Sistemas Operativos

(ano letivo 2018-19)

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 15 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 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"

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 `mpss.c` que crie uma cadeia com np+1 processos, designados 0 a np, onde o processo i+1 é filho do processo i. O processo 0 corresponde ao processo inicial mpss. O objetivo do programa é calcular a soma s(n) do quadrado dos primeiros n números naturais, com a cadeia de processos 1 a np dedicados ao cálculo dos termos do somatório em partes iguais e o processo 0 a somar os resultados parciais obtidos e imprimir o resultado final.

$$s(n) = \sum_{i=1}^{n} i^2$$

- O programa mpss recebe obrigatoriamente 2 argumentos na linha de comandos,

```
>> ./mpss np n
```

- `np` é o nº de processos da cadeia dedicada ao cálculo do somatório, com $1 \leq np \leq 32$.

- `n` é o nº de termos do somatório (ou de números naturais), com $np \leq n \leq 999$.

- O programa mpss deve testar se o número de argumentos dado na linha de comandos é correto e se os seus valores são válidos. Em caso de erro o programa deve emitir uma mensagem e terminar.

- Para a criação de processos o programa mpss deve utilizar a função de sistema fork(), não sendo permitido utilizar a função system().

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

- De um modo geral o processo 0 cria o processo 1 e espera que ele termine (o que implica que a cadeia de processos dedicados 1 a np terminou), lê de um ficheiro auxiliar "data.aux" as somas parciais de cada processo dedicado, calcula e imprime o resultado final assim como o resultado esperado que é dado pela expressão,

$$s(n) = [n(n+1)(2n+1)]/6$$

- As somas parciais são escritas com uma variável tipo int com a função fwrite() no ficheiro auxiliar pela sequência do nº de ordem do respetivo processo (processo 1 no início do ficheiro, etc). O processo 0 deve inicialmente criar o ficheiro e todos os processos 0 a np devem usar o seu próprio file pointer (FILE*) para aceder ao ficheiro, para que possam aceder a posições diferentes sem conflitos.

- Cada processo dedicado calcula a soma parcial de n/np (divisão inteira) termos do somatório, exceto o último processo np que calcula os restantes.

- Dica: utilize um ciclo (cuja estrutura deve ser alvo de reflexão) para gerir/criar os processos dedicados de 1 a np, note que uma cadeia pode ser vista como uma árvore só com um ramo. De um modo geral cada processo:
(i) cria um processo filho (exceto o último np);
(ii) calcula a sua soma parcial;

(iii) escreve a sua soma parcial no ficheiro auxiliar;
(iv) imprime uma mensagem com o seu nº de ordem, PID, PPID, intervalo de valores de i para o qual calculou a soma parcial e o valor dessa soma (ver exemplos)
(v) espera que o seu filho termine (exceto o último np);
(vi) ele próprio termina.

- Apresentam-se a seguir 2 exemplos de execução do programa, que além de referência também mostram as mensagens e dados de saída a imprimir pelo programa:

```
>> ./mpss 3 100

Soma do quadrado dos primeiros 100 numeros naturais com cadeia de
3 processos dedicados
Processo inicial P0: PID= 2351  PPID= 2305
Processo P1 : PID= 2353  PPID= 2351 i=[  1- 33]  soma=    12529
Processo P2 : PID= 2354  PPID= 2353 i=[ 34- 66]  soma=    85492
Processo P3 : PID= 2355  PPID= 2354 i=[ 67-100]  soma=   240329
Resultado calculado:    338350
Resultado esperado :    338350
>> ./mpss 14 660

Soma do quadrado dos primeiros 660 numeros naturais com cadeia de
14 processos dedicados
Processo inicial P0: PID= 2357  PPID= 2305
Processo P1 : PID= 2358  PPID= 2357 i=[  1- 47]  soma=    35720
Processo P2 : PID= 2359  PPID= 2358 i=[ 48- 94]  soma=   245575
Processo P3 : PID= 2360  PPID= 2359 i=[ 95-141]  soma=   663076
Processo P4 : PID= 2361  PPID= 2360 i=[142-188]  soma=  1288223
Processo P5 : PID= 2362  PPID= 2361 i=[189-235]  soma=  2121016
Processo P6 : PID= 2363  PPID= 2362 i=[236-282]  soma=  3161455
Processo P7 : PID= 2364  PPID= 2363 i=[283-329]  soma=  4409540
Processo P8 : PID= 2365  PPID= 2364 i=[330-376]  soma=  5865271
Processo P9 : PID= 2366  PPID= 2365 i=[377-423]  soma=  7528648
Processo P10: PID= 2367  PPID= 2366 i=[424-470]  soma=  9399671
Processo P11: PID= 2368  PPID= 2367 i=[471-517]  soma= 11478340
Processo P12: PID= 2369  PPID= 2368 i=[518-564]  soma= 13764655
Processo P13: PID= 2370  PPID= 2369 i=[565-611]  soma= 16258616
Processo P14: PID= 2371  PPID= 2370 i=[612-660]  soma= 19830104
Resultado calculado:  96049910
Resultado esperado :  96049910
```

- 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 2018-19 (mpss.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 (warnnings) 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