

Sistemas Operativos

(ano letivo 2015-16)

e-fólio B

(correções/atualizações a azul)

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 pdf de formato livre, com um relatório sucinto com informações 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).

I

1. [5] Escreva um programa multitarefa em linguagem C padrão e segundo a norma POSIX, de nome `mth.c`, que calcule um histograma (nº de ocorrências de cada item) dos caracteres ASCII com código 32 a 127 de um ficheiro. Todos os outros caracteres devem ser ignorados. Para efetuar este processamento, o programa deve cumprir as especificações seguintes.

- O programa `mth` recebe obrigatoriamente 2 ou 3 argumentos na linha de comandos,

```
>> ./mth ficheiro nt dimbloco
```

- `ficheiro` é o nome do ficheiro de entrada.

- `nt` é o nº de tarefas a utilizar no processamento (além da tarefa principal).

- `dimbloco` é o nº de caracteres que uma tarefa processa de cada vez. Este argumento é opcional e se omitido deve ser assumido o valor 1024.

- O programa `mth` deve testar se o número de argumentos dado na linha de comandos é correto, se o ficheiro de entrada existe, se `dimbloco` quando especificado é pelo menos 4 e se `nt` é pelo menos 1 e não maior que 64. Em caso de erro o programa deve emitir uma mensagem e terminar.

- O esquema global de funcionamento do programa consiste em cada uma das `nt` tarefas processar blocos de `dimbloco` caracteres e calcular o seu próprio histograma. Uma vez processado todo o ficheiro, as tarefas devem somar os histogramas parciais obtendo o histograma total que deve ser impresso no ecrã pela tarefa principal.

- O programa deve utilizar pelo menos as seguintes variáveis globais,

```
/* variaveis globais */
int htrf[64][96],          /* matriz para histogramas parciais */
    h[96],                /* vetor para histograma total */
    nt;                   /* numero de tarefas */
```

- No início do programa, a tarefa principal (`main`) deve imprimir uma mensagem do tipo “Cálculo de histograma ASCII do ficheiro ‘ficheiro’ com ‘nt’ tarefas”. **Na linha seguinte deve indicar a dimensão do bloco tipo “Bloco=dimbloco caracteres”.**

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

- Cada tarefa deve contar o nº de blocos do ficheiro que processou e imediatamente antes de terminar imprimir uma mensagem do tipo “Tarefa (i) processou m blocos **do ficheiro**” onde “i” é o nº de ordem da tarefa e m o nº blocos.

- Cada tarefa adquire para processamento um bloco de cada vez, após o que deve dar oportunidade a outras tarefas de também adquirirem blocos. Dica: utilize uma variável global comum que controla o nº caracteres já atribuídos para processamento.

- Após **todas as tarefas** terminarem os histogramas parciais, as tarefas devem calcular o histograma total em blocos (cada tarefa um de cada vez) de 16 colunas da matriz htrf, que correspondem a 16 elementos do vetor h. **Cada tarefa que processa um bloco deve imprimir uma mensagem do tipo “Tarefa (i) processou bloco j do histograma”**. Não havendo mais blocos do histograma total para calcular, cada tarefa deve terminar.

- O histograma total deve ser impresso com 8 itens por linha **separados por um espaço**, em que cada item é constituído pelo caracter impresso **entre-pelicas** e a sua frequência num campo de 7 dígitos à direita, separados por um espaço, por exemplo **A 1234567**. Nota: o caracter 127 pode não ser imprimível.

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.

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

```
/*  
** UC: 21111 - Sistemas Operativos  
** e-fólio B 2015-16 (mth.c)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

- Por razões devidas à impressão de relatórios, a inclusão de “print screens” da linha de comandos no relatório só deve ser feita com fundo branco. A cor de fundo do terminal é normalmente alterável na opção “Preferences” do terminal.

Critérios de correção:

- Programa não compila ou produz avisos (warnings) 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