Trabalho prático 1

Serviços tempo-real no sistema operativo de uso geral Linux

Objectivos:

1. Processos periódicos e observação do seu comportamento temporal no SO Linux
2. Utilização de prioridades tempo-real no Linux

Procedimentos:

Realize as tarefas abaixo indicadas. Documente todas as alterações ao código que efectuar. Deverá entregar um relatório (1 página, máximo) com a análise dos resultados observados e indicação das alterações ao software introduzidas.

Utilizar o programa "periodic_task.c", disponível na página da disciplina.

a) Observar o seu conteúdo. Reparar na técnica utilizada para gerar uma activação periódica recorrendo a "itimers".

b) Lançar outros processos em terminais diferentes e observar o comportamento. Use em particular processos que gerem operações de IO intensivas e carga computacional elevada, como por exemplo compactar uma pasta com o comando "tar" e/ou observar vídeos (e.g. do youtube). Repita várias vezes as operações e anote os valores observados.

c) O Linux dispõe de escalonamento tempo-real. Um processo pode ser colocado nesta classe de prioridades por meio da instrução "sched_setscheduler". Modifique o programa fornecido por forma a que este receba prioridade tempo-real. Repita as medidas efectuadas na alínea anterior. Analise os resultados.

   Nota: a execução de programas que usem prioridades tempo-real requerem prioridades de super-utilizador. O comando "sudo" permite executar um programa com este tipo de privilégios.

d) Modifique o programa por forma a que a prioridades possa ser indicada na linha de comando. Execute diversas instâncias em janelas separadas com prioridades diferentes. Analise os resultados.

Nota: Para facilitar a observação do padrão de interferência em computadores com vários processadores é necessário fixar os processos a um mesmo CPU. Tal pode ser feito com o seguinte trecho de código, executado durante a fase de inicialização do programa:

```
/* Variáveis*/
cpu_set_t cpuset;
...
/* Força o processo a correr no CPU0 */
CPU_ZERO(&cpuset);
CPU_SET(0,&cpuset);
if(sched_setaffinity(0, sizeof(cpuset), &cpuset)) {
    printf("\n Lock do processo ao CPU0 falhou!!!");
    return(1);
}
...
```