Operating Systems Midterm Exam (Spring 2014)

(Answers which are too long may not be read.)

No :

Name: Edu-Type : 1 / 2

1. *(10P)* A process can be in five different states. When a user starts a process, which state is it in and which transition does it do at first? In one sentence explain.

The process starts within new (start or create) state and the first transition must be from new state to ready state.

2. *(10P)* Assume that a process is waiting an entry from keyboard. Is it incorrect to say that "this process will run in CPU eventually"? In two sentences explain.

Because the process might starve, or the event that it is waiting on might never happen, it is not guaranteed.

3. *(10P)* Each thread within a process has its own four things. What are they? (Write only their names)

Program counter, Registers, Stack, and State

4. *(10P)* Assume a system with four-CPU provides only a user-level threading package. How many CPUs can a single process use, if the process has four user-level threads? In two sentences explain.

Because kernel is not aware of the user-level threads, user-level threads can reach only one CPU. Therefore, it is not able to run the user-level threads on different processors.

5. *(15P)* Let "*mtx*" be a mutex. When is the following pattern correct? In one sentence explain.

if (! condition) { *wait(&mtx);* *}* *// it is an error if code gets to* *// here with "condition" false*	While the "condition" is still false, because the variable "mtx" may be woken up, the code will never run correctly.

6. *(10P)* You decide to implement a new concurrency primitive, and you want to do it correctly. You will need to consult the manuals for which system components? In one sentence explain the cause.

We need the manuals for compiler (for compiling trends) and processor (for atomic instructions and memory model).

7. *(15P)* Assume you want to implement a web-server by using multithreading, where each thread serves one incoming request by downloading a file from the disk. Assume the OS only provides the normal blocking system call for disk reads. Do you think user-level threads or kernel-level threads should be used? In at most three sentences explain the cause.

Because OS knows only processes (not threads in them) in user-level threading, one I/O call blocks all threads in a process. Whereas one thread does not block others due to I/O calls in kernel-level threads. For this reason we should choose kernel-level threads.

8. *(10P)* If timer interrupt mechanism was not present, what would be? In two sentences explain.

The scheduler mechanism would not be present, so the OS would not be able to control the CPU for multitasking.

9. *(10P)* Identity knowledge of the held threads passed to *thread_sleep(semaphore)*. In one sentence explain the cause.

This idendity knowledge is used to wake up held threads which sleep on the same semaphore variable.