

# Enunciado do Projecto de Sistemas Operativos 2014-15

## ESCRITOR-LEITOR - EXERCÍCIO 1

LEIC-A / LEIC-T / LETI  
IST

### Resumo

Este documento apresenta uma visão global do projecto de SO 2014/15, que está organizado em 5 exercícios encadeados, assim como uma descrição pormenorizada do 1º exercício.

O 1º exercício consiste em desenvolver dois programas: um que escreve em vários ficheiros, outro que lê e confere o conteúdo de um ficheiro.

## 1 Introdução

O projecto de Sistemas Operativos está organizado em 5 exercícios encadeados. Esta secção introdutória apresenta uma visão global do projecto e dos tópicos abordados em cada um dos exercícios. As secções seguintes deste documento descrevem apenas o 1º exercício. Serão disponibilizados documentos similares para cada um dos exercícios seguintes.

Globalmente o projecto consiste em desenvolver um programa denominado ESCRITOR-LEITOR que, tal como o nome indica, executa operações de escrita e de leitura em ficheiros.

Os 5 exercícios que constituem o projecto abordam os tópicos seguintes:

- Exercício 1 - Versão sequencial dos programas ESCRITOR e LEITOR.
- Exercício 2 - Versão paralela do programa ESCRITOR; um processo pai cria vários processos filhos e serão estes que, em concorrência, executam as operações de escrita nos ficheiros, cabendo ao processo pai medir e imprimir o tempo total da execução do programa.
- Exercício 3 - Nesta fase será necessário sincronizar o acesso aos ficheiros por parte dos programas ESCRITOR; com efeito, pretende-se assegurar que, se um processo está a escrever num dado ficheiro, nenhum outro processo pode estar a escrever nesse mesmo ficheiro. Vários processos LEITOR poderão ler os mesmos ficheiros concorrentemente.
- Exercício 4 - Nesta fase pretende-se que os processos LEITOR sejam capazes de verificar a coerência de 3 ficheiros de forma concorrente. Para tal, serão desenvolvidas duas versões: i) uma em que existe um programa pai que cria 3 processos filho (cada um dos quais fará uma verificação), e ii) outra em que serão criados 3 fios de execução (*threads*) para realizar a mesma tarefa.

- Exercício 5 - Nesta fase pretende-se que seja desenvolvida novas versões do LEITOR (com paralelismo) com sincronização mais complexa entre as várias *threads* no acesso a um *buffer* partilhado que contém os nomes dos ficheiros lidos do teclado (do tipo escritores-leitores) e um processo monitor que gere as *threads* antes mencionadas.

## 2 Programa ESCRITOR

No 1º exercício, os alunos deverão codificar um programa ESCRITOR que realiza a seguinte funcionalidade: o programa abre um ficheiro, escolhido de forma aleatória entre um conjunto de 5 ficheiros pré-definidos, e escreve 1024 vezes uma cadeia de 10 caracteres, escolhida também aleatoriamente entre 10 cadeias de caracteres distintas pré-definidas. Este processo deve ser repetido, em ciclo, 5120 vezes. Quando o programa escreve num ficheiro, substitui o conteúdo anterior do mesmo.

Os nomes dos ficheiros acedidos por este programa devem ser necessariamente os seguintes: “S02014-0.txt”, “S02014-1.txt”, ..., “S02014-4.txt”.

As cadeias de caracteres a imprimir devem ser também necessariamente as seguintes:

- “aaaaaaaaa\n”,
- “bbbbbbbb\n”,
- .....
- “jjjjjjjj\n”.

Ao tentar abrir cada ficheiro, caso o mesmo não exista, o programa ESCRITOR deverá criá-lo. As permissões do ficheiro criado devem ser de leitura e escrita para o utilizador que executa o processo e de leitura apenas para os restantes utilizadores da máquina.

## 3 Programa LEITOR

Os alunos deverão codificar um programa LEITOR que realiza a seguinte funcionalidade: o programa abre um ficheiro, escolhido de forma aleatória entre um conjunto de 5 ficheiros pré-definidos e verifica se este ficheiro contém 1024 repetições da mesma cadeia de 10 caracteres. Deve devolver 0 se o ficheiro estiver correcto e -1 caso detecte uma incoerência. O programa LEITOR deve fazer isto exactamente uma única vez.

Os nomes dos ficheiros acedidos por este programa deverão ser os mesmos usados pelo programa ESCRITOR.

## 4 Funções do Unix

As operações sobre os ficheiros devem ser efectuadas usando as funções Unix. Mais informação sobre a sua utilização podem/devem ser obtidas consultando o respectivo manual.

```
int open(const char *pathname, int flags);
int close(int fd);
ssize_t write(int fd, const void *buf, size_t count);
ssize_t read(int fd, void *buf, size_t count);
```

## 5 Experimente

Experimente executar o ESCRITOR e o LEITOR concorrentemente. O programa leitor detecta incoerências no ficheiro?

Experimente executar vários escritores concorrentemente. No final existem ficheiros incoerentes?

Como explica o que observou em ambas as experiências?

## 6 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 de cada um dos exercícios são as seguintes, até às 23h59m do dia indicado (indica-se também a data de disponibilização para os alunos dos enunciados de cada um dos exercícios entre parêntesis):

- Exercício 1 - 10 de Outubro de 2014 (28 de Set.)
- Exercício 2 - 24 de Outubro de 2014 (12 de Out.)
- Exercício 3 - 31 de Outubro de 2014 (26 de Out.)
- Exercício 4 - 14 de Novembro de 2014 (2 de Nov.)
- Exercício 5 - 5 de Dezembro de 2014 (16 de Nov.)

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 do exercício acontece durante a primeira aula laboratorial de cada grupo que se segue à data de entrega. No início dessa aula laboratorial, será dada aos alunos uma alínea adicional que complementa o enunciado apresentado neste documento. A alínea adicional é de resolução rápida para quem preparou e resolveu o enunciado 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.

## 7 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.