CSci 402 - Operating Systems

Midterm Exam (DEN Section)

Fall 2023

(10:00:00am - 10:40:00am, Wednesday, Oct 25)

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: 36 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 kernel mutexes in **weenix**?

(1) since the **weenix** kernel is preemptive, kernel mutexes must be implemented
(2) there can only be one thread running in the **weenix** kernel at a time, so there is really no need for kernel mutexes
(3) since there is only one CPU in **weenix**, there is no reason for having kernel mutexes
(4) **weenix** uses mutexes to ensure that only one kernel thread is accessing a particular device at a time
(5) none of the above is a correct answer

Answer (just give numbers): ____

(Q2) (2 points) Let say that X, Y, Z are regular Unix user processes and X's parent is Y and Y's parent is Z. When process Y dies, what would happen?

(1) the OS kernel will terminate process X because its parent is now dead
(2) process Z becomes parentless
(3) process Z becomes process X's new parent
(4) the INIT process becomes process X's new parent
(5) none of the above is a correct answer

Answer (just give numbers): ____

(Q3) (2 points) Which of the following statements are correct about devices?

(1) PIO devices can only read from physical memory but cannot write to physical memory
(2) DMA devices needs to be told what operation to perform
(3) PIO devices are considered more "intelligent" than DMA devices
(4) DMA device can only function correctly after the CPU downloads a "program" into it
(5) none of the above is a correct answer

Answer (just give numbers): ____

(Q4) (2 points) Which of the following statements are correct about a Unix **pipe**?

(1) a pipe has two ends, a user program decides which end is for reading and which end is for writing
(2) a pipe is like a file and you can use **lseek()** to move the cursor position to the beginning of a the pipe just like you can move it to the beginning of a file
(3) a pipe can be used by one user thread to send characters to the thread itself
(4) a pipe object lives in the user space
(5) none of the above is a correct answer

Answer (just give numbers): ____________________________________

(Q5) (2 points) Which statements are correct about **copy-on-write**?

(1) every time a thread writes to a private page, the page gets copied and the write goes into the new copy of the page
(2) never writes to a shared and writable memory page, only writes to a copy of that page
(3) a private page is read-only and can never be copied or written into
(4) a shared page can never be written into more than once
(5) none of the above is a correct answer

Answer (just give numbers): ____________________________________

(Q6) (2 points) Which statements are correct about what **device independence** may mean when you are **designing or implementing** an OS?

(1) a device driver must be able to run in any OS
(2) the kernel must not depend on any device drivers that's not installed by the kernel
(3) two devices must not share the same code in the kernel
(4) no need to redesign the kernel in order to support a new device made by a new device manufacturer
(5) none of the above is a correct answer

Answer (just give numbers): ____________________________________

(Q7) (2 points) Let's say that you have a uniprocessor and a user thread (thread X) is executing in an x86 CPU when a hardware interrupt occurs. Which of the following statements are true?

(1) in some OS, the interrupt handler will run in user mode and will use the user-space stack of thread X
(2) in some OS, the kernel would allocate a new kernel stack to be used by the interrupt handler
(3) in some OS, a special kernel stack is shared by all interrupt handlers
(4) in some OS, the interrupt becomes pending until thread X traps into the kernel, then the interrupt handler will be invoked using the kernel stack for thread X
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________

(Q8) (2 points) In Unix, a directory is a file. Therefore, I will use the term "directory file" to refer to the **content** of a directory. Which of the following statements are correct about a Unix directory?

(1) a directory file contains a mapping from file names to disk addresses
(2) if file Y is in directory X, the file size of Y is stored in the directory file for X
(3) if directory Z is a subdirectory of directory X, the inode number of directory X is stored in the directory file for Z
(4) if directory Z is a subdirectory of directory X, the inode number of directory Z is stored in the directory file for X
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________

(Q9) (2 points) Assuming that the "`...`" in the code below represents meaningful code and the program compiles perfectly.

```
int main(int argc, char *argv[])
{
  int i;
  for (i = 0; i < 10; i++)
    pthread_create(...);
  return 0;
}
```

Which of the following are **bad things that can happen** with the above code?

(1) main thread may run in parallel with the child threads
(2) **exit()** may get called before any child thread get a chance to run
(3) **pthread_exit()** may get called before any child thread get a chance to run
(4) all child threads may finish before the main thread dies
(5) none of the above is a correct answer

Answer (just give numbers): ________________________________

(Q10) (2 points) Which statements are correct about **Unix signals**?

(1) when a signal is generated, if it's blocked, it is lost
(2) an application cannot prevent the OS from delivering SIGINT
(3) if a SIGINT handler is invoked to deliver SIGINT, SIGINT is blocked inside that SIGINT handler
(4) when a signal is generated, if it's blocked, the kernel will get it delivered by creating a new user thread
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q11) (2 points) What is the difference between a **hard link** and a **soft/symbolic link** in Unix?

(1) when you add a hard link to a file, the kernel increases reference count in the corresponding **file object** while it doesn't do that for a soft link
(2) when you add a soft link to a file, the kernel increases reference count in the corresponding **file object** while it doesn't do that for a hard link
(3) a hard link is a file while a soft link is just a inode reference
(4) a soft link can be added to link to an existing directory while a hard link cannot
(5) on systems with multiple file systems, a hard link can link to a file in another file system while a soft link cannot

Answer (just give numbers): ______________________________

(Q12) (2 points) Assuming that thread X calls **pthread_cond_wait()** to wait for a specified condition, what bad thing can happen if **pthread_cond_wait(cv,m)** is **not atomic** (i.e., pthread library did not implemented it correctly with respect to atomicity)? Please assume that everything else is done perfectly.

(1) thread X may miss a "wake up call" when another thread signals or broadcasts the CV
(2) thread X may cause the kernel to crash when it calls pthread_cond_wait()
(3) thread X may execute critical section code when it does not have the mutex locked
(4) thread X may become uncancellable
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q13) (2 points) Let’s say that you have an infinitely fast and accurate computer and you run your warmup2 on it with the following commandline: ”`./warmup2 -r 1000 -t g0.txt`” and the content of `g0.txt` is as follows:

```
4
1   4   9
2   3   5
1   4   4
4   3   1
```

How many **milliseconds** into the simulation will packet `p4` (i.e., the 4th packet) leaves the system? Please just give an integer value answer (no partial credit for this problem).

(Q14) (2 points) What’s the reason why it is not possible for a thread (with its code written in C) to **completely delete itself**?

(1) a thread cannot have two user space stacks
(2) a thread cannot be running inside two functions simultaneously
(3) a thread cannot delete its own stack when it’s using the stack
(4) a thread cannot be in user space and in kernel space simultaneously
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q15) (2 points) Which of the following are purposes of a **process**?

(1) keeps track of files that have been closed by the running program
(2) keeps track of threads that belong to the process
(3) under multithreading, decides which thread to run next
(4) keeps track of its address space
(5) none of the above is a correct answer

Answer (just give numbers): ______________________________

(Q16) (2 points) Let's say that your **umask** is set to **0416**. (a) What file permissions will you get (in octal) if you use an editor to create the **warmup1.c** file? (b) What file permissions will you get (in octal) if you use a compiler to create the **warmup1** executable? Please note that if your answer is not in octal, you will not get any credit.

(Q17) (2 points) Which of the following is part of the action that must be taken during a Unix **upcall**?

(1) signal handler makes a system call to figure out the reason for the upcall
(2) trapping into the kernel to terminate a user process
(3) the kernel invokes user space code
(4) context switching from privileged mode to user mode
(5) an interrupt service routine makes a system call

Answer (just give numbers): ____

(Q18) (2 points) Which of the following are commonly found address space memory segments in a Unix address space?

(1) stack segment
(2) thread segment
(3) upcall segment
(4) object segment
(5) dynamic/heap segment

Answer (just give numbers): ____