

Sistemas Operativos

(ano letivo 2020-21)

”

E-fólio A | Instruções para a realização do E-fólio



Este enunciado constitui o elemento de avaliação designado por “e-fólio A” no âmbito da avaliação contínua e tem a cotação total de 3 valores. A sua resolução deve ser entregue até às 23h55 do dia 5 de abril 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-efA.zip”

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, “327555-Paulo-Costa-21111-efA.zip”, (sem acentos).

O ficheiro deve ser única e exclusivamente entregue através do recurso “E-fólio A” 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. [3] Escreva um programa em linguagem C padrão, de nome `dit.c`, que dada uma diretoria de entrada cria uma lista do conteúdo dessa diretoria, um item (ficheiro ou diretoria) por linha, adiciona no final o total de itens da lista e imprime a lista na saída padrão. Para efetuar este processamento, o programa deve utilizar a metodologia “original” que a seguir se descreve.

- O nome (ou pathname) da diretoria de entrada é o único argumento do programa na linha de comandos.

- O programa deve testar se o número de argumentos dado na linha de comandos é correto e se existe a diretoria de entrada (Dica: testar com a função de biblioteca `fopen()`). Em caso de erro deve emitir uma mensagem.

- O programa `dit` (processo A) executa a sua tarefa em 4 passos distintos criando 4 novos processos denominados B, C, D e E, um de cada vez, com recurso à função de sistema `fork()` e substituindo a sua imagem respetivamente pela dos comandos `ls`, `echo`, `wc` e `cat`. Os 4 passos são equivalentes à execução dos seguintes 4 comandos na linha de comandos (Shell):

```
$ ls -l dir > tmp1.txt # Passo 1
$ echo -n "Itens encontrados: " >> tmp1.txt # Passo 2
$ wc -l < tmp1.txt > tmp2.txt # Passo 3
$ cat tmp1.txt tmp2.txt # Passo 4
```

onde `$` representa a prompt da linha de comandos, `#` representa um comentário, `>` e `>>` são operadores de redireção da saída padrão (stdout). O operador `>` redireciona e cria um ficheiro enquanto o operador `>>` redireciona e acrescenta ao ficheiro. O operador `<` redireciona a entrada padrão (stdin).

- Por exemplo, para o passo 1 o programa `dit` (processo A) deve gerar uma lista com um item por linha no ficheiro "`tmp1.txt`", criando um novo processo (processo B) que substitui a sua imagem pela do comando `ls` e o executa com a opção "`-l`". Dica: antes da substituição da imagem, redirecionar a saída padrão stdout para o ficheiro "`tmp1.txt`" usando a função de biblioteca `freopen()`. Nota: Um processo filho herda os ficheiros abertos do processo pai.

- O processo A deve criar os processos C, D e E de modo semelhante. Só deve iniciar um após o anterior terminar. A substituição da imagem nos 4 processos criados deve ser feita com o recurso a pelo menos 2 funções de sistema diferentes da família `exec()` de entre o conjunto {`execl`, `execlp`, `execv`, `execvp`}, uma com a letra p no nome e outra sem.

- Cada processo distinto deve no seu início imprimir uma mensagem do tipo "Processo X: PID=xxx PPID=xxx". Neste trabalho a criação de processos deve ser feita recorrendo unicamente à função de sistema `fork()`, não devendo ser utilizada a função de biblioteca `system()`.

- Deve ser testado se ocorre erro na chamada à função de sistema `fork()`, caso em que o programa termina.

Pondere quais as funções de sistema/biblioteca que vai utilizar no programa e consulte as respectivas `man` pages para se informar dos detalhes de funcionamento de cada uma, assim como dos comandos `ls`, `echo`, `wc` e `cat`.

Apresenta-se a seguir um exemplo de execução do programa:

```
$ ./dit .
Processo A: PID= 1467  PPID= 1107
Processo B: PID= 1468  PPID= 1467
Processo C: PID= 1469  PPID= 1467
Processo D: PID= 1470  PPID= 1467
Processo E: PID= 1471  PPID= 1467
dit
dit.c
tmp1.txt
tmp2.txt
Itens encontrados: 4
$
```

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

```
/*
** UC: 21111 - Sistemas Operativos
** e-fólio A 2020-21 (dit.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 (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