



Utilização do Sistema Operativo de Tempo-Real Xenomai

Objetivos:

Tomar contacto com um sistema operativo de tempo-real para PC, no caso o Xenomai

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 “xenomai-class-sources.zip”, disponível na página da disciplina.

1. Observar o código fonte do módulo (“sample_task.c”). Ver as chamadas ao sistema para definição, criação e ativação de tarefas (consulte a API do Xenomai). Analisar a estrutura de uma tarefa.
2. Compile e execute o exemplo fornecido (note que existe uma makefile). Observe a execução da tarefa periódica.

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. Modifique a tarefa para que seja mostrado o tempo máximo e mínimo entre activações. Execute novamente a tarefa e abra outros processos enquanto esta está em execução. Compare os resultados com os obtidos usando as extensões tempo-real do Linux.



2. Modifique a aplicação para que esta apresente três tarefas com prioridades distintas. Faça com que estas tarefas partilhem o mesmo *core*. Teste várias ordenações de prioridades e observe o padrão de ativações resultantes. Anote eventuais diferenças relevantes, justificando-as.
3. 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 uma das tarefas se mantenha periódica, sendo posteriormente ativadas cada uma das restantes tarefas, após a conclusão da tarefa anterior. Cada uma das tarefas deve enviar dados à tarefa seguinte da cadeia. Deve ser usado um número inteiro sequencial, para facilitar a verificação do funcionamento.
 1. Sugestão: usar semáforos (ver `rt_sem_create()`, `rt_sem_p()`, `rt_sem_v()`) combinados com “shared memory” (uma simples variável global, neste caso).
4. Existem mecanismos de comunicação entre processos que permitem a comunicação entre os domínios tempo-real e o “user-space”. Use um destes mecanismos para que cada tarefa reporte o início e terminação de execução de cada instância. Construa depois um programa em “user-space” que leia esta informação e a mostre num terminal Linux.