Computer Science Comprehensive Exam

Systems

January 16, 1996

This is a closed-book exam. You have 1.5 hours to complete it. Answer all of the questions. Be concise in your answers. You will receive partial credit for partially correct answers, but extraneous remarks may count against you.

1. (a) Describe the "global clock" page replacement algorithm. Be sure to explain what data needs to be maintained by hardware and what data needs to be maintained by software.

   (b) How would the "global clock" algorithm be approximated on a memory architecture with a software managed TLB that supports only *dirty* (modified) and *valid* (present) bits for each page?

2. Suppose that you had a concurrent programming language based on threads (light-weight processes) and monitors. Consider the implementation of a "bare-machine" kernel to support this language on a RISC machine in a single user, single program environment. The "bare-machine" kernel is a runtime support for the language that runs directly ("stand-alone") on the machine, rather than interfacing to an existing operating system such as UNIX.

   (a) Describe the design/implementation of such a kernel. Focus on the concurrency aspects of the language only (e.g., you may ignore I/O.) What functionality and what primitives would the kernel support? How would these primitives be implemented? Describe any pertinent state data structures and sketch how each primitive transforms the state.

   (b) Describe how to implement **wait** and **signal** primitives (**wait**(c) and **signal**(c), where c is a condition variable) for each of the following hardware environments:

      (a) Single processor

      (b) Multi-processor with shared memory

      (c) Network of processors (no shared memory)

   Make *reasonable* assumptions about what primitives are supported by the hardware.

3. With *remote paging*, virtual memory pages that are evicted from a computer's main memory are temporarily stored in the main memory of another computer. The idea of remote paging recently became feasible because of two technological trends: (1) With modern processors and high-speed LANs, it can be faster to request a page from another computer on the same LAN than reading the page from the local disk; and (2), large main memories increase the likelihood that a nearby workstation has some unused main memory.

   (a) Discuss the issues that need to be addressed in the design of a remote paging system. Be sure to consider issues of performance, scalability, fairness, and robustness.

   (b) Sketch a *simple* remote paging algorithm. Specifically, the algorithm must choose a physical page frame on a remote node to be used as temporary storage for an evicted page (page-out), and find any physical page frame on a remote node that contains the data for a requested page (page-in).