



# Utilização do Sistema Operativo de Tempo-Real RTAI

## Objetivos:

Tomar contacto com um sistema operativo de tempo-real para PC, no caso o Linux/RTAI.

1. Observar o interface deste RTOS, os modelos de tarefas, os módulos de aplicação RT, as funções de inicialização e terminação da aplicação RT.
2. Desenvolver e executar programas sobre este sistema operativo.

## Procedimentos:

### *1. Análise do código fornecido*

Descarregar o software de apoio à sessão “rtai-class-sources.zip”, disponível na página da disciplina.

1. Observar o código fonte do módulo (“str\_test.c”). Ver as chamadas ao sistema para definição, criação e ativação de tarefas (consulte o manual do RTAI). Analisar a estrutura de uma tarefa.
2. Consulte o ficheiro README. Compile e execute o exemplo fornecido. Observe a execução concorrente das várias tarefas. Observe e explique os efeitos observados.

Nota 1: Sendo a escrita no ecrã feita por um processo Linux, não se garante que ocorra nos instantes precisos em que as tarefas são ativadas e terminam. Contudo, a informação do estado relativo das tarefas no momento da escrita vem correta do RTAI, pelo que se mantém correta no Linux, embora que possivelmente um pouco atrasada.

Nota 2: Como pode observar no código fonte do módulo, os instantes de início e terminação das tarefas são também escritos para o “kernel ring buffer” (função “printk”). Este buffer pode ser consultado através do comando “dmesg”. Pode desta forma verificar a sequência e instantes de ativação das tarefas.



## ***2. Tarefas a realizar:***

Realize as tarefas abaixo indicadas. Documente todas as alterações ao código que efetuar. Deverá entregar um relatório (2 páginas, máximo) com a análise dos resultados observados e indicação das alterações ao software introduzidas.

1. Inverta a prioridade das tarefas. Observe o padrão de ativações resultantes e anote eventuais diferenças relevantes, justificando-as.
2. Reponha as prioridades originais. Modifique o código fonte por forma a que a tarefa 0 seja atrasada de 100ms em relação às outras tarefas. Observe o padrão de ativações resultantes e anote eventuais diferenças relevantes, justificando-as.
3. Adapte a aplicação por forma a que seja calculado o tempo que decorre entre ativações consecutivas. Este valor deve ser enviado para o “user space” por meio de “shared memory”. Construa um programa em “user space” que mostre estes valores no ecrã.
4. Em muitas aplicações estrutura-se o software em diversos módulos/tarefas, cuja execução deve ser encadeada, garantindo-se relações de precedência. Modifique o software fornecido por forma a que a tarefa 0 se mantenha periódica, sendo posteriormente ativadas cada uma das restantes tarefas, por ordem numérica ascendente, cada uma apenas após a conclusão da tarefa anterior.