

Sistemas de tempo real / Real-time Systems – 2020/2021

Practical Assignment #1

Scheduling, Priorities Assignment, and Measurement of Computation Times

Introduction

The work to be performed consists on the implementation of an application that makes use of POSIX routines, with the goal of studying the scheduling by priorities assignment and the measurement of the computation times. Consider the system is preemptive. There are three functions, `f1(class,group)`, `f2(class,group)` and `f3(class,group)` each assigned to one of three distinct periodic tasks. These functions have as input parameters two integer numbers that identify the class (1, 2 or 3, corresponding to PL1, PL2 or PL3) and the group. The corresponding code is in the distributed files `func.h` and `func.o`, and must be compiled with the application developed.

This work should be performed in POSIX [1], [2] with Real-Time Extension [1], [2]. The kernels of recent distributions largely implement the real-time part of the POSIX standard. If, and only if, some part of the work may not be strictly performed with the POSIX functionalities, then for that part of the work it can also be used functions of the "Official Linux" [4], [5] (for example, functions of the Linux library).

All the software in this practical assignment should be developed/configured to run in only one processor. Even if the work is developed in a system with only one processor, the software should be developed in a manner that it executes in only one processor if later that software is compiled and executed in a computer with multiple processors.

The work should be preferably developed in a 64 bits Linux architecture. Alternately, it can be developed in a 32 bits architecture, but should then be able to be correctly compiled, run, and tested in a 64 bits architecture.

Description

The items of the work are as follows:

1. Using the POSIX functionalities that you may consider necessary, measure the (maximum) computation time of each of the three given functions. The given functions are expected to run within a very short time variation (small execution ripple).
2. Using the results of 1 verify if the system is schedulable, for two priority orderings: RMPO (rate monotonic priority ordering) and its inverse. The three tasks should be periodically activated with periods of 0.1 [s], 0.2 [s] and 0.3 [s], respectively. Use Gantt charts and the Audsley method. Present in the report all the calculations and charts done.

3. Write the code of an application that, during about 6 seconds, maintains the three tasks jointly running using RMPO. The tasks 1 to 3 should be periodically activated with periods of 0.1 [s], 0.2 [s] and 0.3 [s], respectively. The application should show if the tasks have met the deadlines. To do so, prior to the activation of each task and after its execution, a log table should be filled with those time instants, being printed in the end of the application. In the end, the application should also show the largest response times for each task and the response time jitter.
4. ALTERNATIVE A: Change the code developed in 3 in order to change the priorities (between RMPO and the inverse) during the execution of the tasks at instant $t=1.95$ s and at $t=3.95$ s. Comment the obtained results.

ALTERNATIVE B: Change the code developed in 3 in order to run in any number of processors (CPU cores). Try also to inverse the priorities (to inverse RMPO priorities). Comment the obtained results, namely the largest response times for each task and if they met the deadlines.
5. Develop a source code `func2.c` of a module `func2.o` that imitates the functioning of the module `func.o`. Test the module `func2.o` using it jointly with the created application. For this purpose, you should in particular repeat the work of item 3 above, using `func2.o` in substitution of `func.o`. Comment the obtained results.
6. Assigning equal priorities to the tasks, test the application developed in 3 with the Round Robin scheduling method. Comment the results, namely the largest response times for each task and the response time jitter.

Resources on the Internet (POSIX and Linux Documentation)

- **POSIX Specification:** <http://www.opengroup.org/onlinepubs/9699919799/>
- **POSIX and its Real-Time Extension:** Consult and study the references, investigating what are the parts of POSIX which are most important for the POSIX part of the work. Some topics to study: pthreads, their attributes (e.g. priorities) and scheduling options [1], functionalities for measurement and management of time. Consult [6], including the example in C. Consult the POSIX specification [1], including the following parts:
 - System Interfaces / General Information / Realtime;
 - Topic / Realtime;
 - System Interfaces / General Information / Threads.
- **Linux Manpages:** <http://www.kernel.org/doc/man-pages/>
The pages of the Linux manual may have relevance to the development of the work. Section ``2: System calls'' may have particular relevance.



Hints

Use of pthreads

Some relevant functions:

```
pthread_create()  
pthread_join()  
pthread_exit()  
mlockall()
```

Compilation flags: `-D_REENTRANT -lpthread`

Compilation with the POSIX real-time extension

Compilation flags: `-lrt`

Report and Material to Deliver

A report should be delivered in PDF format. The report should contain the following information in the first page: name of the course ("disciplina"), title and number of the practical assignment, names of the students, number of the class ("turma"), number of the group. It should be submitted through the Nónio system (<https://inforestudante.uc.pt/>) area of the "Sistemas de Tempo Real" course in a single file in zip format that should contain all the relevant files that have been involved in the realisation of the practical assignment. The name of the submitted zip file should be "tXgYrZ-str2020-21.zip", where "X", "Y" and "Z" are characters that represent the numbers of the class ("turma"), group, and practical assignment, respectively. The report should have references and bibliography.

References

- [1] The IEEE and The Open Group. [POSIX Specification] The Open Group Base Specifications Issue 7; IEEE Std 1003.1™-2008, 2008. [Online]. Available: <http://www.opengroup.org/onlinepubs/9699919799/>.
- [2] Wikipedia. POSIX. [Online]. Available: <http://en.wikipedia.org/wiki/POSIX>.
- [3] The Realtime Extension: A White Paper from the X/Open Base Working Group, 1997. [Online]. Available: <http://www.unix.org/version2/whatsnew/realtime.html>.

- [4]The Linux Man-Pages Project. The Linux Man-Pages Project. [Online]. Available:
<http://www.kernel.org/doc/man-pages/> .
- [5]Linux Kernel Documentation. Linux Kernel Documentation Index. [Online]. Available:
<http://www.kernel.org/doc/> .
- [6][Linux] RT PREEMPT HOWTO. [Online]. Available:
http://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO .