

Sistemas Operativos

(ano letivo 2021-22)

”

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 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”, (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.

1. [5] Escreva um programa multitarefa em linguagem C padrão e segundo a norma POSIX, de nome `pteste.c`, para teste de uma função fornecida (Combinacoes), constituído por uma tarefa principal e várias tarefas de teste. O programa recebe dois argumentos:

```
pteste <ntarefas> <casos.txt>
```

O programa recebe o número de tarefas a criar, e um ficheiro `casos.txt`, com os casos de teste, em que cada linha é um caso de teste, com dois números a serem fornecidos à função em teste, e um terceiro número que é o resultado esperado.

O programa deve efetuar as seguintes operações na **tarefa principal**:

- a) Ler o ficheiro `casos.txt`
- b) Iniciar as tarefas de teste
- c) Quando todas as tarefas terminarem, imprimir o número de casos de teste passados, falhados e totais, bem como para os casos de teste falhados, os dados de entrada e o valor esperado e observado, terminando de seguida

As **tarefas de teste** devem efetuar as seguintes operações em ciclo:

- a) Executar o primeiro caso de teste fornecido, não atribuído (utilizar uma variável partilhada entre as tarefas para o efeito). Caso não existam mais casos de teste, terminar.
- b) Executar o teste e gravar resultado, se o teste foi passado ou não, gravando o valor do resultado.
- c) Ir para passo a)

Exemplo de execução:

```
$ ./pteste 15 casos.txt
```

Resultados:

Sucesso: 912

Falhas: 86

Total casos: 998.

Erro no caso 404: input (30, 15) esperado/observado: 155117520 / -131213633.

Erro no caso 464: input (31, 13) esperado/observado: 206253075 / -124129024.

Erro no caso 492: input (33, 14) esperado/observado: 818809200 / 40051392.

Erro no caso 511: input (35, 18) esperado/observado: 242600354 / 79433710.

Erro no caso 546: input (35, 15) esperado/observado: -1047024136 / -28345986.

...

Notar que o número de tarefas poderá ser muito menor que o número de casos, pelo que cada tarefa irá executar o teste seguinte, podendo cada teste demorar tempos distintos.

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 2021-22 (pteste.c)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

Função para teste:

```
int Combinacoes(int n, int r)  
{  
    int i, combinacoes;  
  
    if (n < r || r < 1)  
        return 0;  
  
    i = 1;  
    combinacoes = 1;  
    while (i <= r)  
    {  
        combinacoes *= (n - r + i);  
        combinacoes /= i;  
        i++;  
    }  
    return combinacoes;  
}
```

Critérios de correção:

- Programa desenvolvido difere significativamente das especificações e instruções do enunciado => 0 valores.
- 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.
- O programa em conjunto com o relatório não está estruturado, comentado ou explicado de modo à fácil compreensão da sua estrutura e funcionamento => 0 valores.
- 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 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