

Laboratórios de Sistemas Operativos

Tutorial #4: Programação com processos e tarefas

Os tutoriais práticos de SO consistem num conjunto de exercícios práticos que permitem aos alunos familiarizarem-se com um determinado tema que será necessário para resolver os projetos da disciplina. Os tutoriais podem ser resolvidos individualmente ou em grupo. A sua resolução é recomendada mas não obrigatória. Não são avaliados.

Cada tutorial pressupõe que os exercícios são realizados numa interface de linha de comandos (shell) de um sistema Unix/Linux ou equivalente. Assume também que os alunos já resolveram os tutoriais anteriores.

Este guião pretende ajudar os alunos a ambientarem-se com a programação paralela com tarefas, que é um dos requisitos essenciais do 1º exercício do projeto de SO.

Este exercício não será avaliado.

1. Descarregue o ficheiro `thread.zip` (a partir do site de SO no Fénix) para a sua área de trabalho e descomprima o seu conteúdo:
`unzip thread.zip`
 - a) Estude, compile e execute a aplicação. Consulte o diagrama na página seguinte deste guião e compreenda o resultado do programa.
 - b) Por que razão é indeterminado o conteúdo da variável global `Value`?
 - c) Monte uma experiência que prove a afirmação da alínea b). *Sugestão: utilize a chamada sistema `sleep`.*
2. Componha um programa paralelo alternativo, em que as chamadas a `pthread_create` e `pthread_join` são substituídas por chamadas a `fork` e `wait` (tenha o cuidado de assegurar que a função `wait` só é chamada pelo processo pai).
 - a) Neste caso, o paralelismo acontecerá com processos distintos, e não com tarefas dentro do mesmo processo. O que muda no resultado que este novo programa imprime no `stdout`?

Tarefas

Criação de Tarefas

Código Fonte

```
int Value = 0;

void* thr_func(void* ptr)
{
    Value = 1;
    return NULL;
}

void main()
{
    int tid;
    pthread_create(&tid, NULL, thr_func, NULL);
    Value = 2;
    pthread_join(tid, NULL);
    printf("Value=%d\n", Value);
}
```

Output da Execução

Value=?

