

Operating Systems
January 2009

1. Define multitasking. Define and differentiate cooperative and preemptive multitasking. Explain why preemptive multitasking requires hardware support. Describe the kind of hardware support that is required. Outline the sequence of events that happen when one task (processes) is preempted and another process is executed.
2. Define the bounded-buffer producer-consumer problem. Propose a solution (write pseudocode) for it. Give an informal argument describing why your code is correct. You are free to use synchronization-primitives; if you use them, you have to describe their operation first.
3. Define process CPU-burst. Explain why a CPU burst size is important for process scheduling. It seems that the OS needs to know this size before executing the process to decide when to schedule the process. How would the OS determine the CPU-burst size in advance?

Out of the elementary scheduling disciplines:

- first come first served (FCFS),
- shortest job first (SJF),
- round-robin (SRT),
- shortest remaining time (SRT),

describe one and list its advantages compared to the other three.

4. Define a disk block. Explain its relationship to a sector. Explain why a file system implementation has to keep track of free blocks. Early Unix file system implementations maintained a linked list of free blocks. Modern implementations use bitmaps to keep track of free blocks. Describe both strategies and explain why the bitmap strategy is superior.