FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
MIEIC - 2007/2008
SISTEMAS OPERATIVOS
Trabalhos Práticos

TRABALHO Nº 1

Indexador de palavras

Objectivos do trabalho

Proporcionar a familiarização com a programação de sistema, em ambiente Unix/Linux, envolvendo, principalmente, a manipulação de ficheiros e directórios, o desenvolvimento de aplicações multiprocesso e a utilização de sinais como mecanismo de comunicação/sincronização entre processos.

Especificação do trabalho

Pretende-se desenvolver uma aplicação que mantenha, num ficheiro, um índice ordenado das palavras contidas nos ficheiros de texto de uma árvore de directórios.

Cada linha do ficheiro-índice contém a palavra indexada, em minúsculas, o nome e caminho do ficheiro onde a palavra se encontra e a hora e data de modificação do ficheiro, como se ilustra a seguir:

```
abanico         prog/sub/test.txt      13:23   12/6/2007
abano           prog/teste.txt         15:23    7/7/2007
                prog/foo               22:34    4/6/2007
...
zamora          prog/a/b/popo.txt      23:34    6/3/2006
```

O programa principal, de nome **`indexer`**, recebe os seguintes argumentos, pela ordem indicada,

1. o nome absoluto ou relativo do directório cujos sub-directórios e ficheiros deve processar,
2. o nome do ficheiro-índice,
3. o número mínimo de letras que uma palavra deve ter,
4. e um argumento opcional que serve para indicar ao programa que *não deve* ordenar o ficheiro-índice:

```
indexer Lusiadas Lusiadas.idx 5    #chamada linha com.do; ordena ficheiro-índice
indexer Lusiadas Lusiadas.idx 5 1  #chamadas seguintes; não ordena fich.-índice
```

Durante a sua execução, cada processo `indexer` deve monitorizar o directório corrente, lançando um novo processo `indexer` para processar cada subdirectório encontrado, assim como novos subdirectórios que vão surgindo, ao longo tempo. Cada processo não percorre, portanto, a árvore de subdirectórios.

O processos devem manter-se em execução até receberem o sinal SIGINT, sinal que devem enviar a todos os processos *por si* lançados e que não tenham já terminado.

Após todos os processos por si lançados terem terminado, o processo inicial deve ordenar o ficheiro-índice por ordem alfabética de palavras, removendo linhas duplicadas, e deve terminar.
Para ordenar o ficheiro-índice o programa deve invocar o programa de sistema **`sort`** (ver a secção 1 do manual sobre este comando).
Como a ordenação só é efectuada pelo primeiro processo da cadeia de processos, os processos adicionais lançados devem receber o quarto argumento, que serve para os distinguir do primeiro processo da cadeia.

O processo **`indexer`** deve ainda lançar um processo auxiliar, de nome **`aux_indexer`**,

- para cada ficheiro regular encontrado no directório corrente,
- para novos ficheiros regulares que possam aparecer,
- ou ainda, sempre que um ficheiro regular já existente seja alterado.

O programa auxiliar **`aux_indexer`** deverá processar esse ficheiro, recebendo os seguintes argumentos, pela ordem indicada,

1. o nome do ficheiro a processar,
2. o nome do ficheiro-índice,
3. o número mínimo de letras que uma palavra deve ter:

```
aux_indexer Canto-1.txt Lusiadas.idx 5
```

Para cada palavra encontrada, o programa auxiliar deve acrescentar a informação relevante no fim do ficheiro-índice. O programa auxiliar deve finalizar a sua execução após terminar de processar o ficheiro, ou imediatamente após receber o sinal SIGINT.

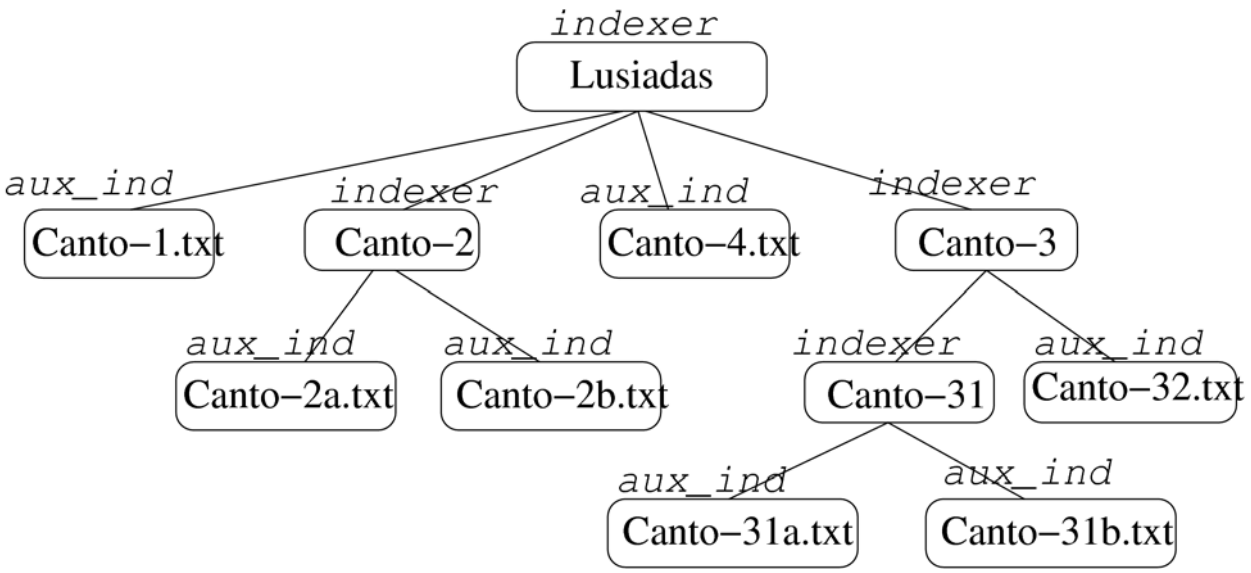
Valorização (*2 valores*)**:**

Quando um ficheiro regular é alterado durante a execução da aplicação, é necessário, antes de acrescentar os novos dados no ficheiro-índice, apagar deste todas as entradas relativas a esse ficheiro, de modo a não haver entradas duplicadas ou inconsistentes.

Para poder fazê-lo com segurança, o processo **`indexer`** que detectar a modificação deve obter acesso exclusivo ao ficheiro-índice, avisando através do sinal SIGUSR1 todos os processos **`aux_indexer`** que devem parar de escrever nele. Para mais detalhes sobre como enviar um sinal a um grupo de processos ler a secção 2 do manual sobre **`kill`** e **`setpgrp`**.

O acesso exclusivo ao ficheiro-índice deve ser efectuado de *forma cooperativa* entre todos os processos através da chamada **`flock`**, que só garante acesso exclusivo a um ficheiro se todos os processos usarem a função. Para mais detalhes consultar a secção 2 do manual sobre **`flock`**.

O processo de eliminação de entradas deve fazer-se copiando apenas a informação desejada do ficheiro-índice para um novo ficheiro que, no fim, deverá ser renomeado com o mesmo nome do ficheiro-índice original.



Exemplo de uma árvore de directórios e dos processos associados;
todos os processos associados a um directório foram lançados pelo processo **`indexer`** do directório-pai.