Câu Hỏi Về Process

1. Describe the two general roles of an operating system, and elaborate why these roles are important
2. Using a simple system call as an example (e.g. getpid, or uptime), describe what is generally involved in providing the result, from the point of calling the function in the C library to the point where that function returns
3. Why must the operating system be more careful when accessing input to a system call (or producing the result) when the data is in memory instead of registers?
4. Describe the three state process model, describe what transitions are valid between the three states, and describe an event that might cause such a transition.
5. Multi-programming (or multi-tasking) enables more than a single process to apparently execute simultaneously. How is this achieved on a uniprocoessor?
6. What is a process? What are attributes of a process?
7. What is the function of the ready queue?
8. What is the relationship between threads and processes?
9. Describe how a multi-threaded application can be supported by a user-level threads package. It may be helpful to consider (and draw) the components of such a package, and the function they perform
10. Name some advantages and disadvantages of user-level threads
11. Why are user-level threads packages generally cooperatively scheduled?
12. Enumerate the advantages and disadvantages of supporting multi-threaded applications with kernel-level threads
13. Describe a sequence the sequence of step that occur when a timer interrupt occurs that eventually results in a context switch to another application.
14. Context switching between two threads of execution within the operating system is usually performed by a small assembly language function. In general terms, what does this small function do internally?
15. Why is it generally correct to favour I/O bound processes over CPU-bound processes?
16. Is a single ready queue on a multiprocessor a good idea? Why?
17. What is thread afinity. Why might it improve performance?
18. What are some of the differences between a processor running in *privileged mode* (also called *kernel mode*) and *user mode*? Why are the two modes needed?
19. What are the two main roles of an Operating System?
20. Which of the following instructions (or instruction sequences) should only be allowed in kernel mode?
21. The following code contains the use of typical UNIX process management system calls: fork(), execl(), exit() and getpid(). If you are unfamiliar with their function, browse the man pages on a UNIX/Linux machine get an overview, e.g: man fork
22. Give An Example Of A Process State.?
23. What Is The Resident Set And Working Set Of A Proc**ess?**