

CS 3204 Midterm

This is a closed-book, closed-internet, closed-cellphone and closed-computer exam. However, you may refer to your sheet of prepared notes. Your exam should have 7 pages with 5 questions totaling 50 points. You have 60 minutes. Please write your answers in the space provided on the exam paper. If you unstaple your exam, please put your initials on all pages. You may use the back of pages if necessary, but please indicate if you do so we know where to look for your solution. You may ask us for additional pages of scratch paper. You must submit all sheets you use with your exam. However, we will not grade what you scribble on your scratch paper unless you indicate you want us to do so. Answers will be graded on correctness and clarity. You may lose points if your solution is more complicated than necessary or if you provide extraneous, but incorrect information along with a correct solution.

Name (printed) _____

I accept the letter and the spirit of the Virginia Tech graduate honor code – I have not given or received aid on this exam.

(signed) _____

| # | Problem | Points | Score |
|---|-----------------|--------|-------|
| 1 | System Calls | 10 | |
| 2 | Protection | 8 | |
| 3 | Semaphores | 14 | |
| 4 | Scheduling | 10 | |
| 5 | Short Questions | 8 | |
| | Total | 50 | |

1 System Calls (10 pts)

Suppose you decide to add a *gettimeofday* system call to Pintos. To user programs, *gettimeofday()* looks like an ordinary function call:

```
struct timeval {
    uint32_t tv_sec;    // seconds
    uint32_t tv_usec;  // microseconds
};

int gettimeofday(struct timeval *p);
```

If successful, *gettimeofday* will write the current time to **p* and return 0. Suppose that the kernel has some way of reading the current time, such as a device driver that can read the PC's built-in real-time clock.

- a) (2 pts) Name one change you would have you to make to Pintos's standard C library for user programs (located in `lib/user/*.`)

- b) (4 pts) What changes, if any, would you have to make to your system call handling framework in `userprog/syscall.c` (excluding the actual implementation of *gettimeofday* itself)?

- c) (4 pts) Suppose you implement *gettimeofday* as a function `sys_gettimeofday()` in `syscall.c`. Suppose a real-time clock device driver implements a function `rtc_gettimeofday()` that has the same signature as `gettimeofday()`, i.e. `int rtc_gettimeofday(struct timeval *p);` Implement `sys_gettimeofday` in terms of `rtc_gettimeofday`!

```
int
sys_gettimeofday(struct timeval *userp)
{
```

```
}
```

You may use the function names you used in your code without explaining them, as long as your names are descriptive.

2 Protection (8 pts)

- a) (4 pts) Suppose an architecture does not provide software trap instructions such as the `int` instruction on the i386. Describe how you would implement system calls on such an architecture.
- b) (4 pts) Suppose an i386 ELF executable gets corrupted on disk in such a way that all bytes inside the code segment are overwritten with the value `0x90`, which is the machine code for a NOP instruction on the i386. What would happen if a process tried to `exec()` this executable?

scratch space

3 Semaphores (14 pts)

- a) (8 pts, 2 pts each) Suppose you have 2 semaphores and 3 concurrent threads as in the example below. Assume that the threads run for a sufficiently long time under the regime of a preemptive scheduler.

| | | |
|--|--|---|
| <pre>struct semaphore S, U; sema_init(&S, 2); sema_init(&U, 0);</pre> | | |
| <pre>// Thread 1 while (1) { sema_down(&S); putchar('A'); sema_up(&U); }</pre> | <pre>// Thread 2 while (1) { sema_down(&U); putchar('B'); putchar('C'); sema_up(&U); }</pre> | <pre>// Thread 3 while (1) { sema_down(&U); putchar('D'); }</pre> |

- i. How many times will 'A' be printed?
- ii. How many times will 'D' be printed?
- iii. What is the minimum number of times that 'B' might be printed?
- iv. Can the output string start with 'D'?

scratch space

- b) (6 pts) Suppose you have 1 semaphore and 3 concurrent threads as in the example below.

```
struct semaphore S;
sema_init(&S, 4); // suppose S represents a resource with 4 units

void resource_user_thread(void *_) {
    while (1) {
        // get two units of S
        sema_down(&S);
        sema_down(&S);
        // (use resource)
        // return two units of S
        sema_up(&S);
        sema_up(&S);
    }
}
...
tid_t t1 = thread_create(..., resource_user_thread, NULL);
tid_t t2 = thread_create(..., resource_user_thread, NULL);
tid_t t3 = thread_create(..., resource_user_thread, NULL);
```

Can this program deadlock?

If so, give the sequence of events leading to deadlock.

If not, say why not. Be specific.

5 Short Questions (8 pts)

a) (4 pts) Why are page sizes always powers of 2?

b) (4 pts) Is it ever safe to pass a pointer to a local variable to another thread? If not, say why not. If so, say under what circumstances.

scratch space