

# Sistemas Operativos

(ano letivo 2022-23)

”

**E-fólio B** | Instruções para a realização do E-fólio



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 16 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 (sem acento) com um relatório 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”, (sem acentos).

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 `mtts.c`, constituído pela tarefa principal e mais `nt` tarefas designadas tarefas trabalhadoras, num total de `nt+1` tarefas. O objetivo do programa é analisar um ficheiro e calcular a frequência de ocorrência em percentagem de cada letra do alfabeto, sendo indiferente minúsculas ou maiúsculas.

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

```
>> ./mtts nt nc ficheiro
```

- `nt` é o nº de tarefas trabalhadoras, com  $nt \geq 1$ .

- `nc` é o nº de caracteres a analisar de cada vez, com  $nc \geq 1$ .

- `ficheiro` é o nome do ficheiro a analisar.

- O programa `mtts` 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 três mensagens em linhas separadas: "Análise de letras do alfabeto com `nt` tarefas e blocos de `nc` caracteres" (substituir valores); "Ficheiro `ficheiro` (dimensão bytes)" (substituir valores); "Relatório de tarefas:".

- Inicialmente, a tarefa principal lê o ficheiro inteiro para um array `buf[]` de caracteres global previamente alocado com a função `malloc()` com a dimensão do ficheiro. Admite-se que a dimensão do ficheiro pode ser guardada numa variável do tipo `int`.

- Ciclicamente, as tarefas trabalhadoras acedem de cada vez a **um** bloco de `nc` caracteres, analisam o seu conteúdo e guardam a frequência das letras (sem acentos) e do total de caracteres “brancos” (espaços, tabs e new lines) num vetor de contadores de inteiros local. Os blocos devem ser acedidos (ou atribuídos às tarefas) sequencialmente, sendo que o primeiro bloco tem início no primeiro carácter de `buf[]`.

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

- Antes (e só antes) de terminar, cada tarefa trabalhadora deve adicionar o resultado das suas contagens a um vetor de contadores de inteiros global e imprimir uma mensagem com o formato “T%d= %7.3f%%\n” com o nº de ordem da tarefa e o total de caracteres do ficheiro que a tarefa analisou em percentagem.

- Antes de terminar, a tarefa principal, que deve ser a última a terminar, deve imprimir uma tabela final com o título "Relatório de caracteres:" e com a frequência de ocorrência das letras, uma por linha, com o formato “%c= %7.3f%%\n” para a letra e respetiva frequência total de ocorrência no ficheiro em percentagem (excluindo dos cálculos os caracteres brancos).

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

- 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.

Dica: Neste programa não é necessário copiar blocos de caracteres, bastando saber o endereço do início do bloco e o número respetivo de caracteres do bloco.

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

```
/*  
** UC: 21111 - Sistemas Operativos  
** e-fólio B 2022-23 (mtts.c)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

Exemplo de saída do programa mtts (abreviada):

```
Análise de letras do alfabeto com 2 tarefas e blocos de 256 caracteres  
Ficheiro ficheiro.txt (2000 bytes)  
Relatório de tarefas:  
T0= 70.000%  
T1= 30.000%  
Relatório de caracteres:  
A= 2.123%  
B= 0.027%  
...  
Z= 0.000%
```

### **Critérios de correção:**

- O programa desenvolvido difere significativamente das especificações e instruções do enunciado => 0 valores.
- O programa não compila ou produz avisos (warnings) com `gcc -Wall` => 0 valores.
- O código do programa não está correta e uniformemente indentado de modo a permitir a sua leitura fácil => 0 valores
- O programa não está comentado => 0 valores. Os comentários no programa elucidam questões relevantes do código locais ao comentário.
- O relatório deve explicar a estrutura e funcionamento global do programa de modo à sua fácil compreensão.
- O programa não funciona corretamente ou não cumpre todas as especificações ou é demasiado complexo => de 0 a 100% valores, sendo o programa avaliado como um todo em conjunto com o relatório e tendo em conta a implementação das características pedidas.

**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