Concurrent Systems II

Practical 3
Semaphores

February 11, 2014

This practical is worth 4% of your year-end result. Have your program ready one week from today—i.e. have it ready for inspection on February 18 (MCS students) or February 19 (MAI students). Have a printout also.

Find out what a *semaphore* is, as invented by Edsger Dijkstra around 1968. There are a number of kinds of semaphore, in particular *weak* and *strong* semaphores. In this practical you will have to use semaphores, but you must not use built-in semaphore packages; instead, develop your own semaphores using pthread mutexes and condition variables.

Write a complete threaded program in C on a Linux machine—e.g. `stoker.cs.tcd.ie`—to implement a simulated printer queueing system, where a 'pool' of three printers is available for [simulated] print-job requesters. The printers are all identical, so it makes no difference to the print-job requester which printer is allocated to it. Implement the print-job requesters on separate threads.

Your implementation should ensure two things:

1. Print Jobs are dispatched to printers in the order they are received.
2. Print jobs should be dispatched *fairly* to printers. (What does 'fair' mean in this context?)

(`http://www.scss.tcd.ie/CourseModules/CS3015/Assets/Practicals/p3/practical.pdf`)