**CSE 422S – Spring 2021 Exam 2**

**Released 4pm Thursday, March 25th, 2021**

Name (please print):

This exam focuses on concepts and details we have covered in the assigned readings, lectures, and studios. Please answer each question as completely and correctly as possible. Partial credit may be given for incorrect answers that show understanding of the material.

During the exam you may use your notes, text books, and on-line sources of information, but you may not post content from this exam anywhere at any time. Communicating or sharing materials with other people during the exam is not permitted.

Please sign or print your name below to indicate your understanding of, and agreement to abide by, the exam conditions described above:

Exam scoring (to be completed by the grader)

|  |  |  |
| --- | --- | --- |
| Question | Possible | Score |
| 1 | 12 |  |
| 2 | 12 |  |
| 3 | 12 |  |
| 4 | 12+1 |  |
| 5 | 12 |  |
| 6 | 12 |  |
| 7 | 8 |  |
| 8 | 8 |  |
| 9 | 12 |  |
| Total | 100+1 |  |

1. (12 points) which syscalls are used to open and close a file?

When using those syscalls, how is the identity of the file represented?

Which standard I/O library functions are used to open and close a file?

When using those standard I/O library functions, how is the identity of the file represented?

Which syscalls are used to output data to a file and to input data from a file?

Which standard I/O library functions are used to output formatted data to a file and to input formatted data from a file?

2. (12 points) Linux organizes page tables into four levels: Page Map Level (PML), Page Global Directory (PGD), Page Middle Directory (PMD), and Page Table Entry (PTE). If 9 bits are used to index the entries in one of those tables, how many entries can that table hold?

Please explain briefly what the kernel must check first, when it is translating a virtual address and it finds a page table entry that does not have a physical page mapped to it.

Please explain briefly what the kernel does if that check succeeds.

Please explain briefly what the kernel does if that check fails.

3. (12 points) Briefly, what is avoided by always acquiring a given set of locks in the same order, within any thread that needs to hold those locks at the same time?

Briefly, how does having each of those threads always acquire those locks in the same order prevent that?

Briefly, what is one advantage of using a spin lock over using a sleeping lock?

Briefly, what is one advantage of using a sleeping lock over using a spin lock?

4. (12 points) Briefly, how does a fast user space mutex differ from a standard mutex or semaphore?

What operation is implemented by the following code?

int expected, desired;

expected = UNLOCKED;

desired = LOCKED;

while (!\_\_atomic\_compare\_exchange(p,

&expected, &desired, 0,

\_\_ATOMIC\_ACQ\_REL, \_\_ATOMIC\_ACQUIRE)){

expected = UNLOCKED;}

Briefly, why does that code need to (possibly repeatedly) reset the value of the expected variable to be UNLOCKED?

Briefly, what is guaranteed for an atomic\_t variable that is not guaranteed for a volatile int variable?

Extra credit (1 point): Please name another atomic function besides \_\_atomic\_compare\_exchange that we have used in the studio exercises or the lab assignments.

5. (12 points) Briefly, what does the ftruncate call do to a shared memory region?

Briefly, why is using shared memory for communication between two processes more efficient than using other inter-process communication mechanisms we’ve studied?

Briefly, how does the layout of virtual addresses relative to to physical addresses differ between user-space virtual memory and kernel virtual memory?

Briefly, in the copy-on-write optimization following a fork() call, what does setting the parent and child processes’ page table entries to unwriteable ensure will happen if either process then calls write() at one of its virtual addresses?

6. (12 points) Next to each of the statements below, please write T next to it if it is true or write F next to it if it is false.

\_\_\_\_\_ Demand paging maps physical pages as soon as

virtual addresses are allocated.

\_\_\_\_\_ Prepaging maps physical pages as soon as virtual

addresses are allocated.

\_\_\_\_\_ Shared memory always maps to the same virtual

addresses in the processes that share it.

\_\_\_\_\_ Formatted output to a file may be buffered both in

user space and in the kernel.

\_\_\_\_\_ Direct I/O improves most applications’ performance.

\_\_\_\_\_ Code without deadlocks is always free of data races.

\_\_\_\_\_ Code without data races is always free of deadlocks.

\_\_\_\_\_ A process can use a syscall to expand its heap.

\_\_\_\_\_ A single syscall can read data into multiple buffers.

\_\_\_\_\_ All content in the virtual file system is backed up by a

hard disk or other persistent storage device.

\_\_\_\_\_ The MMU performs virtual to physical memory

address translations.

\_\_\_\_\_ Most 32-bit architectures have physical memory

pages that are each 4KB in size.

7. (8 points) Briefly, what does the percpu interface do?

Briefly, what is stored in a cache created by the slab allocator interface?

If a kernel module calls vmalloc() in its init() function, what function should it call in its exit() function?

What function should be used when allocating memory, to improve system security by ensuring that (possibly sensitive) data that had been written to a physical page is overwritten?

8. (8 points) Briefly, what is contained in an operations object in the Linux virtual file system?

Briefly, how do those operations objects support a form of object-oriented programming within the Linux kernel?

Briefly, when is a new file object added to the virtual file system data structures corresponding to a user-space process?

Briefly, what does the bootloader program do, once it is loaded from the /boot directory?

9. (12 points) In the blank next to each phrase on the left, please write the letter for the description that best matches it and that it best matches.

\_\_\_ atomic A. sleep/wake if a lock is not yet acquired

\_\_\_ deadlock B. retries without sleeping until lock is acquired

\_\_\_ data race C. represents a specific mounted filesystem

\_\_\_ file stream D. represents a file or directory

\_\_\_ file descriptor E. exponent for number of pages to allocate

\_\_\_ file offset F. created/used by stdio functions

\_\_\_ order G. created/used by syscalls

\_\_\_ futex H. modified by read, write, and lseek

\_\_\_ spin lock I. represents a single component of a path

\_\_\_ superblock J. completes without interruption

\_\_\_ dentry K. interleaving may corrupt memory

\_\_\_ inode L. can be caused by interdependences of locks