go to sleep in the run queue

Answer (just give numbers): _____