



**Ciências
ULisboa**

Departamento de Informática, Faculdade de Ciências da
Universidade de Lisboa

Sistemas Operativos

2018/19

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 *grep* procura as linhas dos ficheiros que têm correspondência com um padrão. Quando usado com um conjunto alargado de ficheiros, o *grep* apresenta alguns problemas de desempenho.

Com este trabalho pretende-se desenvolver o comando *pgrep* (parallel *grep*), uma versão do *grep* que faz a pesquisa em paralelo.

Descrição do trabalho

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

NOME

pgrep – procura, em paralelo, em vários ficheiros, as linhas que contêm o texto indicado

SINOPSE

pgrep [-p n] {ficheiros} texto

DESCRIÇÃO

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

Podem ser dados zero ou mais ficheiros, sobre os quais é feita a procura. 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. Estes processos/*threads* procuram as linhas de um ficheiro que têm o texto e escrevem o resultado para *stdout*. Estes processos/*threads* continuam as pesquisas até estas terem sido realizadas em todos os ficheiros. Os resultados das pesquisas são escritos para *stdout* de forma não intercalada.

No final, o processo pai deve escrever para *stdout* o número total de linhas onde foi encontrada a palavra.

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

Desafios

Como garantir que a pesquisa 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 linhas onde foi encontrada a palavra?

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 o ficheiro `pgrep.py` do projeto numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (por exemplo grupo01 ou grupo23). Juntamente com o ficheiro `pgrep.py` deverá ser enviado 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 devem 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 o ficheiro .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 **4 de Novembro de 2018 (domingo) às 23:59h.**

Avaliação dos Trabalhos

As avaliações dos trabalhos serão realizadas na **semana de 10 a 14 de dezembro de 2018**. 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.