

Problema:

Escrever um programa *multithreaded* que faça uma pesquisa em “**paralelo**” de uma *string* em vários ficheiros de texto. (Em rigor, como se verá, pretende-se o desenvolvimento de vários programas independentes de dificuldade crescente, um em cada alínea.)

Os requisitos base são:

- a *string* a pesquisar, bem como os nomes dos ficheiros de texto, devem ser passados como argumentos da linha de comando, como mostra o exemplo:

```
pp "texto a pesquisar" fich_a fich_b fich_c fich_d [...]
```

[...] indica que o número de ficheiros é variável

- cada ficheiro será processado por um *thread* criado exclusivamente para processamento desse ficheiro;

- cada *thread* terá de abrir o seu ficheiro e efectuar a pesquisa invocando, possivelmente repetidas vezes, a função fornecida:

```
int match_line(int fd, char *str);
// retorna o nº de linha em que se encontrou a ocorrência seguinte da string;
// uma nova pesquisa (a partir da linha seguinte) poderá ser efectuada reinvocando a função;
// o retorno de 0 (zero), indica que não há mais ocorrências da string no ficheiro;
// fd é o descriptor do fich, já aberto, a pesquisar;
// str é a string a pesquisar;
```

- obtém-se uma “**ocorrência**” (*match*), se a *string* ocorrer uma ou mais vezes na linha;

- o resultado parcial de cada ocorrência (*match*) a apresentar na saída padrão (numa linha) é:

```
<<nome_fich>:\t<num_linha>», como no exemplo:
fich_a:      35
```

Nota: os ficheiros de texto fornecidos junto (*testfile1*, *testfile2*, *testfile12*) podem ser usados para efectuar testes ao código a desenvolver. O utilitário *grep* também pode ser usado para validar os testes!

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 "texto a pesquisar" fich_a fich_b fich_c
```

a) [25%] Como “aquecimento”, considere uma simplificação do programa: suponha que só se irá pesquisar um ficheiro, pelo que a invocação será do tipo:

```
ppa "texto a pesquisar" fich
```

Desenvolva o código adequado em *ppa.c*, mas criando explicitamente um *thread*, que terá de efectuar a pesquisa.

Note que a *string* a pesquisar não é específica ao *thread* (no programa *multi-thread* apresentado inicialmente, ela é comum a todos os *threads*), pelo que poderá ser colocada em variável global. No entanto, os dados específicos ao *thread* deverão ser-lhe passados no argumento da função.

b) [25%] Resolva agora o problema como foi apresentado inicialmente (múltiplos ficheiros, múltiplos *threads*) e imprima na saída padrão os resultados das pesquisas, mas sem preocupações de agrupamento de ficheiros: isto é, as linhas com ocorrências (*matches*) de cada ficheiro podem aparecer misturadas. Por exemplo:

```
fich_a:      35
fich_b:      18
fich_a:      40
fich_c:     234
```

c) [25%] Na sequência da alínea anterior, resolva a pesquisa tal como foi inicialmente proposta, mas imprima na saída padrão os resultados com agrupamento por ficheiro: isto é, embora a pesquisa deva ser feita por múltiplos *threads* em simultâneo, as linhas com ocorrências (*matches*) de cada ficheiro devem aparecer juntas. Por exemplo:

```
fich_a:    35
fich_a:    40
fich_b:    18
fich_c:   234
```

Nesta alínea, deverá ser usada a função `pthread_join()`.

d) [25%] Esta alínea é idêntica à alínea anterior em termos de resultado impresso na saída padrão, mas agora não se pode usar a função `pthread_join()`. Além disso, se a solução apresentada exibir uma forma qualquer de “espera activa” (*busy waiting*), a cotação máxima será apenas 15%.

Submissão da sua prova

Quando terminar a sua prova deverá:

1. Verificar se os ficheiros com o seu código estão no diretório correspondente;
2. Executar: `make clean`
3. Comprimir num arquivo zip todos os ficheiros com o seu código;
4. Transferir esse arquivo (zip) para `http://sigex.fe.up.pt/` Note que poderá ter que fazer “login” mais uma vez com as suas credenciais para o SIGARRA e introduzir o código exame: **IWP541**