

Concurrent Systems II

Practical 6—OSP 2

March 19, 2014

This practical is worth 2% of your year-end result. Have your program ready for next week.

The Practical

OSP 2 is an Operating System in a simulated environment with simulated user programs and a simulated event queue. The whole thing is written in Java.

- Download and install OSP 2. It's available from <http://www.springer.com/computer/swe/book/978-1-84628-842-5>.
- Get the demo part of it working in the Threads directory (see handout).
- Download two files from <http://www.scss.tcd.ie/CourseModules/CS3015/Assets/Practicals/p6/P6Stuff.zip>
These files are replacements for the similarly-named files in the Threads directory. They contain fully implemented “vanilla” versions of the relevant routines. Compile them¹ and make sure they work, say by introducing print statements or errors. At this point, you have a version of OSP 2 with a working Threads module you can program.
- By examining the code, identify the type of thread scheduling implemented.
- Modify the code to collect the *mean* and *variance* of the following measures of thread management:
 1. *Start time* — the time from when a thread is created to the time it first runs.
 2. *Execution time ratio* — the ratio of *run time* to *elapsed time* of each thread.
 3. *Turnaround time* — the time from when a thread is created to the time it is killed.

If simulation parameters are not changed between runs, results should be comparable across different runs.

Note that not every thread gets killed before the end of the simulation. Make sure you only gather statistics for threads that have gone through the complete life cycle.

You can get the simulated current time, in clock “ticks”, with the call: `HClock.get()`, which returns a `long`.

¹There's a slight error in the `make` instructions. The correct command to run OSP2 GUI from the makefile is:
`$ make gui`
and for the text edition use:
`$ make run`