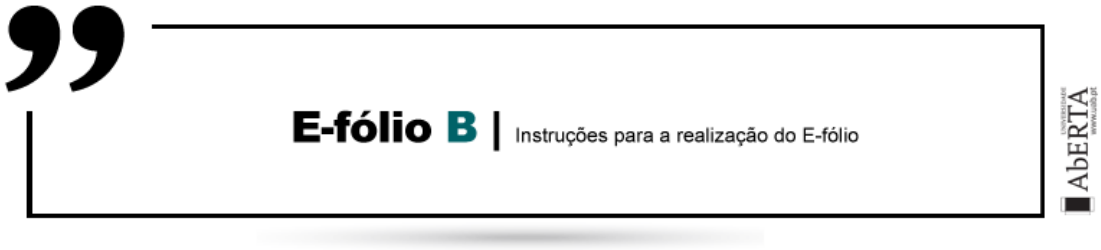
Sistemas Operativos

(ano letivo 2018-19)



Este enunciado constitui o elemento de avaliação designado por "e-fólio B" no âmbito da avaliação contínua e tem a cotação total de 5 valores. A sua resolução deve ser entregue até às 23h55 do dia 26 de maio pelos alunos que escolheram a modalidade de avaliação contínua.

A resolução deve ser entregue através de um único ficheiro compactado .zip, que:

(i) contém os ficheiros .c que constituem o código dos programas, prontos a serem compilados;

(ii) contém um ficheiro de nome relatorio.pdf com um relatório simples e sucinto com informações solicitadas e/ou complementares de modo a permitir uma fácil compreensão do trabalho realizado. É desnecessário incluir uma listagem integral do código.

(iii) O nome do ficheiro .zip a entregar deve seguir a seguinte convenção para o seu nome,

"NumeroAluno-PrimeiroNome-Apelido-21111-efB.zip"

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, "327555-Paulo-Costa-21111-efB.zip"

O ficheiro deve ser única e exclusivamente entregue através do recurso "E-fólio B" disponibilizado na plataforma (Nota: apenas é visível para os alunos inscritos em avaliação contínua), não sendo aceites trabalhos enviados por outras vias, como por exemplo por e-mail.

Esta é uma prova de avaliação **individual** e não "um trabalho de grupo". A sua resolução deve provir unicamente do conhecimento adquirido e trabalho original desenvolvido pelo próprio aluno. Os alunos deverão saber distinguir claramente entre discutir os conteúdos abordados na unidade curricular (permitido) e discutir a resolução específica do e-fólio (não permitido).

No caso de dúvidas de interpretação do enunciado, utilize o fórum de avaliação para pedidos de esclarecimento.

I

1. [5] Escreva um programa multitarefa em linguagem C padrão e segundo a norma POSIX, de nome `mtss.c`, constituído pela tarefa principal, por nt tarefas designadas tarefas calculadoras e 1 tarefa designada tarefa somadora, num total de nt+2 tarefas. O objetivo do programa é calcular a soma s(n) do quadrado dos primeiros n números naturais,

$$s(n) = \sum_{i=1}^{n} i^2$$

com as tarefas calculadoras dedicadas ao cálculo de somas parciais correspondentes a blocos do somatório, a tarefa somadora a somar os resultados parciais obtidos e a tarefa principal a imprimir o resultado final.

- O programa mtss recebe obrigatoriamente 3 argumentos na linha de comandos,

```
>> ./mtss nt n nbloco
```

- `nt` é o nº de tarefas calculadoras, com nt≥1.

- `n` é o nº de termos total do somatório (ou de números naturais), com $1 \leq n \leq 999$.

- `nbloco` é o nº de termos de cada bloco do somatório, com $1 \leq$ nbloco.

- O programa mtss deve testar se o número de argumentos dado na linha de comandos é correto e se os seus valores são válidos. Em caso de erro o programa deve emitir uma mensagem e terminar.

- No início do programa, a tarefa principal (main) deve imprimir uma mensagem do tipo "Soma do quadrado dos primeiros n numeros naturais com nt tarefas e blocos com nblocos termos".

- Quando criadas, as nt tarefas calculadoras devem receber respetivamente como argumento o seu número de ordem, entre 0 e nt-1.

- Cada tarefa calculadora deve tentar adquirir um bloco de cada vez para processamento, sendo os blocos no geral atribuídos sequencialmente às tarefas até não haver mais termos. Após adquirir um bloco a tarefa deve calcular a soma parcial correspondente e colocar o resultado num buffer de dimensão nt destinado a somas parciais. Após a colocação do resultado parcial no buffer ou no caso de este estar cheio, a tarefa calculadora deve invocar a função sched_yield() que liberta o processador (ver man page) dando oportunidade às outras tarefas de serem escalonadas para execução. De seguida a tarefa deve (conforme o caso) voltar a tentar colocar o resultado parcial no buffer e/ou tentar adquirir novo bloco para processamento e assim sucessivamente até não haver mais termos do somatório, caso em que as tarefas calculadoras terminam.

- Note que n não é necessariamente um múltiplo inteiro da dimensão do bloco nbloco. Note também que para o cálculo de uma soma parcial apenas é necessário saber o índice inicial (ou final) e a dimensão do bloco.

- Imediatamente antes de terminar, cada tarefa calculadora deve imprimir uma mensagem do tipo "Tarefa (%d) somou %d elementos" onde os %d representam respetivamente o nº de ordem da tarefa e o total de termos do somatório que a tarefa somou.

- A tarefa somadora tem a função de manter uma variável com o valor da soma total acumulada, devendo cada vez que acede ao buffer retirar e somar todas as somas parciais que se encontrem no buffer deixando-o vazio, após o que deve invocar a função sched_yield() dando oportunidade às tarefas calculadoras de serem escalonadas para execução, antes de tentar aceder ao buffer novamente.

- Conceba uma estratégia para que a tarefa somadora determine quando deve terminar e para que a tarefa principal tenha conhecimento do valor da soma total acumulada.

- Antes de terminar, a tarefa principal, que deve ser a última a terminar, deve imprimir o resultado final assim como o resultado esperado que é dado pela expressão,

$$s(n) = [n(n+1)(2n+1)]/6$$

- Pondere quais as funções da biblioteca pthread que vai utilizar no programa e consulte as respetivas man pages para se informar dos detalhes de funcionamento de cada uma. Pondere também cuidadosamente quais os recursos e as estruturas de dados manipuladas pelas tarefas e que requeiram exclusão mútua no seu acesso para o bom funcionamento do programa.

- Indique no relatório os troços de código correspondentes a regiões críticas do programa e justifique a sua existência/necessidade.

- O programa deve estar identificado com um cabeçalho similar ao seguinte,

```
/*
** UC: 21111 - Sistemas Operativos
** e-fólio B 2018-19 (mtss.c)
**
** Aluno: 327555 - Paulo Costa
*/
```

Critérios de correção:

- Programa desenvolvido difere significativamente das especificações e instruções do enunciado => 0 valores.
- Programa não compila ou produz avisos (warnnings) com `gcc -Wall` => 0 valores.
- Código do programa não está correta e uniformemente indentado de modo a permitir a sua leitura fácil => 0 valores
- Programa não está comentado => 0 valores. Os comentários no programa elucidam questões relevantes do código locais ao comentário.
- Funcionalidade do programa de acordo com o pedido, estrutura, nível de simplicidade e qualidade do código (até 65%)
- Relatório. Explique o como e porquê relativamente às opções e soluções técnicas que tomou para a estrutura e funcionamento do programa (até 35%)

Nota ética: Nunca é de mais referir que o código a apresentar como solução para este e-fólio deve ser 100% original do aluno. A probabilidade de duas pessoas que efetivamente não comunicaram entre si, apresentarem programas "quase iguais" é considerada nula. Isto é válido para qualquer par de alunos (cópia), assim como entre um aluno e qualquer outra pessoa, em particular através da Internet (cópia/plágio), onde existem inúmeras soluções e código para os mais variados problemas, em sites, fóruns, blogs, etc.
Cumpra estritamente as normas de realização individual, como se estivesse num exame com consulta, onde pode consultar a documentação mas não pode falar com ninguém.

FIM