Ciências ULisboa | Informática

Sistemas Operativos

2021/22

Trabalho de projeto – 1ª Fase

Objetivo

Realização de um trabalho de programação em Python envolvendo a criação de processos/*threads* e a comunicação entre processos/*threads*.

Introdução

O comando *grep* pesquisa as linhas dos ficheiros que contêm determinada palavra ou expressão regular. Uma dada palavra a procurar pode aparecer isolada dentro de uma linha (i.e., está separada das restantes palavras dentro da linha de texto) ou fazer parte de outra palavra. Por exemplo, nas duas frases seguintes: "O Sistema Operativo utiliza o hardware" e "O Sistema Operativo satisfaz os pedidos do utilizador"; se procurarmos a palavra "utiliza", o comando *grep* devolve ambas as linhas. Este comando quando usado com várias palavras a pesquisar e com um conjunto alargado de ficheiros, apresenta alguns problemas de desempenho.

Com este trabalho pretende-se desenvolver o comando *pgrepwc* (parallel grep with counting), uma versão do *grep* com funcionalidades acrescidas e que funciona em paralelo. O comando irá emitir as linhas de texto que contêm as palavras isoladas a pesquisar, a contagem das linhas resultantes e o número de ocorrências das palavras a pesquisar.

Descrição do trabalho

Pretende-se que os alunos implementem o comando `pgrepwc` descrito de seguida:

NOME

`pgrepwc` – pesquisa até um máximo de três palavras em um ou mais ficheiros, devolvendo as linhas de texto que contêm unicamente uma das palavras (isoladamente) ou todas as palavras. Também, conta o número de ocorrências encontradas de cada palavra e o número de linhas devolvidas de cada palavra ou de todas as palavras. A pesquisa e contagem são realizadas em paralelo, em vários ficheiros.

SINOPSE

Pedro e Lucas - a opção -a apenas dá ou só Pedro ou só Lucas

```
pgrepwc [-a] [-c|-l] [-p n] palavras {ficheiros}
```

a - apenas uma palavra (tem de ser exatamente igual)

DESCRIÇÃO

FAZER VALIDAÇÕES (se não há um comando indica erro)

-a é opcional

`-a`: opção que define se o resultado da pesquisa são as linhas de texto que contêm unicamente uma das palavras ou todas as palavras. Por omissão, somente as linhas contendo unicamente uma das palavras serão devolvidas.

`-c`: opção que permite obter o número de ocorrências encontradas das palavras a pesquisar. ou -c ou -l

`-l`: opção que permite obter o número de linhas devolvidas da pesquisa. Caso a opção `-a` não esteja ativa, o número de linhas devolvido é por palavra.

`-p n`: opção que permite definir o nível de paralelização `n` do comando (ou seja, o número de processos

ex:

python grepwe.py -e -p 2 Lucas file1 file2 file3

pai - cria 2 filhos - filho 1 - procura Lucas no file 1 e 2 conta e imprime
- filho 2 - procura Lucas no file 3 conta e imprime

quando p > nº files, o programa redifine para os existentes

se não tiver -p, o pai não faz nada, quem faz são os filhos

(filhos)/*threads* que são utilizados para efetuar as pesquisas e contagens). Por omissão, deve ser utilizado apenas um processo (o processo pai) para realizar as pesquisas e contagens.

`palavras`: as palavras a pesquisar no conteúdo dos ficheiros. O número máximo de palavras a pesquisar é de 3.

`ficheiros`: podem ser dados um ou mais ficheiros, sobre os quais é efetuada a pesquisa e contagem. Caso não sejam dados ficheiros na linha de comandos, estes devem ser lidos de *stdin* (o comando no início da sua execução pedirá ao utilizador quem são os ficheiros a processar).

Inicialmente, após a validação das opções do comando, o processo pai deve criar os processos filhos/*threads* definidos pelo nível de paralelização do comando (valor `n`). Estes processos/*threads* pesquisam as palavras nos ficheiros, contam as ocorrências das palavras e o número de linhas onde estas foram encontradas nos ficheiros e escrevem os resultados (linhas encontradas e contagens) para *stdout*. Os resultados das pesquisas e contagens são escritos para *stdout* de forma não intercalada, ou seja, os resultados de cada processo/*thread* são apresentados sequencialmente, sem serem intercalados com os resultados dos outros processos/*threads*.

Os processos/*threads* realizam as pesquisas e as contagens nos ficheiros atribuídos pelo processo pai. Um dado ficheiro é atribuído a um só processo/*thread*, não havendo assim divisão do conteúdo de um ficheiro por vários processos/*threads*. Neste sentido, se o valor de `n` for superior ao número de ficheiros, o comando redefine-o automaticamente para o número de ficheiros.

No final, o processo pai terá de escrever para *stdout* o número total de ocorrências das palavras ou de linhas encontradas, de acordo com a opção especificada de contagem (`c` ou `l`).

Os alunos têm de implementar duas soluções: uma com processos e outra com *threads*.

Desafios

- Como garantir que a pesquisa e a contagem são efetuadas em todos os ficheiros uma e apenas uma vez?
- Como garantir que os resultados para *stdout* são escritos de forma não intercalada?
- Como passar a informação necessária ao processo pai de modo a ser possível calcular o número total de ocorrência das palavras a pesquisar ou linhas encontradas?

Ficheiros de teste

Juntamente com o enunciado do projeto, serão disponibilizados quatro ficheiros de texto que servirão para os alunos testarem as duas soluções do comando `pgrepwc`. Os alunos terão de descarregar estes ficheiros para a sua máquina. Não os deverão abrir no moodle, principalmente o ficheiro `file1.txt` por este ser grande (~500 MB).

Entrega

A entrega do trabalho é realizada da seguinte forma:

- Os grupos inscrevem-se atempadamente, de acordo com as regras afixadas para o efeito, no moodle.
- Colocar os ficheiros `pgrepwc.py e pgrepwc_threads.py` do projeto numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (por exemplo grupo01 ou grupo23). Juntamente com os dois ficheiros, incluir um ficheiro de texto README.txt (não é .pdf nem .rtf nem .doc nem .docx) onde os alunos devem colocar: (1) a identificação dos elementos do grupo; (2) como executar ambos os ficheiros .py; (3) relatar a informação que acharem pertinente sobre a sua implementação do projeto (ex., limitações, qual a abordagem usada para a divisão dos ficheiros pelos processos). A diretoria será incluída num ficheiro ZIP cujo nome deve

seguir exatamente o padrão **grupoXX.zip**. Esse ficheiro deverá ser submetido no moodle (um por grupo).

Note que a **entrega deve conter apenas os dois ficheiros .py e o ficheiro README.txt, qualquer outro ficheiro vai ser ignorado**.

Se não se verificar algum destes requisitos o trabalho é considerado não entregue.

Não serão aceites trabalhos entregues por mail nem por qualquer outro meio não definido nesta secção.

Prazo de entrega

O trabalho deve ser entregue até dia **07 de novembro de 2021 (domingo) às 23:59h**.

Avaliação dos trabalhos

A avaliação do trabalho será realizada:

(1) pelos alunos, pelo preenchimento do formulário de contribuição de cada aluno no desenvolvimento do projeto. O formulário será disponibilizado no Moodle e preenchido após a entrega do projeto.

(2) pelo corpo docente sobre dois conjuntos de ficheiros de texto: os ficheiros de teste disponibilizados aos alunos e outros somente usados pelos docentes.

Para além dos testes a efetuar, os seguintes parâmetros serão avaliados: funcionalidade, estrutura, desempenho, algoritmia, comentários, clareza do código, validação dos parâmetros de entrada e tratamento de erros.

Divulgação dos resultados

A data prevista da divulgação dos resultados da avaliação dos trabalhos é 26 de novembro de 2021.