

Sistemas Operativos

(ano letivo 2023-24)

”

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 8 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 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 `fdup.c`, que dada uma diretoria de entrada procura ficheiros duplicados (com mais de uma ocorrência) com o mesmo nome em todas as subdiretorias e imprime a respetiva 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 primeiro argumento do programa na linha de comandos.

- O nome do ficheiro a procurar ou alternativamente uma expressão geral com os caracteres `*` ou `?` (por exemplo `*.c`) é o segundo argumento do programa na linha de comandos.

- Exemplos de invocação do programa, supondo que está na diretoria corrente, são,

```
$ ./fdup ~/Doc prog.c
$ ./fdup ~/Doc '*.c'
```

onde no segundo exemplo é especificado para procurar ficheiros com extensão `.c` a partir da diretoria `Doc` na diretoria *home* do utilizador. Note-se que as pelicas em torno de `*.c` apenas servem para impedir o interpretador de comandos (shell) de expandir a expressão geral, o programa `fdup` irá receber a string `"*.c"` em `argv[2]`.

- 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 e terminar.

- O programa `fdup` (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 `find`, `sort`, `uniq` e `awk` (usualmente localizados na diretoria `/bin` ou `/usr/bin`). Os 4 passos do segundo exemplo são equivalentes às execuções seguintes na linha de comandos:

```
$ find ~/Doc -type f -name '*.c' -printf "%h %f\n" >tmp1.txt # Passo 1
$ sort -t ' ' -k2,2 <tmp1.txt >tmp2.txt # Passo 2
$ uniq -f1 -D <tmp2.txt >tmp3.txt # Passo 3
$ awk '{print $2,$1}' <tmp3.txt # Passo 4
```

onde `$` representa a prompt da linha de comandos, `#` representa um comentário, `<` e `>` são operadores de redireção da entrada padrão (stdin) e saída padrão (stdout) respetivamente. No caso do operador `>`, o ficheiro destino é criado ou reescrito. Note-se que aqui `~/Doc` representa a diretoria e `*.c` os ficheiros a procurar especificados respetivamente pelos argumentos dados ao programa `fdup`.

- O comando `find` lista todos os ficheiros regulares (opção `-type f`) encontrados com o nome especificado (opção `-name *.c`) para o ficheiro `tmp1.txt` com dois campos separados por um espaço em cada linha, a path da diretoria (representada por `%h`) e o nome do ficheiro (`%f`). O comando `sort` ordena as linhas do ficheiro `tmp1.txt` de acordo com o segundo campo (nome do ficheiro, opção `-k2,2`) e escreve o resultado no ficheiro `tmp2.txt`. Finalmente, o comando `uniq` lista apenas as linhas de `tmp2.txt` que têm o nome de ficheiro repetido e o comando `awk` troca de posição os dois campos em cada linha de `tmp3.txt` e imprime o resultado final na forma de nome de ficheiro e

respetiva pathname em cada linha. Estes comandos podem ser executados e inspecionados os conteúdos dos ficheiros tmp?.txt para compreender os vários passos envolvidos.

- Para o passo 1, por exemplo, o programa fdup (processo A) deve gerar uma lista de ficheiros no ficheiro "tmp1.txt", criando um novo processo (processo B) que substitui a sua imagem pela do comando find e o executa com as opções indicadas. 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 e só deve iniciar um após o anterior terminar. A substituição da imagem nos 4 processos criados deve ser feita obrigatoriamente com o recurso a 3 funções de sistema diferentes da família exec() do conjunto {execl, execlp, execv, execvp}.

- 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 respetivas man pages para se informar dos detalhes de funcionamento de cada uma, assim como dos comandos envolvidos.

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

```
/*  
** UC: 21111 - Sistemas Operativos  
** e-fólio A 2023-24 (fdup.c)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

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 geral 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