



Trabalho prático de avaliação I

Objetivo

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

Introdução

O comando *wc* conta o número de caracteres, palavras e linhas que um ficheiro contém, bem como indica qual o tamanho da linha mais longa do ficheiro. Quando usado com um conjunto alargado de ficheiros, o *wc* apresenta alguns problemas de desempenho.

Com este trabalho pretende-se desenvolver o comando *pwc* (parallel *wc*), uma versão do *wc* que faz a contagem de linhas, palavras e caracteres em paralelo.

Descrição do trabalho

Pretende-se que os alunos implementem o comando *pwc* descrito de seguida:

NOME

pwc – conta, em paralelo, em vários ficheiros, o número de caracteres, palavras e linhas que um ficheiro contém e indica para cada ficheiro qual o tamanho da sua linha mais longa.

SINOPSE

```
pwc [-c|-w|-l [-L]] [-p n] {ficheiros}
```

DESCRIÇÃO

-c: opção que permite obter o número de caracteres de um ficheiro.

-w: opção que permite obter o número de palavras de um ficheiro.

-l: opção que permite obter o número de linhas de um ficheiro.

-L: opção opcional que permite calcular o tamanho da linha mais longa do ficheiro. Esta opção só pode ser combinada com a opção -l.

-p n: opção opcional que permite definir por *n* o nível de paralelização do comando (ou seja, o número de processos/*threads* que são utilizados para fazer as contagens). Por omissão, deve ser utilizado apenas um processo para fazer as contagens.

Podem ser dados zero ou mais ficheiros, sobre os quais é feita a contagem. Caso não sejam dados ficheiros na linha

de comandos, estes devem ser lidos de *stdin*.

Inicialmente, o processo pai deve criar os processos/*threads* definidos pelo nível de paralelização do comando (valor *n*). Estes processos/*threads* contam os caracteres, palavras ou linhas de um ficheiro e escrevem o resultado para *stdout*. Estes processos/*threads* continuam as contagens até estas terem sido realizadas em todos os ficheiros. Os resultados das contagens são escritos para *stdout* de forma não intercalada.

No final, o processo pai deve escrever para *stdout* o número total de caracteres, palavras ou linhas, de acordo com a opção especificada de contagem.

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

Desafios

- Como garantir que a contagem é feita 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 caracteres, palavras ou linhas?

Entrega

A entrega do trabalho é realizada da seguinte forma:

- Os grupos devem inscrever-se atempadamente, de acordo com as regras afixadas para o efeito, no moodle.
- Colocar os ficheiros `pwc.py` e `pwc_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 a identificação dos elementos do grupo e relatar a informação que acharem pertinente sobre a sua implementação do projeto (por exemplo, limitações). 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 **10 de novembro de 2019 (domingo) às 23:59h.**

Avaliação dos Trabalhos

As avaliações dos trabalhos serão realizadas na **semana de 16 a 20 de dezembro de 2019**. Todos os elementos do grupo terão de comparecer à avaliação e a avaliação é feita **individualmente**. Deste modo, cada elemento do grupo deve estar preparado para responder a qualquer questão relacionada com os trabalhos e com a matéria das aulas teórico-práticas.

Alguns parâmetros de avaliação

Funcionalidade, Estrutura, Desempenho, Algoritmia, Comentários, Clareza do código.