CSci 402 - Operating Systems

Final Exam (DEN Section)

Spring 2022

[9:00:00am-9:40:00am), Friday, May 6

Instructor: Bill Cheng

Teaching Assistant: Zhuojin Li

(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 **I/O virtualization**?

(1) VMware's I/O virtualization solution performs better than Xen's I/O virtualization solution
(2) in Xen's solution to I/O virtualization, only a few device drivers in the guest OS has to be rewritten in order for Xen to use them
(3) I/O virtualization is not as big of a problem in building virtual machines for high performance servers because only a small number of devices need to be supported
(4) I/O virtualization for desktop machines is challenging because it's virtually impossible for virtual machine vendors to support all devices
(5) in VMware's solution to I/O virtualization, device drivers in the guest OS does not have to be modified to be used inside the virtual machine

Answer (just give numbers): ____________________

(Q2) (2 points) Let's say that you have a 32-bit virtual address and it's divided into 15 bits of **tag**, 6 bits of **key**, and 11 bits of **offset**. If your processor's **translation lookaside buffer (TLB)** uses a 8-way associative cache structure, (a) how many **cache lines** does this TLB have, and (b) how many **bits** of **tags** can be stored in the **entire TLB**? Please give either a numerical answer or a simple numerical expression.

(Q3) (2 points) Which of the following statements are correct about the **inverted page table** scheme?

(1) in an inverted page table scheme, a hash function is used and the input to this hash function is a combination of process ID and physical page number
(2) in an inverted page table scheme, a hash function is used and the input to this hash function is a combination of thread ID and virtual page number
(3) the main advantage of using a inverted page table scheme is its lookup speed
(4) in an inverted page table scheme, the page table used by each process looks like an "inverted tree"
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q4) (2 points) The following implementation of the infinite loop inside **sched_switch()** in **weenix** is known to have a **race condition** when the **RunQueue** is empty and the weenix kernel may appear to be frozen until another interrupt is generated:

```
(1)  intr_setipl(IPL_HIGH);
(2)  while(queme_empty(RunQueue)) {
(3)    intr_setipl(IPL_LOW);
(4)    intr_wait(); // atomically enable interrupt and halt CPU
(5)    intr_setipl(IPL_HIGH);
(6)  }
```

Assuming that the rest of the weenix kernel is implemented correctly and the infinite loop is entered becuase the RunQueue was empty. Under what condition will this interrupt cause the weenix kernel to appear to be "frozen" with a pending interrupt?

(1) if interrupt is disabled in the CPU and an interrupt is **generated** between lines (2) and (3) above
(2) if interrupt is disabled in the CPU and an interrupt is **generated** between lines (3) and (4) above
(3) if interrupt is enabled in the CPU and an interrupt is **generated** between lines (2) and (3) above
(4) if interrupt is enabled in the CPU and an interrupt is **generated** between lines (3) and (4) above
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q5) (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)	5	9	8	9	-
wt (hrs)					

(Q6) (2 points) Which of the following statements are correct for a **forward-mapped (multilevel) page tables** in an x86 CPU, where a 32-bit virtual address is divided into a 10-bit page directory number, a 10-bit page table number, and a 12-bit offset?

(1) since every user space program must have at least 3 memory segments, at least 3 entries in the user portion of a page directory table must be valid
(2) compare to a basic (two-level) page table scheme, address translation is faster when a multilevel page table scheme is used
(3) the basic (two-level) page table scheme is more space efficient than the multilevel page table scheme
(4) the size of a page table is the same as the size of a physical memory page
(5) the size of a a page directory entry is the same as the size of a page table entry

Answer (just give numbers): ________________________________

(Q7) (2 points) Which of the following statements are correct about having a **monolithic kernel**?

(1) the main disadvantage of a monolithic kernel is poor security
(2) a monolithic kernel encourages system programmers to write more elegant code
(3) the main advantage of a monolithic kernel is performance
(4) the weenix kernel is a monolithic kernel
(5) a monolithic kernel is typically more robust (i.e., crashes less) because it's not broken into little pieces

Answer (just give numbers): ________________________________

(Q8) (2 points) Which of the following statements are correct about the **free block list** in **S5FS**?

(1) the on-disk data structure for the free block list in S5FS is basically a singly-linked list
(2) each node in the free block list in S5FS can contain 100 disk block pointers
(3) the head and tail of the free block list in S5FS are stored in the superblock
(4) free inodes are used to keep track of free disk blocks in S5FS
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________

(Q9) (2 points) Which of the following statements are correct about **futex**?

(1) futex is considered "fast" because if the futex is available, a user thread can lock it quickly in user space without making a system call
(2) in order for a futex to function correctly when there are multiple CPUs, the kernel is required to be a non-preemptive kernel
(3) futex is designed to work only in multi-CPU systems and will not work in single-CPU systems
(4) the best place to use a futex is inside the kernel
(5) if a futex is currently **locked** and not being released, a thread calling `futex_lock()` must enter the kernel to wait for the lock to be released

Answer (just give numbers): ____________________

(Q10) (2 points) Which of the following is correct about **management of page frames**?

(1) in Linux, a dirty and inactive page frame must be freed/deallocated after it has been "cleaned" (i.e., content written back to disk)
(2) Linux uses clock algorithm to determine if a page frame is recently used or not
(3) if the "local allocation" scheme is implemented in the OS, thrashing cannot happen
(4) in Linux, user pages can be found in all three physical memory "zones"
(5) if the idea of "working set" is fully implemented in the OS, thrashing can be prevented

Answer (just give numbers): ____________________

(Q11) (2 points) Which of the following statements are correct about what would happen when an appplication running inside a virtual machine makes a system call?

(1) the system call will trap into the guest OS first
(2) the system call will trap into the VMM first
(3) the system call will eventually be emulated by the VMM
(4) the application cannot tell if it's running inside a virtual machine or not when it makes a system call
(5) the system call will eventually get turned into an upcall to reach the guest OS

Answer (just give numbers): ____________________

(Q12) (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 7 ticket, thread B 7 tickets, thread C 6 tickets, and thread D 9 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:9" for iteration 1 since A is the "winner" of iteration 1 and the winning pass value is 9.) 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	9	19	29	12
2				
3				
4				
5				
6				
7				

(Q13) (2 points) Let's say that you are using a **rate-monitonic scheduler** to schedule 4 periodic tasks with $T_1 = 0.5$, $P_1 = 4$, $T_2 = 1$, $P_2 = 4.5$, $T_3 = 0.5$, $P_3 = 5$, and $T_4 = 1$, $P_4 = 6.5$. Let's say that you schedule all 4 period tasts 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-monitonic 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).

(Q14) (2 points) Which of the following statements are correct about the **FIFO** scheduler?

(1) "starvation" at the scheduler is not possible for the FIFO scheduling policy
(2) the FIFO scheduler has the largest variance in waiting time among all scheduling disciplines
(3) for the FIFO scheduler, average waiting time does not depend on the ordering of jobs at the run queue
(4) the FIFO scheduler is inherently unfair to short jobs
(5) it appears to be a fair scheduling policy

Answer (just give numbers): ______________________________

(Q15) (2 points) A correct implementation of **straight-threads synchronization** for a single CPU system 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 currently, thread X owns mutex **m1** and thread Y owns a different mutex **m2**. Let's say that thread X is currently waiting for I/O. Which of the following statements are correct about what would happen if thread Y now calls **mutex_lock(m1)**?

(1) thread X will be moved into the run queue immediately
(2) thread X will enter **mutex_unlock()** immediately and thread Y will return from mutex_lock() immediately with mutex **m1** locked
(3) thread Y will enter **thread_switch()** and thread X will return from **thread_switch()** immediately
(4) thread X will enter **thread_switch()** and thread Y will return from **thread_switch()** immediately
(5) thread Y will go to sleep in **m1**'s mutex queue

Answer (just give numbers): ______________________________

(Q16) (2 points) Which are the reasons the **line-discipline code** is made into a separate **module**?

(1) make the terminal device appear to be more responsive
(2) reduce the size of kernel modules so that the kernel would run faster
(3) hardware manufacturers should not be trusted to access kernel data structures
(4) so that device manufacturers can focus on device-specific issues and don't have to write the common character-handling code
(5) separate the device dependent part from the device independent part in dealing with terminal devices

Answer (just give numbers): ________________________________

(Q17) (2 points) which of the following statements are correct about the **N x 1 (two-level)** thread implementation model?

(1) in this model, when a user thread makes a system call and gets blocked inside the kernel, other threads in the same process can still run as long as they don't make system calls
(2) this model is used in the old days when the kernel didn't know about multithreading in user space programs
(3) in this model, when one user thread wants to give up the processor to switch to another user thread in the same process, it must make a system call
(4) in this model, a user thread does not need to trap into the kernel to lock or unlock a mutex
(5) in this model, thread creation and destruction must be implemented as system calls

Answer (just give numbers): ________________________________

(Q18) (2 points) Which of the following statements are correct about **physical vs. virtual addresses** on a 32-bit machine?

(1) there is a system call a user thread can call to ask the OS to convert a user space virtual address into physical address
(2) a device driver uses physical addresses to execute code but use virtual address to access data on a device
(3) a thread uses physical addresses when it first got created in the kernel and switch to use virtual addresses when it runs in the user space for the first time
(4) a user thread can use physical addresses if it makes a system call and turns into a kernel thread
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________