Systems Qualifying Exam, Fall 1999
Indiana University Computer Science Dept.

For every problem, you must justify your answer.

Problem 1 (34 points)

a (5 pts). Define semaphores.
b (19 pts). Give pseudo-code for an implementation of semaphores on a uniprocessor. Also give an informal explanation of how and why your code works.
c (10 pts). Describe an implementation of semaphores on a shared-memory multiprocessor. Feel free to use English, pseudo-code, or a mixture. (You do not need to be as detailed as in part b.)

You may make reasonable assumptions about the architecture, provided you state the assumptions explicitly.

Problem 2 (34 points)

Consider a UNIX-like OS that supports the following system calls for process management:

`ProcessID exec(char *file);` Creates a new process that will execute the executable in the file named `file`. Returns -1 if an error occurs, otherwise returns a positive integer that the calling process can use to identify the new process.

`int waitpid(ProcessID pid);` Waits until the child with the specified ProcessID has terminated (if it hasn't terminated already), and returns that process's return value. The return value of a process that terminates by calling `exit(`x`)` is x. The return value of a process that terminates without calling `exit` is -1. If the specified ProcessID does not correspond to a child of the calling process, `waitpid` returns -2.

`void exit(int x);` Terminates the calling process, and sets the process's return value to `x`.

Describe an implementation of these three system calls. Describe appropriate data structures and algorithms. Feel free to use English, pseudo-code, or a mixture. Don't forget about synchronization. Assume the OS supports demand-paged virtual memory.

Problem 3 (32 points)

This problem is about 2-level page tables. "L1" abbreviates "level-1". "PT" abbreviates "page table".

a (16 pts). Is paging L2 PTs feasible and reasonable? In other words, is it feasible and reasonable not to lock a process's L2 PTs in memory while the process is running? If so, describe how this would work. If not, explain why.
b (16 pts). Same question as a., for L1 PTs.

Some issues you might need to consider: what information must always be in memory? how are disk addresses of paged-out PTs computed? what synchronization is needed?