

Real Time Systems

Assignment 2

Instructions

- Deadline is April 10th 8:00. Deadline is strict.
- Assignment must be delivered through IS (Homework Vaults) as a single zip or tar file (no rar please), file name: yourlogin.zip or yourlogin.tar.
- You must work independently.
- The zip/tar file should contain only commented source codes (no binaries, project files, etc.) and brief documentation (plain txt or pdf file). If some task is not completely solved, the documentation should explicitly mention all “known bugs”.

The assignment is basically a continuation of the programming exercises from the lab session. Nevertheless, if you want, you can do the assignment in a different programming language (e.g., Java).

Tasks

Task 1. *Mutual exclusion protocols.* Implement several variants of mutual exclusion protocols:

- wrong attempts from slides (first, second, third),
- correct algorithms: Peterson’s algorithm, Lamport’s algorithm (for at least three processes),
- solution using semaphores (`pthread_mutex_lock` in case of C+POSIX).

The implementation should be done with the use of **one** parametrized function for description of a process (as opposed to the provided example, which uses a separate function for each process).

Task 2. *Accumulating time drift.* Implement a simple cyclic executive which does some periodic computation and simple waiting (you can use the provided code). Measure the accumulating drift under different conditions, for example:

- different hardware or operating system,
- different “background” load of the computer,
- several parallel task of the same type (as processes, as threads).

Do the measurements repeatedly and focus not just on the average results, but also on the variance. Provide summary and interpretation of your measurements.

Implement a solution without an accumulating drift, e.g., by use of timers or simply by measuring the current time and delaying for the remaining time.