INFO-F-404 : Real-Time Operating Systems

2013 – 2014 Project 1: Least Laxity First

1 Main goal

Study the performance of LLF scheduling algorithm on systems with periodic, synchronous tasks and constrained deadlines.
We will consider systems of n periodic, synchronous and independent tasks $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ with constrained deadlines embedded on a uniprocessor devices.

2 Project details

For this project, we ask you to do the following steps:

1. Implement a descrete LLF scheduler and a simulator that can simulate LLF scheduling of a system (given as a parameter in a file) in its study interval $I = [0, lcm\{T_i \mid i = 1, 2, \ldots n\}]$. Priorities of jobs should be recalculated each Δ_R units of time, Δ_R given as an input to your simulator.

   You should be able to execute your program using the following command line:

   `./simLLF <deltaR> <tasksFile>`

   for example:

   `./simLLF 10 tasks.txt`

   In this case $\Delta_R = 10$ and the system of tasks is described in the file `tasks.txt`

2. Implement a generator of random periodic, synchronous systems with constrained deadlines. This generator should be able to generate a system with given parameters (utilization factor and number of tasks), for example:

   `./taskGenerator -u 70 -n 8 -o tasks.txt`

   have to generate a file `tasks.txt` that describes a system of 8 tasks, the utilization of this system is 70% (utilization of the generated system does not have to be equal to 70%, but should be very close to it).

3. Implement a program (`LLF_study`) to test LLF's performances. You should wisely use modules that you've already implemented.

   Comparison tests: number of preemptions and schedulability depending on the total utilization of the system, number of tasks in the system as well as depending on the value of Δ_R.

4. Write a short report that contains:

   (a) short description of your code (diagrams) and implementation choices,

(b) a section where you describe difficulties that you met during this project (and solutions that you found),

(c) a section that describe and discuss the result of your simulations (tables, graphics)[1].

All programs (`simLLF`, `taskGenerator` and `LLF_study`) have to be written in *C++* programming language.

Program `simLLF` have to output following information: study interval I as well as number of preemptions and the total idle time during the study interval. Program `simLLF` should also inform the user in case if the system is not schedulable.

We ask you to use the following file format to describe systems of tasks. Each line of a file describes one task and contains: `Offset Period Deadline WCET`.

Here is an example:

```
0 50 50 30
0 100 70 30
0 80 60 10
```

3 Bonus part

This part is not mandatory, but it is worth some bonus points (only if all other parts work).

Simulator (`simLLF`) have to generate a visual output of the scheduling. You may choose the format (pdf, png, bmp, avi, etc.). Here are some sources that might be helpful:

- PNGwriter `http://pngwriter.sourceforge.net/main-en.php`
- LaTeXGraphics `http://en.wikibooks.org/wiki/LaTeX/Creating_Graphics`

4 Submission and planning

This project should be done in groups of 2, you may choose your partner. This project has to be submitted before 23:59:59 o'clock on October 28 of 2013. Your project has to work properly (compile and execute) under *Linux* in rooms NO3.007, NO4.008 and NO4.009.
To submit (in a *zip* file) a folder that contains at least following files:

- all your C++ sources,
- a Makefile (that creates all executable files),
- A short report (pdf format, about 4-5 pages plus an appendix, figures and graphics are welcom).

The zip file with your project has to be send by e-mail to **nikita.veshchikov@ulb.ac.be** with subject "INFO-F-404 project1". The name of the folder and of the containing zip file : if Chuck Norris makes his project with Michael Jackson they should send a file named *norris-jackson-project1.zip*.

Good luck!

[1] very important part of the report!