

# Operating Systems Midterm Exam Spring 2018

Name \_\_\_\_\_ NetID \_\_\_\_\_

- **Do not open this exam** until the proctor tells you to begin.
- There are 6 pages in this exam, including this one.
- This exam is closed book - closed notes.
- Write clearly - if we can't read or can't find your answer, it is wrong.

Question Type	Point Total	Scored Amount
0. Processes	18	
1. Scheduling	12	
2. Synchronization	18	
3. Memory	12	
Total:	60	



## 0. Processes

0. While `fork()` makes it seem as if a process is duplicated, in fact it is not. What does the `fork()` call duplicate, however?
1. The virtual memory system enforces process memory boundaries. Name two ways processes can communicate data to each other directly, other than on return.
2. Why can't user threads from the same process be run simultaneously on different CPU cores, while kernel threads can?
3. `init`/the scheduler can not be `fork()`ed, so how is it created during boot?
4. List two things/properties threads can do/have that processes can/do not.
5. What user-written functions are usually never invoked by anyone but the OS?

## 1. Scheduling

0. In what case might a task that has the shortest computation time remaining actually take longer to complete than the task with greatest computation time remaining?
  
1. Describe priority inversion.
  
2. How is it possible to deadlock on resource allocation?
  
3. At what point does reducing the RR quantum start to make processes take longer to complete?

## 2. Synchronization

0. What are the differences between a binary semaphore and a mutex lock?
  
1. Blocking functions often cause context switches and a loss of optimizations like caching and processing time to update scheduler state. Under what circumstances would the overhead of blocking be beneficial?

2. Implement a spinlock below, using the synchronization primitives specified. You should write a `spinlock_create` and `spinlock_acquire` function for each. You can set the return types and parameters of each as you like.

- a. `type sync_lock_test_and_set (type *ptr, type value, ...)`

- b. `type _sync_fetch_and_add (type *ptr, type value, ...)`

3. How do deadlock, livelock and starvation differ?

4. There are two cases where we need hardware instructions. One is synchronization primitives. What is the other, and why can it not be done in software?

### 3. Memory

0. List three benefits of virtual memory, other than swapping out pages.
  
  
  
  
  
  
  
  
  
  
1. Describe internal and external fragmentation and how they differ.
  
  
  
  
  
  
  
  
  
  
2. Page faulting often results in expensive hard drive I/O. Why would you want to have at least some amount of page faulting?
  
  
  
  
  
  
  
  
  
  
3. What are the differences between a segment and page fault?