CSci 402 - Operating Systems

Final Exam (TT Section)

Spring 2022

[9:00:00am-9:40:00am), Tuesday, May 10

Instructor: Bill Cheng

Teaching Assistant: Rohan Madhani

(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) In **weenix** (which runs on an x86 CPU), when a **user program** makes a **memory reference**, which of the following are possible conditions that can cause a **page fault** and the user program **may not** be terminated (assuming that your kernel code is perfect)?

(1) the memory access is a write operation and the corresponding page table entry's validity/present bit is 0 (i.e., invalid)
(2) the memory access is a write operation and the corresponding page table entry's access protection bits are set for "read-only" and its validity/present bit is 1
(3) the memory access is a read operation and the corresponding page table entry's validity/present bit is 1 but the physical page number is invalid
(4) the memory access is a write operation and the corresponding page table entry is "private" and its validity/present bit is 1
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q2) (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 5 ticket, thread B 6 tickets, thread C 8 tickets, and thread D 8 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:3" for iteration 1 since A is the "winner" of iteration 1 and the winning pass value is 3.) 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	(3)	24	35	16
2				
3				
4				
5				
6				
7				

(Q3) (2 points) Which of the following are approaches used to **reduce page fault latency**?

(1) lazy evaluation
(2) use a pageout daemon
(3) prefetching
(4) use a FIFO page replacement policy
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q4) (2 points) Which of the following statements are correct about the **round-robin (RR) / time-sliced** scheduler?

(1) time-slice values should be as large as possible to improve average waiting time
(2) with this scheduler, "starvation" at the scheduler cannot occur
(3) it typically has a smaller average waiting time than SJF scheduling
(4) it typically has a larger variance in waiting time than SJF scheduling
(5) time-slice values should be as small as possible to improve average waiting time

Answer (just give numbers): _____

(Q5) (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) LFS's append-only and never delete/modify make LFS not very useful in practice for any type of file systems
(3) the checkpoint file in LFS is equivalent to the superblock in S5FS
(4) the inode map in LFS is equivalent to the disk map in S5FS
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q6) (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 0x0000c 0x0803a-0x08049
0x08049000-0x0804d000  r-x  PRIVATE 0xcfe0c004 0x0000b 0x08049-0x0804d
0x0804d000-0x08062000  rw-  PRIVATE 0xcfe0c064 0x0000a 0x0804d-0x08062
```

If you get a page fault with vaddr = 0x0805c668, 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).

(Q7) (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 `thread_switch()` is missing an argument that specifies which CPU to use
(2) cannot use the code as-is because access to **RunQueue** needs to be protected if there is only one RunQueue
(3) cannot use the code as-is because **swapcontext()** must include an argument to specify which CPU to use for context swapping
(4) cannot use the code as-is because **RunQueue** must be an array since we have multiple CPUs
(5) cannot use the code as-is because **CurrentThread** must be an array since we have multiple CPUs

Answer (just give numbers): _____

(Q8) (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 translation lookaside buffer
(2) a hash table that uses extensible hashing
(3) a B tree or a B+ tree
(4) a memory map and a page table
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q9) (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 the mutex **m** (i.e., has it locked). If thread X calls **mutex_unlock()** and the mutex queue is **not** empty, the thread at the head of the mutex queue (let's call it thread Y) is supposed to own the mutex next. The above code would dequeue thread Y from the mutex queue and enqueue thread Y to the run queue **without unlocking the mutex**. Referring to the above code, which of the following statements are correct about **the next time thread Y will run** in the CPU?

(1) thread Y will return from the **thread_switch()** function inside **mutex_lock()** but thread X is still the owner of mutex **m**
(2) since thread Y is the new mutex owner, thread Y will call **mutex_lock()** again and will successfully lock mutex **m**
(3) thread Y may wake up inside **thread_switch()** but will go to sleep again insdie **thread_switch()** without returning from **thread_switch()**
(4) even though thread Y is the new mutex owner, thread Y will still call **mutex_lock()** and may go to sleep again in **thread_switch()**
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q10) (2 points) Considering only **clustered hash page table** schemes and **(non-clustered) hashed page table** schemes, which of the following statements are correct?

(1) the performance of non-clustered hash page tables depends on the lengths of the hash conflict/collision resolution chains
(2) clustered hash page tables would perform better if address space is truly sparsely allocated
(3) the performance of clustered hash page tables is independent of how address space is allocated
(4) non-clustered hash page tables typically performs better than clustered page tables
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q11) (2 points) Which of the following statements are **correct** about **undo journaling** and **redo journaling**?

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

Answer (just give numbers): _____

(Q12) (2 points) Which of the following statements are correct about scheduling using **multi-level feedback queues**?

(1) aging can be used to address the thread starvation problem for this scheduler
(2) dynamic thread priorities are used with this scheduler
(3) the main goal of aging is to increase throughput for this scheduler
(4) aging is not necessary because threads can never starve if you use this scheduler
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q13) (2 points) Which of the following statements are correct about **backing store**?

(1) a page in a shadow object must have its backing store in swap space
(2) none of the pages in a read-write private mapping of a file needs a backing store in swap space
(3) none of the pages in a read-write shared mapping of a file needs a backing store in swap space
(4) a page in a read-only mapping of a file must have its backing store in swap space
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q14) (2 points) Which of the following statements are correct about using **base and bounds** registers in a **segmented virtual memory** scheme?

(1) the base register contains a physical address
(2) 4 is a valid value for a bounds register for a memory segment of size 4
(3) the base register contains a virtual address
(4) the value in a bounds register must be an integer multiple of the size of a page
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q15) (2 points) The naive **spin lock** looks like the following:

```
void spin_lock(int *mutex) {
  while(CAS(mutex, 0, 1))
    ;
}
```

Which of the following statements are correct about a thread calling the above function to lock a mutex in a **multiple CPU** system?

(1) this code is inefficient because **CAS()** is a system call
(2) **CAS()** is only inefficient if this code runs in the user space
(3) by calling this function, a thread that does not own the mutex can slow down the mutex owner thread running in another processor
(4) when the mutex is currently held by another thread, this code can then be very inefficient
(5) the code is every inefficient no matter if another thread is holding the mutex or not

Answer (just give numbers): _____

(Q16) (2 points) Which of the following statements are **correct** about **microkernel**?

(1) one main differences between a message port and a Unix pipe is that you can assign names to Unix pipes
(2) access control in a microkernel system typically is based on user IDs and group IDs just like a traditional Unix system
(3) in the design of the microkernel architecture, device drivers cannot be moved into user space
(4) almost all microkernel implementations have good performance
(5) none of the above is a correct answer

Answer (just give numbers): _____

(Q17) (2 points) Let's say that you are using a **rate-monitonic scheduler** to schedule 4 periodic tasks with $T_1 = 0.5$, $P_1 = 4.5$, $T_2 = 1$, $P_2 = 5$, $T_3 = 0.5$, $P_3 = 5.5$, and $T_4 = 1$, $P_4 = 7$. 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).

(Q18) (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)	9	6	7	6	-
wt (hrs)					