CSS-430 : Operating Systems : Mid-Term-Exam

The Mid-Term Exam will be held online on 04-May-2020, starting at 5:45 pm, for 2 hours. Deadline for uploading answers to Canvas is 8:00 pm (allowing an extra 15 minutes for any upload glitches).

During the exam you may consult notes, books, lectures or the Internet.

Please answer all 20 questions.

Good luck!

1. 5 points.   
   What is the difference between user mode and kernel mode?  
   How does kernel mode prevent Apps from taking control of the computer?  
   Can we build a secure Operating System without kernel mode?  
   Do popular processors support any modes in addition to user and kernel?
2. 5 points.  
   What is the Von Neumann bottleneck?  
   The term was invented many decades ago. Is it still relevant?  
   What strategies do modern computers follow to avoid any Von Neumann bottlenecks?
3. 5 points.  
   What is an interrupt?   
   What is a trap?   
   We describe one as synchronous, the other as asynchronous. What does this mean? And which is which?  
   What is each used for in an operating system such as Linux?   
   Can a user program cause a trap?
4. 5 points.  
   What challenges ae faced by an Operating System for mobile devices, compared with a deskside PC?  
   Name two popular Operating Systems used in mobile devices.
5. 5 points.  
   What is a “system call” or “syscall”?  
   Name any 3 system calls in an Operating System.  
   Does a syscall execute in user mode, or kernel mode?  
   Does the calling process undergo a context switch when it makes a syscall?  
   Does the calling thread undergo a context switch when it makes a syscall?
6. 5 points.  
   There are two main techniques used for IPC (“Inter Process Communication”). Name them.  
   Which technique is best for large transfers? Why?  
   Which technique is used to communicate with remote PCs?  
   Explain briefly the differences in synchronization needs of the two techniques
7. 5 points.  
   Explain the difference between a Monolithic kernel and a Micro kernel?  
   Which one is “better”, and why?  
   Is Linux monolithic or micro?  
   Is Windows monolithic or micro?
8. 5 points.  
   The memory for a process is divided into 4 sections. What are they called?  
   Which for the 4 sections are writeable?  
   Which for the 4 sections are executable?
9. 5 points.  
   What is a context switch?  
   Describe the main steps involved in a Process context switch.  
   Describe the main steps involved in a Thread context switch.  
   Which one is faster? And why?
10. 5 points.  
    40% of a certain process can be parallelized. Use Amdahl’s law to calculate the speedup on a 2-core system. A 3-core system? A 4-core system?
11. 5 points.  
    Describe the difference between so-called user-mode threads, and kernel-mode threads.  
    Can both make use of multiple cores? Explain your answer.  
    Which are lighter-weight (in the sense the consume less CPU to start, and to context switch)?  
    Which kind are “green” threads?
12. 5 points.  
    Suppose our computer has only 1 core. Suppose we have 3 large jobs to run, and each takes an hour. How is it possible that running them side-by-side in the computer will take less than 3 hours.
13. 5 points.  
    Explain the meaning of the following times, associated with CPU scheduling:

* Arrival time
* CPU Burst time
* Response time
* Execution time
* Turn-around time (“TAT”)
* Wait time

1. 5 points.  
   In CPU scheduling, answer the following yes/no questions?

* Can wait time ever exceed execution time?
* Can response time ever be zero?
* is TAT = wait time + execution time?
* Can TAT ever be less than execution time?
* If wait time is zero, is response time also zero?

1. 5 points.  
   Consider the following processes with arrival time and CPU burst times:

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | CPU Burst Time |
| P1 | 4 | 6 |
| P2 | 0 | 5 |
| P3 | 5 | 4 |
| P4 | 8 | 3 |

Draw the Gannt chart for the FCFS (First Come First Served) schedule.

Note: no need to draw a real Gannt chart with different processes on different lines. Just write a text version, like this:  
  
 1 1 3 3 2 2 4 4 etc, where “1” means P1, “2” means P2 and so on. Let’s call this a “simplified Gannt chart”.

1. 5 points.  
   For the same processes P1 thru P4, draw the simplified Gannt chart for the SJF (Shortest Job First) schedule.
2. 5 points.  
   For the same processes P1 thru P4, draw the simplified Gannt chart for the STR (Shortest Time Remaining) schedule. Reminder: STR is like SJF, but with pre-emption possible at each ‘event’
3. 5 points.  
   Describe RR (Round Robin) scheduling  
   What is a “quantum” in RR?   
   What is a typical size for quantum?  
   What happens if we make the quantum too short?  
   What happens to fairness if we make the quantum too long?
4. 5 points.  
   Can a program have more than 1 critical section?  
   Can different threads compete to execute the same critical section?  
   Can different processes compete to execute the same critical section?
5. 5 points.  
   Explain your understanding of the terms “deadlock” and “starvation”  
   How would you know that a multi-thread program has become “deadlocked”? What are its symptoms?  
   Same question, but for “starvation”?