

Sistemas Operativos

(ano letivo 2022-23)

”

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 11 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 `gsn.c`, que gera e ordena uma lista de `n` bytes aleatórios e imprime a lista em decimal na saída padrão. Para efetuar este processamento, o programa deve utilizar a metodologia “original” que a seguir se descreve.

- O número de bytes `n` da lista 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 `n > 0`. Em caso de erro deve emitir uma mensagem e terminar.

- O programa `gsn` (processo A) executa a sua tarefa em 3 passos distintos criando 3 novos processos denominados B, C e D, um de cada vez, com recurso à função de sistema `fork()` e substituindo a sua imagem respetivamente pela dos comandos `head`, `hexdump` e `sort` (usualmente localizados na diretoria `/bin`). Os 3 passos são equivalentes à execução dos seguintes 3 comandos na linha de comandos (shell):

```
$ head -c NNN /dev/urandom >tmp.bin           # Passo 1
$ hexdump -v -e "/1 \"%d\\n\"" <tmp.bin >tmp.txt # Passo 2
$ sort -g tmp.txt                               # Passo 3
```

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

- O comando `head` lista os primeiros `NNN` bytes do ficheiro especial `/dev/urandom` gerador de bytes aleatórios para o ficheiro `tmp.bin`. O comando `hexdump` lista em decimal os bytes do ficheiro `tmp.bin` para o ficheiro `tmp.txt`. Finalmente, o comando `sort` ordena os números (linhas) de `tmp.txt` segundo um critério numérico.

- Tenha em conta que a string de formatação do comando `head`, quer para a shell quer em linguagem C, deve ser escrita como `"/1 \"%d\\n\""` devido ao significado especial dos caracteres `"` e `\` no interior da string, necessitando de ser precedidos pelo carácter `\` (procedimento em inglês designado por *escaped characters*) para anular o seu significado especial.

- Para o passo 1, por exemplo, o programa `gsn` (processo A) deve gerar uma lista de bytes constituída pelo ficheiro binário `"tmp.bin"`, criando um novo processo (processo B) que substitui a sua imagem pela do comando `head` e o executa com as opções `"-c"` e `"NNN"`, onde `NNN` significa o valor de `n` dado como argumento ao programa. Dica: antes da substituição da imagem, redirecionar a saída padrão `stdout` para o ficheiro `"tmp.bin"` 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 e D de modo semelhante e só deve iniciar um após o anterior terminar. A substituição da imagem nos 3 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 respectivas man pages para se informar dos detalhes de funcionamento de cada uma, assim como dos comandos head, hexdump e sort.

Apresenta-se a seguir exemplos de execução do programa:

```
$ ./gsn 8
Processo A: PID=12492  PPID=11677
Processo B: PID=12493  PPID=12492
Processo C: PID=12496  PPID=12492
Processo D: PID=12497  PPID=12492
5
54
67
97
134
148
194
243
$ ./gsn 1
Processo A: PID=12498  PPID=11677
Processo B: PID=12499  PPID=12498
Processo C: PID=12501  PPID=12498
Processo D: PID=12502  PPID=12498
161
$ ./gsn 0

Erro n<=0!

$
```

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

```
/*
** UC: 21111 - Sistemas Operativos
** e-fólio A 2022-23 (gsn.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