CS 4/53201 **Homework #2** **Operating Systems**

Due in class on Monday 21 October 2002

1. **Why is a context switch between threads faster than a context switch between processes?**

   A context switch between threads involves saving and restoring general-purpose registers, program counter, and stack pointer, while a context switch between processes also involves more extensive state information (e.g., memory state). Furthermore, if the threads are user-level thread, a context switch between threads doesn't even require the overhead of OS involvement.

2. **As discussed in class, when a signal operation is performed on a semaphore, if there are processes waiting, one is woken up. Generally the waiting processes are kept in a first-in first-out (FIFO) queue, which gives a type of semaphores called "strong semaphores". "Weak semaphores", in contrast, are removed from the queue in arbitrary order. Are there any disadvantages in weak semaphores?**

   If there are multiple processes waiting, weak semaphores could result in the starvation of a particular waiting processes if it is repeatedly not the process randomly picked for removal.

3. **If a signal operation is performed on a semaphore, and the corresponding wait operation is performed later, the thread performing the wait operation does not actually wait. Explain.**

   First, consider the case of a wait operation occurring before the signal. If the value of the semaphore is 0, then when the wait occurs, it decrements the value of the semaphore to –1, causing that thread to block. Later on, when the signal occurs, the value of the semaphore is incremented to 0, and the waiting thread is woken up and put on the ready list.

   In contrast, when a semaphore signal is done first, it increments the value of the semaphore, perhaps from 0 to 1. If a wait occurs later, it decrements the value of the semaphore back to zero, but since the value of the semaphore is not negative, the thread continues instead of waiting.

4. **(Exercise 15.9 from OSC) What are the advantages of using dedicated hardware devices for routers and gateways? What are the disadvantages compared to using general-purpose computers?**

   Dedicated hardware can be designed more specifically for a particular task, perhaps resulting in increased performance, and unessential hardware can be eliminated, resulting in lower cost over a general-purpose computer. Unfortunately, additional hardware must be purchased, instead of using an existing computer, and it may not be as easy to upgrade dedicated hardware as it would be to upgrade software in a general-purpose computer.

5. **In multilevel feedback queue CPU scheduling, why doesn't the mechanism for reducing the priority of a process cancel out the aging mechanism?**

   The reduction mechanism gradually reduces the priority of a process, allowing it a chance to get at least some CPU time initially but then reducing its priority the longer it stays in the system.

The aging process raises the priority of a process that has been in the system a long time, so that it eventually will finish. In effect, it does somewhat cancel out the reduction process, but in a way that works together with the reduction process.