

Computer Science Qualifying Examination

CPS 210: Operating Systems

January, 1998

1. Two approaches to synchronization are semaphores and mutex/condition variable pairs. Are these facilities equivalent in power, i.e., is there any synchronization problem that can be solved with either facility that cannot be solved with the other? Prove your answer.
2. This question concerns operating system mechanisms for process coordination.
 - a) Most operating systems provide a means to asynchronously notify processes of events (e.g., the Unix *signal* mechanism). Modern Unix systems provide additional primitives to arrange processes into groups, and send signals to all processes in the group. Why is this facility needed? Give an example of its use.
 - b) Most operating systems provide a primitive that allows one process to *join* on another, i.e., wait for it to complete. Modern Unix systems supplement this primitive with a mechanism to asynchronously notify a parent process (via a signal) if a child process exits. If the child process paused or exited due to an event, the parent can issue a system call to determine the type of signal the child received. Why are these facilities needed? Give an example of their use.
3. This question concerns the effect of cache architecture on the implementation of operating system facilities for managing processes and memory.
 - a) Describe the actions an operating system kernel must take to preserve correct cache functioning for the Unix *fork*, *exec*, and *exit* system calls on architectures using the following cache variants: (1) virtual index with virtual tag, (2) virtual index with physical tag, and (3) physical index with physical tag. Be clear about any additional assumptions you make.
 - b) OS memory management choices can have significant performance effects on machines with physically indexed cache architectures, even if only one process is running. Explain. Outline an example of a naive memory management strategy and a more sophisticated one.
 - c) OS memory management choices can have significant performance effects on machines with virtually indexed cache architectures, even if only one process is running. Explain. Outline an example of a naive memory management strategy and a more sophisticated one.
4. This question concerns the FIFO-with-second-chance page replacement policy used in operating systems derived from Mach and 4.4 BSD.
 - a) Explain how this page replacement scheme can adapt to respond to variations in memory demand. What effect does this adjustment have on the quality of the page replacement choices? What effect does it have on the overhead of page replacement?
 - b) Modern implementations of FIFO-with-second-chance scan the inactive list periodically even when there is adequate free memory. Why?

- c) Most modern virtual memory implementations attempt to issue their pageout requests in regularly scheduled bursts. Why is this important?
5. Remote Procedure Call is not a universally accepted model for building distributed systems. Ken Birman has stated that “RPC is a failed paradigm” because it is inadequate to coordinate groups of processes. John Ousterhout and others have pointed out that the RPC model exacerbates concurrency problems that are difficult for programmers to manage. In Mach, most interactions between the microkernel, servers, and user tasks are structured using asynchronous messaging rather than RPC-style request/response exchanges; similarly, the HTTP protocol has been restructured to use primarily asynchronous messaging rather than request/response exchanges. Some designers of peer-peer distributed systems for clusters have also moved away from pure RPC models for a variety of reasons. Discuss the limitations of the RPC model.