

Guião 1 : RT Services on Linux

João Martins - 93183, Raquel Pinto - 92948

A1

Observou-se que ao usar Real Time based scheduling que os tempos de *inter-arrival* mantêm-se próximos do período (100ms) mesmo colocando o sistema sob alguma pressão ao fazer playback de vídeos em 4k do *youtube*, no entanto, ao usar o scheduling standard do linux os tempos de *inter-arrival* sofrem de desvios consideráveis do período, pois está a competir com todas as outras threads do sistema ao contrário da *thread* em *real time* que tem prioridade sobre estas.

Programa	Período/ms	Min/ms	Max/ms
Tarefa A0 (pt)	100	96,718	113,286
Tarefa A1 (pt_a1)	100	99,578	100,419

A2

Observou-se que a diferença entre o período configurado e o real não é elevada (consultar a1), ao fazer o playback de um vídeo de 4k, este várias vezes parava por breves instantes. No entanto, há uma relação inversa entre prioridade e desvio, quanto menor a prioridade maior o desvio, sendo este o comportamento esperado da política de scheduling *real time* em linux.

Desvios de +- 1ms.

A3

O programa A3 foi alterado a partir do A2 de modo que as tarefas fossem executadas no CPU 0. Dos resultados voltou-se a retirar a relação inversa entre prioridade e desvio, sendo estes bem maiores que no programa A2 e bastante menor na thread de prioridade 99 do que na thread de prioridade 23, algo esperado pois o escalonador dá prioridade a *threads* com prioridades mais altas, adormecendo as threads com prioridades mais baixas que estão a ser executadas. No caso da menor prioridade (19) a thread não chegou a fazer *output* de qualquer resultado, *starvation*.

Prioridade	Periodicidade	Min/ms	Max/ms
99	89	39,517	138,473
66	111	420,39	179,960
55	103	337,75	177,540
55	107	360,29	185,674
46	97	105,92	288,393
35	121	9,829	779,712
23	27	9,797	207,120
19	97	—	—