

Obtenção dos elementos necessários para a prova

Para realizar a sua prova deverá:

1. ligar-se a **`http://sigex.fe.up.pt/`** e aceder à área dedicada a alunos (Estudantes, Realizar Exame): terá de fazer “login” com as suas credenciais do SIGARRA e introduzir o código de exame: **GSK660**.
2. Recolher o pacote ZIP contendo o enunciado da prova, ficheiros auxiliares de código e, eventualmente, mais alguns documentos que lhe poderão ser úteis para a prova.
3. Criar no computador uma pasta específica onde desenvolverá o código da prova.

Problema

Escrever um programa *multithread* que, em “**paralelo**”, efectue o cálculo do tamanho (número de *bytes*) de vários ficheiros. (Em rigor, como se verá, pretende-se o desenvolvimento de vários programas independentes de dificuldade crescente, um em cada alínea.)

Os requisitos base do programa são:

- ler da entrada padrão (*stdin*) nomes de ficheiros, cada um numa linha, até surgir a indicação de EOF (*end of file*);
- para cada nome, invocar um *thread* cuja função associada deverá ter o nome *thrfunc* ;
- cada *thread* recolhe do argumento da sua função (pelo menos) o nome do ficheiro cujo tamanho deverá calcular;
- o cálculo do tamanho de um ficheiro deve ser efectuado através de uma função, *du ()*, já preparada e fornecida para o efeito, cujo protótipo e descrição é:
 - `long long du(int fd);`
// retorna o tamanho (número de bytes) do ficheiro cujo descritor lhe é passado no argumento
// *fd* é o descritor do ficheiro, já aberto
- o resultado do cálculo do tamanho de cada ficheiro deve ser apresentado na saída padrão (*stdout*) no formato:
 - «<nome_ficheiro>:\t<tamanho> bytes\n», como no exemplo:
`fich_XYZ: 3532 bytes`

Notas:

1. Para efectuar testes ao código que desenvolver pode usar, para além dos inúmeros ficheiros que residem no computador onde está a fazer a prova(!), os ficheiros fornecidos junto (*testfile1*, *testfile2*, *testfile3*). O utilitário de sistema *du*, com a opção *-b*, também pode ser usado para validar parte dos testes!
2. Não se esqueça de que deve sempre **garantir o máximo de paralelismo** possível nos programas que desenvolver.
3. Por outro lado, deverá também garantir soluções **sem “espera activa”** (*busy waiting*), a não ser que tal seja explicitamente autorizado.

Alíneas

As alíneas seguintes especificam requisitos adicionais e com grau de dificuldade crescente; para cada uma, o valor mostrado entre parênteses rectos é a percentagem da cotação da prova correspondente à alínea.

Para cada uma das alíneas terá de ser desenvolvido um programa diferente; o nome do código fonte de cada um será do tipo «*pp?*.*c*», onde *?* será a letra da alínea correspondente e o nome do correspondente executável será «*pp?*». Por exemplo, para a alínea *b* o programa *ppb.c* depois de compilado, será invocado como, por exemplo,
ppb

a) [25%] Como “aquecimento”, considere uma simplificação do programa, em que sabe que só irá ser apresentado na entrada padrão (*stdin*) o nome de um ficheiro. No entanto, crie explicitamente um *thread*, que terá de operar como especificado acima.

b) [25%] Resolva o problema como inicialmente proposto (múltiplos nomes de ficheiros e múltiplos *threads*) e imprima na saída padrão os tamanhos dos ficheiros, mas sem preocupações com a ordem em que os nomes foram recebidos na entrada padrão. Por exemplo, para ficheiros nomeados pela ordem `fich_1`, `fich_2`, `fich_3`, `fich_4`, o resultado impresso poderá ser:

```
fich_3:      112 bytes
fich_2:     9918 bytes
fich_1:      876 bytes
fich_4:    12234 bytes
```

c) [25%] Na sequência da alínea anterior, resolva o problema inicial, mas agora garantindo que na saída padrão os tamanhos dos ficheiros são listados pela ordem em que os nomes foram recebidos na entrada padrão. Por exemplo, para os mesmos ficheiros nomeados pela ordem `fich_1`, `fich_2`, `fich_3`, `fich_4`, o resultado impresso terá de ser:

```
fich_1:      876 bytes
fich_2:     9918 bytes
fich_3:      112 bytes
fich_4:    12234 bytes
```

d) [25%] Esta alínea é idêntica à alínea anterior em termos de resultado impresso na saída padrão (resultados apresentados pela ordem de submissão dos ficheiros), mas com uma restrição adicional: só poderá usar threads independentes (*detached*).

Submissão da prova para avaliação

Quando terminar a sua prova deverá:

1. colocar todos os ficheiros com o seu código fonte (apenas esses!) numa pasta;
2. comprimir a pasta (com o conteúdo!) num arquivo zip;
3. transferir esse arquivo (zip) para <http://sigex.fe.up.pt/> , na área dedicada a alunos e, especificamente, à prova de Sistemas Operativos (EEC0139, MIEEC), ainda a decorrer.
Poderá ter de fazer “login” mais uma vez, com as suas credenciais do SIGARRA e introduzir o código de exame: **GSK660**.