CSci 402 - Operating Systems
Final Exam (AM Section)
Spring 2021

(9:00:00am - 9:40:00am, Tuesday, May 11)

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) What did **FFS** do to **improve disk access time** of **S5FS**?

(1) use a new disk map data structure in inodes to improve lookup time
(2) increase block size
(3) use cylinder group to improve rotational latency
(4) bit-interleaving to improve seek time
(5) use RAID

Answer (just give numbers): ______________________________

(Q2) (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	8	7	-
wt (hrs)					

(Q3) (2 points) On a uniprocessor, which of the following steps are **required** when a **pageout daemon** wants to **free up** a **dirty/modified page** that belongs to a user process on an "inactive page list"?

(1) unmap this page from the memory maps of all the processes that share this page
(2) "pin" the page when it is being written to the disk
(3) look at data structures in the memory map of the user process to locate the corresponding backing store
(4) flush the entire translation lookaside buffer
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q4) (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?

(1) the memory access is a read operation and the corresponding page table entry's access protection bits are set for "write-only" and its validity/present bit is 1
(2) the corresponding page table entry's validity/present bit is 0 (i.e., invalid)
(3) the memory access is a write operation and the corresponding page table entry is "private" and its validity/present bit is 1
(4) 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
(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 **stride scheduling**. You have decided to give thread A 1 ticket, thread B 3 tickets, thread C 2 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 "D:3" for iteration 1 since D 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	8	7	5	3
2				
3				
4				
5				
6				
7				

(Q6) (2 points) Which of the following are possible steps that the kernel would take during the handling of a **regular page fault** (i.e., the kind that you would expect) where the **corresponding page table entry (PTE) is invalid**?

(1) if no free page is available, find an in-use and clean page and use it immediately (without getting blocked in the kernel)
(2) if no free page is available, find a "pinned" page that is not-recently-used and free it immediately (without getting blocked in the kernel)
(3) if no free page is available, find an in-use but dirty page that is not-recently-used and use them immediately (without getting blocked in the kernel)
(4) find a free page from the free page list maintained by the kernel
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________

(Q7) (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 enabled 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 (2) and (3) above
(3) if interrupt is enabled in the CPU and an interrupt is **generated** between lines (3) and (4) above
(4) if interrupt is disabled 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): ________________________________

(Q8) (2 points) Let's say that you have a 32-bit virtual address and it's divided into 14 bits of **tag**, 6 bits of **key**, and 12 bits of **offset**. If your processor's **translation lookaside buffer (TLB)** uses a 4-way associative cache structure, (1) how many **cache lines** does this TLB have, and (2) how many **bits** of **tags** can be stored in the **entire TLB**? Please give either a numerical answer or a numerical expression.

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

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

Answer (just give numbers): ________________________________

(Q10) (2 points) Which of the following statements are correct about **scheduling**?

(1) rate-monotonic scheduling is also useful for scheduling non-periodic jobs
(2) in a **multi-processor** system, to take advantage of **cache affinity**, it's better to use **one shared queue** for multiple processors because you won't have to **load balance**
(3) in a hard real-time system, an EDF (earliest deadline first) scheduler usually performs worse (i.e., schedule less jobs successfully) than a rate-monotonic scheduler
(4) in a **multi-processor** system, **cache affinity** means that after a thread has run on a particular processor, it's beneficial to schedule it on the same processor next time it needs to run
(5) priority inversion is possible even if all you have are kernel threads

Answer (just give numbers): ________________________________

(Q11) (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 0x0000e 0x0803a-0x08049
   0x08049000-0x0804d000  r-x  PRIVATE 0xcfe0c004 0x0000f 0x08049-0x0804d
   0x0804d000-0x08052000  rw-  PRIVATE 0xcfe0c064 0x00009 0x0804d-0x08052
```

If you get a page fault with vaddr = 0x0804c668, 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).

(Q12) (2 points) Which of the following statements are correct about the **scheduler activations model**?

(1) scheduler activations model is not popular because its insecure to let user-space schedulers to make scheduling decisions
(2) in scheduler activations model, multiple user threads in the same user process can be making system calls concurrently
(3) it's difficult to make time-slicing work well in scheduler activations model
(4) scheduler activations model does not schedule threads in the kernel; instead, it schedules processes in the kernel
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q13) (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) the size of a page table is the same as the size of a physical memory page
(2) compare to a basic (two-level) page table scheme, address translation is faster when a multilevel page table scheme is used
(3) the size of a a page directory entry is the same as the size of a page table entry
(4) the basic (two-level) page table scheme is more space efficient than the multilevel page table scheme
(5) since every user space program must have at least 3 memory segments, at least three entries in the user portion of a page directory table must be valid, even in weenix

Answer (just give numbers): ______________________________

(Q14) (2 points) Which of the following are useful approaches to **reduce page fault latency**?

(1) prefetching
(2) use a pageout daemon
(3) use a larger translation lookaside buffer
(4) use a FIFO replacement policy
(5) lazy evaluation

Answer (just give numbers): ______________________________

(Q15) (2 points) Which of the following statements are correct about **real-time** systems and threads?

(1) a real-time thread is a thread that can schedule itself to run at the time it wants without the help of the scheduler
(2) a real-time thread is a thread that uses timer-related system calls
(3) there is not much difference between a soft real-time system and a hard real-time system
(4) a real-time thread is a thread that must start running inside the CPU before a deadline
(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 **round-robin (RR) / time-sliced** scheduler?

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

Answer (just give numbers): ______________________________

(Q17) (2 points) Which of the following statements are correct about a **B+ tree of order** $m = 7$?

(1) since m is 7, it's okay if an intermediate node (i.e., neither a root node nor a leaf node) has 2 or 3 child nodes
(2) since m is 7, it's okay if an intermediate node (i.e., neither a root node nor a leaf node) has 5 or 6 child nodes
(3) since m is 7, the root node must have at least 4 child nodes
(4) since m is 7, the height of the B+ tree must be strictly less than 4
(5) none of the above is a correct answer

Answer (just give numbers): ____________________

(Q18) (2 points) Which of the following are maintained in a **S5FS superblock**?

(1) disk map
(2) inode cache
(3) inode numbers of all free inodes
(4) root node of the B+ tree for locating all the inodes
(5) first and last nodes of the free list

Answer (just give numbers): ____________________