

Sistemas Operativos

(ano letivo 2016-17)

e-fólio A

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 17 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 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-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”

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).

I

1. [3] Escreva um programa em linguagem C padrão, de nome `mpchc.c` que crie uma cadeia com `np` processos, designados 1 a `np`, onde o processo `i+1` é filho do processo `i`. O processo 1 corresponde ao programa original `mpchc`. O objetivo do programa é dado um ficheiro de entrada trocar as suas letras não acentuadas minúsculas por maiúsculas (e vice-versa) e escrever o resultado num ficheiro de saída.

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

```
>> ./mpchc np ficheiro1 ficheiro2
```

- `np` é o nº total de processos da cadeia, incluindo o processo inicial `mpchc`.

- `ficheiro1` é o nome do ficheiro de entrada.

- `ficheiro2` é o nome do ficheiro de saída.

- O programa `mpchc` deve testar se o número de argumentos dado na linha de comandos é correto, se `np` é pelo menos 1 e se o ficheiro de entrada existe (testar com a função `fopen()`). Em caso de erro o programa deve emitir uma mensagem e terminar.

- Para formar a cadeia de processos o programa `mpchc` deve com recurso à função de sistema `fork()` criar os processos necessários. Dica: utilize um ciclo para criar os processos, note que uma cadeia pode ser vista como uma árvore só com um ramo.

- O ficheiro de entrada inicialmente aberto pelo programa `mpchc` deve permanecer aberto e o file pointer associado (variável tipo `FILE *`) utilizado por todos os processos da cadeia (um ficheiro aberto permanece aberto no processo filho após a chamada à função `fork()`).

- Cada processo deve esperar que o seu processo filho termine antes dele próprio terminar (processo `i` espera pelo processo `i+1`).

- Nenhum processo deve ser criado com recurso à função `system()`.

- Cada processo deve processar um bloco com $1/np$ do nº de caracteres do ficheiro de entrada, com exceção do processo `np` caso a divisão não seja exata.

- Após processar a sua parte dos caracteres e antes de terminar, cada processo deve imprimir uma mensagem do tipo "Processo X: PID=XXX, PPID=XXX, convertidos XXX caracteres de XXX a XXX".

- 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.

- O programa deve estar identificado com um cabeçalho similar ao seguinte,
/*
** UC: 21111 - Sistemas Operativos
** e-fólio A 2016-17 (mpchc.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