

# Enunciado do Projeto de Sistemas Operativos 2015-16

## SHELL PARALELA - EXERCÍCIO 3

LEIC-A / LEIC-T / LETI  
IST

### 1 Shell paralela com limitação de paralelismo e monitorização melhorada

O 3º exercício baseia-se na implementação de 2 requisitos, descritos de seguida.

Para este exercício devem ser usadas as seguintes funções da API do Unix/Linux: `sem_wait`, `sem_post`, `sem_init` e `sem_destroy`. Recomenda-se fortemente que a solução se baseie nas soluções estudadas nas aulas teóricas para problemas clássicos de sincronização.

**O desenvolvimento da solução deste exercício deve assentar no código que entregaram anteriormente (ou solução publicada pelo corpo docente caso optem por isso), sem a modificação da alínea extra realizada durante a aula de avaliação.**

#### 1.1 Limitação do número de processos filho em execução simultânea

O utilizador menos cauteloso pode facilmente lançar um conjunto de comandos para execução paralela que excederá o paralelismo suportado pela máquina. Esse tipo de situações podem levar a interferências entre os processos (e.g. comutações de processos frequentes) que podem prejudicar o desempenho. Por exemplo, numa máquina quad-core, caso o utilizador lance 8 processos CPU-intensivos, o desempenho provavelmente será pior do que se executar primeiro 4 desses processos em paralelo e só depois executar os restantes 4.

Este exercício pretende estender a par-shell desenvolvida até ao momento por forma a limitar o número de processos filho em execução simultânea, para ajudar a evitar os cenários descritos anteriormente.

Deverá haver uma constante *MAXPAR*, de valor positivo, que determina o número máximo de processos filho que a par-shell pode ter a correr a cada instante. À medida que o utilizador pede a execução de um novo processo filho, este só deverá ser lançado de imediato caso o número de processos filho ativos não exceda *MAXPAR*. Caso contrário, a tarefa principal deve esperar até que algum processo filho ativo termine para lançar o novo processo filho.

#### 1.2 Remoção da espera por ciclos de sleep

No **exercício 2** do projeto par-shell, foi pedido para criar uma tarefa monitora que consiste num ciclo infinito que, quando não existem processos filho em execução (consultando o valor do inteiro *numChildren*), espera 1 segundo antes da próxima iteração (através da função `sleep`).

Para tornar a par-shell mais eficiente, altere a tarefa monitora por forma a que esta deixe de esperar por recurso ao `sleep`. Em alternativa, a tarefa monitora deverá passar a esperar pela existência de novos processos filho usando um semáforo partilhado com a tarefa principal.

## 2 Experimente

Pode testar o funcionamento da nova par-shell recorrendo ao programa fibonacci fornecido no primeiro exercício. Experimente lançar o programa com *valores elevados* para o fibonacci (por exemplo, valores à volta de 45000), com vários comandos na mesma instância da par-shell.

Experimente com valores crescentes de *MAXPAR* até exceder o número de CPUs da máquina que está a usar. Compare os tempos de execução para ganhar noção do impacto nos tempos de execução dos processos filho resultante de exceder o paralelismo suportado pela máquina. Relacione com a matéria dada na aula teórica sobre as noções de paralelismo.

## 3 Entrega e avaliação

Os alunos devem submeter um ficheiro no formato zip com o código fonte, uma Makefile, e um executável através do sistema Fénix. O exercício deve obrigatoriamente compilar e executar nos computadores dos laboratórios.

A data limite para a entrega do exercício é 30 de Outubro até às 23h59m do dia indicado.

Após a entrega de cada um dos exercícios, será disponibilizado pelo corpo docente, a codificação da respectiva solução que pode ser usada pelos alunos para desenvolverem os exercícios seguintes.

A demonstração da solução do exercício entregue pelos alunos é feita em simultâneo com a demonstração da solução desenvolvida para o 2º exercício e acontece durante a aula laboratorial da **semana de 2-6 Novembro**. No início dessa aula laboratorial, será dada aos alunos uma alínea adicional que complementa os enunciados dos 2º e 3º exercício. A alínea adicional é de resolução rápida para quem preparou e resolveu os enunciados base.

Cada grupo tem até ao fim da aula para concluir a alínea adicional e mostrar o exercício completo (exercício base + alínea dada na aula) ao docente. Todos os membros do grupo terão que estar preparados para explicar a solução apresentada.

A nota é individual a cada membro do grupo. Membros que não compareçam na aula de demonstração têm nota nula neste exercício.

**É imperativo que todos os elementos do grupo sejam pontuais na aula laboratorial na qual farão a demonstração.**

## 4 Cooperação entre grupos

Os alunos são livres de discutir com outros colegas soluções alternativas para o exercício. No entanto, *em caso algum* os alunos podem copiar ou deixar copiar o código do exercício. Caso duas soluções sejam cópias, ambos os grupos reprovarão à disciplina.