



**Ciências
ULisboa**

Departamento de Informática, Faculdade de Ciências da
Universidade de Lisboa

Sistemas Operativos

2019/20

Trabalho prático de avaliação II

Objetivo

Realização de um trabalho de programação em Python envolvendo a sincronização de processos, a manipulação de ficheiros e o tratamento de sinais, tempo e alarmes.

Introdução

Este trabalho pretende estender o comando *pwc* (versão desenvolvida com processos, no primeiro trabalho prático de avaliação) com algumas funcionalidades adicionais.

Descrição do trabalho

Pretende-se que os alunos concretizem 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] [-a s] [-f file] {ficheiros}
```

DESCRIÇÃO

Às funcionalidades já descritas no trabalho 1, os alunos devem adicionar as seguintes.

Caso o número de processos definidos por *n*, pela opção *-p*, seja maior do que o número de *ficheiros* a processar, o programa terá de dividir o conteúdo dos ficheiros pelos processos e sincronizar o acesso dos processos aos ficheiros, quando, por exemplo, dois processos estão a processar um mesmo ficheiro.

Caso o processo receba o sinal *SIGINT*, deve terminar corretamente, isto é, os processos devem concluir o processamento nos ficheiros correntes e terminar de seguida. O processo pai escreve para *stdout* o número de caracteres, palavras, ou linhas contadas até ao momento, considerando apenas os ficheiros que foram processados.

A opção *-a* é opcional e permite definir o intervalo de tempo (dado pelo argumento *s*) em que o processo pai escreve para *stdout* o estado da contagem com a seguinte informação: número de caracteres, palavras ou linhas contadas até ao momento, número de ficheiros completamente processados e tempo decorrido desde o início da execução do programa (em micro-segundos).

A opção *-f* é opcional e permite definir o ficheiro usado para guardar o histórico da execução do programa. O conteúdo do ficheiro *file* deve ser armazenado em binário. A informação guardada neste ficheiro deve ser a que é necessária para o comando *hpwc*.

Adicionalmente os alunos devem desenvolver o seguinte comando:

NOME

hpwc – lê o histórico de execução do programa pwc

SINOPSE

hpwc file

DESCRIÇÃO

Lê o histórico de uma execução do programa *pwc* guardada em *file* e apresenta a seguinte informação em *stdout*:

Início da execução da pesquisa: <dia/mês/ano, horas:minutos:segundos:microsegundos>

Duração da execução: <horas:minutos:segundos:microsegundos>

Processo: <pid>

ficheiro: <nome>

tempo de pesquisa: <horas:minutos:segundos:microsegundos>

dimensão do ficheiro: <dimensão>

número de [*caracteres* | *palavras* | *linhas*]: <número> ficheiro: ...

Processo: ...

Obs: a linha do histórico que indica a contagem é definida de acordo com o tipo da contagem, ou seja, aparecerá “número de caracteres”, “número de palavras” ou “número de 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 .py do projeto numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (por exemplo grupo01 ou grupo23). Juntamente com os 2 ficheiros .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 os 2 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 **14 de dezembro 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

Validação de inputs no programa, Funcionalidade, Estrutura, Desempenho, Algoritmia, Comentários, Clareza do código.