Tutorial 1: RT Services on Linux

Authors

- David José Araújo Ferreira - 93444
- Tiago Adónis Hipólito Dias - 88896

Q1 - The improvements observed in A1 with respect to the execution of the task using the standard Linux scheduling services.

By default, the minimum and maximum *Inter-Arrival-Time* suffer a **considerable drop** and **rise**, respectively, after the video load initiates, having a minimum of **98147.07 ms** and a maximum of **101849.664 ms**.
When the task is set with a **fixed priority of 58**, this time the *Inter-Arrival-Times* perverse some stability, with **99263.133 ms** as a minimum and **100734.844 ms** as a maximum.

This difference is because in **non-RT services** the default Linux scheduler is being utilized as opposed to the **scheduler of type FIFO** being utilized in the RT service, which prioritizes tasks with fixed priorities and that have been initiated first.

Q2 - The impact of priorities in Assignment A3.

By initiating 3 parallel processes, even with different priorities and periodicities, we can see that non of them deviated significantly from the expected time values of the *Inter-Arrival-Times*. This can be explained by the fact that being processed in an 4-core CPU, **the processes were distributed by the cores** in a way that has not affected each process.

By limiting the processes to just one core, the priority of each task will have a greater effect on the stability of the time, since tasks will need to be scheduled between them to be processed, with bigger priorities tasks showing a smaller time deviation.

In the table below, is possible to check the experiment results. Notice that all processes have a periodicity of 90.

	proc1	proc2	proc3
priority	30	60	90
min	26680.146	32670.758	37486.707
max	965439.312	360775.688	142507.938