CSci 402 - Operating Systems

Final Exam

Summer 2022

[9:30:00am-10:10:00am), Tuesday, August 2

Instructor: Bill Cheng

Teaching Assistant: Zhuojin Li / Bowen Song

(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) Which of the following statements are correct about approaches to deal with the problem caused by the **popf** instruction so that a virtual machine can be built for **x86 processors**?

(1) VMware's solution is a compile-time solution, i.e., some sensitive instructions are replaced with hypercalls when kernel is compiled
(2) in Intel's solution, the hypervisor runs in "ring 0" while a guest OS runs in "ring 1"
(3) with paravirtualization, certain sensitive instructions are replaced with hypercalls at the time the kernel is compiled
(4) in Intel's solution, the `popf` instruction is disabled so that it won't cause any problem
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q2) (2 points) The first procedure of an **idle thread** is shown here:

```
void idle_thread() {
  while (1) {
    euqueue(runqueue, CurrentThread);
    thread_switch();
  }
}
```

Which of the following statements are correct about such an **idle thread**?

(1) an idle thread can never sleep in a mutex queue or an I/O queue
(2) an idle thread is often used in user space when there are multiple CPUs
(3) an idle thread is a kernel thread that never gives up the CPU
(4) an idle thread does not need a thread control block because it never needs to wait for mutex or I/O
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q3) (2 points) Let's say that you are using a **rate-monotonic scheduler** to schedule 4 periodic tasks with $T_1 = 0.5$, $P_1 = 3.5$, $T_2 = 1$, $P_2 = 6$, $T_3 = 0.5$, $P_3 = 5.5$, and $T_4 = 1$, $P_4 = 6.5$. Let's say that you schedule all 4 periodic tasks to start a time = 0. Since the total utilization is too large to guarantee that all jobs will meet their deadlines, the only way to know is to simulate the **rate-monotonic scheduler**. How many seconds into the simulation would be the first time all 4 jobs would start executing at exactly the same time again? Please just give a numeric answer (no partial credit for this problem).

(Q4) (2 points) Which of the following statements are correct about using **base and bounds** registers in a **segmented virtual memory** scheme?

(1) 4 is not a valid value for a bounds register because it's too small
(2) the base register contains a virtual address
(3) the bounds register contains a "validity" bit to indicate if the value stored in the base register is valid or not
(4) the base register contains a physical address
(5) the value in a bounds register must be an integer multiple of the size of a page

Answer (just give numbers): ______________________________

(Q5) (2 points) Which of the following statements are correct about **paravirtualization**?

(1) inside a commercial paravirtualized OS, usually there are no device drivers and there are no file systems
(2) a paravirtualized OS is indistinguishable from the real OS in the sense that it can also run on the hardware the real OS was designed to run on
(3) VMware is well-known for its paravirtualization patent
(4) one way to implement paravirtualization is to to fix the hardware so that an OS can run inside a virtual machine without modification
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q6) (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 (1 pt)**
T (hrs)	**8**	**10**	**9**	**8**	**-**
wt (hrs)					

(Q7) (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 to lock mutex `m` correctly and safely?

(1) it's safe to use the code as-is to lock mutex `m` because reading the value of m->locked is an atomic operation
(2) it's safe to use the code as-is to lock mutex `m` because setting the value of m->locked is an atomic operation
(3) it's safe to use the code as-is to lock mutex `m` because it's impossible for multiple CPUs, sharing the same bus, to read from the same memory location at exactly the same time
(4) it's safe to use the code as-is to lock mutex `m` because it's impossible for multiple CPUs, sharing the same bus, to write to the same memory location at exactly the same time
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q8) (2 points) Let's say that the address space of a user space in **weenix** looks like the following:

VADDR RANGE	PROT	FLAGS	MMOBJ	OFFSET	VFN RANGE
0x0803a000-0x08049000	rw-	PRIVATE	0xcfe0c034	0x00009	0x0803a-0x08049
0x08049000-0x0804e000	r-x	PRIVATE	0xcfe0c004	0x00008	0x08049-0x0804e
0x0804e000-0x08062000	rw-	PRIVATE	0xcfe0c064	0x00006	0x0804e-0x08062

If you get a page fault with vaddr = 0x0805d668, what **pagenum** would you use to lookup a page frame when you are handling a page fault? Please just give an integer value answer (no partial credit for this problem).

(Q9) (2 points) Which of the following statements are correct about the basic (two-level) virtual memory scheme where a virtual address is divided into a virtual page number (say 20 bits) and an offset (say 12 bits) on a 32-bit machine?

(1) virtual page number is just an array index into a page table which has 2^{20} entries
(2) an entry in a page table contains a 20-bit physical page number no matter how much physical memory is present
(3) during address translation, a physical address is obtained by adding a physical page number with the 12-bit offset
(4) during the lifetime of a user process, the mapping of virtual pages to physical pages must stay the same
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q10) (2 points) Which of the following statements are **correct** about **undo journaling** and **redo journaling**?

(1) in undo journaling, you write “after images” of disk blocks in the journal
(2) in redo journaling, you write “before images” of disk blocks in the journal
(3) in redo journaling, you write “after images” of disk blocks in the journal
(4) in undo journaling, you write “before images” of disk blocks in the journal
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q11) (2 points) Which of the following statements are correct about the naive spin lock implementation vs. the **“better” spin lock implementation**?

(1) no matter which spin lock implementations you use, the impact on system performance is the same if the spin lock is currently unavailable
(2) both the naive spin lock and the “better” spin lock do busy-waiting
(3) using the “better” spin lock can lock the spin lock faster than the naive spin lock if the spin lock is currently unavailable
(4) using the “better” spin lock can lock the spin lock faster than the naive spin lock if the spin lock is currently available
(5) the naive spin lock does busy-waiting while the “better” spin lock does not do busy-waiting

Answer (just give numbers): ______________________________

(Q12) (2 points) Which of the following statements are correct about **LFS (log-structured file system**?

(1) LFS is designed to achieve what other file systems cannot which is to use close to 100% of the disk transfer capacity when reading from to the disk
(2) the inode map in LFS achieves the same functionality as the disk map in S5FS
(3) the checkpoint file in LFS achieves the same functionality as the superblock in S5FS
(4) LFS's append-only and never delete/modify make LFS not very useful in practice for any type of file systems
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q13) (2 points) Which of the following statements are correct about the **scheduler activations model**?

(1) the down side of the scheduler activations model is that if a user thread makes a system call, another user thread in in the user process cannot make a system call until the first thread has returned from the kernel
(2) scheduler activations model is not popular because its insecure to let user-space schedulers to make scheduling decisions
(3) it's difficult to make time-slicing work well in scheduler activations model
(4) in the scheduler activations model, the kernel does not schedule/assign CPUs to threads; instead, the kernel schedules/assigns CPUs to processes
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q14) (2 points) Which of the following is correct about **management of page frames**?

(1) in Linux, user pages can be found in all three physical memory "zones"
(2) in Linux, a dirty and inactive page frame must be freed/deallocated after it has been "cleaned"
(3) if the idea of "working set" is fully implemented in the OS, thrashing can be prevented
(4) if the "local allocation" scheme is implemented in the OS, thrashing cannot happen
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q15) (2 points) Which of the following statements are correct about what would happen when an application running inside a virtual machine makes a system call?

(1) the system call will first trap into the VMM
(2) the system call will eventually get turned into an upcall to reach the guest OS
(3) the system call will trap into the guest OS first
(4) the system call will eventually be emulated by the VMM
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q16) (2 points) Which of the following statements are correct about the **NOR** vs. **NAND** flash memory technologies?

(1) for a NOR flash, the smallest addressable unit for reading is a page
(2) a NAND flash is byte-addressable while a NOR flash is not byte-addressable
(3) a NAND flash is more suitable to be used in a file system than a NOR flash
(4) for writing, a NAND flash is page-erasable but not block-erasable
(5) a NOR flash is byte-addressable while a NAND flash is not byte-addressable

Answer (just give numbers): ____________________

(Q17) (2 points) For a terminal, input characters may need to be processed/edited in some way before they reach the application. Which of the following **data structures** are used to solve this problem?

(1) a B tree or a B+ tree
(2) a translation lookaside buffer
(3) a hash table that uses extensible hashing
(4) a memory map and a page table
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q18) (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 8 ticket, thread B 7 tickets, thread C 5 tickets, and thread D 7 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:4" for iteration 1 since A is the "winner" of iteration 1 and the winning pass value is 4.) 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	4	34	23	17
2				
3				
4				
5				
6				
7				