



Trabalho de projeto – 2ª Fase

Objetivo

Continuação da realização do 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 *pgrepwc* (**versão desenvolvida com processos**, na 1ª Fase trabalho de projeto de avaliação) com algumas funcionalidades adicionais.

Descrição do trabalho

Pretende-se que os alunos concretizem 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

```
pgrepwc [-a] [-c|-l] [-p n] [-w s] [-o file] {palavras} [-f ficheiros]
```

DESCRIÇÃO

Às funcionalidades já descritas na 1ª Fase do trabalho de projeto, os alunos devem melhorar as já implementadas e adicionar as seguintes.

A pesquisa das palavras nesta fase do projeto é *case-sensitive*, ou seja, o resultado da pesquisa deve ser o mais restrito e preciso de acordo com as palavras a pesquisar. Por exemplo, se queremos pesquisar a palavra “sistemas”, as linhas devolvidas e ocorrências contabilizadas são todas as que contêm a palavra “sistemas”, e não “Sistemas” ou qualquer outra variante *case-insensitive* da palavra (ex., SisTemaS). No entanto, mantem-se os resultados com caracteres especiais que antecedem ou precedem a palavra (ex., sistemas!).

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 os processos ao acesso aos ficheiros, quando, por exemplo, dois processos estão a processar um mesmo ficheiro. Por exemplo, se o *n* for igual a 3 e o número de ficheiros for igual a 5 (f1 a f5), uma possível distribuição dos ficheiros pelos processos será a seguinte: o 1º processo processa 2 ficheiros (f1 e f2); o 2º processo processa o ficheiro f3 e metade do ficheiro f4; o 3º processo processa a outra metade do ficheiro f4 e o ficheiro f5. O algoritmo de distribuição dos ficheiros pelos processos é da responsabilidade do grupo de trabalho. A distribuição dos ficheiros (ou conteúdo

destes) pelos processos deverá ser a mais equitativa possível.

Caso o processo pai receba o sinal SIGINT (i.e., CTRL+C), o processamento dos ficheiros deve terminar corretamente, isto é, os processos devem concluir o processamento nos ficheiros correntes e terminar de seguida. Por seu turno, o processo pai escreve para *stdout* o número de ocorrências encontradas de cada a palavra a pesquisar ou o número de linhas onde cada palavra foi encontrada até ao momento, considerando apenas os ficheiros que foram processados pelos processos.

A opção `-w` 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 até ao momento, com a seguinte informação: (1) número de ocorrências de cada palavra ou número de linhas resultantes da pesquisa (independentemente se a opção `-a` está ativa ou não); (2) número de ficheiros completamente processados; (3) número de ficheiros em processamento; (4) tempo decorrido desde o início da execução do programa (em micro-segundos).

A opção `-o` 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 a execução do comando `hpgrepwc`.

Todos os acessos a zonas de memória partilhada, tanto para escrita como para leitura, terão de ser devidamente sincronizados para evitar erros e outputs inesperados. Assim, os acessos por vários processos ao mesmo ficheiro, bem como os acessos a buffers partilhados entre o processo pai e os processos filhos terão de ser sincronizados.

Adicionalmente os alunos devem desenvolver o seguinte comando:

NOME

`hpgrepwc` – lê o histórico de execução do programa `pgrepwc`

SINOPSE

`hpgrepwc file`

DESCRIÇÃO

Lê o histórico de uma execução do programa `pgrepwc` 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>

Número de processos filhos: <valor de n>

Opção `-a` ativada: [Sim | Não]

Emissão de alarmes no intervalo de <valor de s> segundos

Processo: <pid>

ficheiro: <nome>

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

dimensão do ficheiro: <número de linhas processadas>

número de [ocorrências | linhas] da palavra_1: <número>

número de [ocorrências | linhas] da palavra_2: <número>

número de [ocorrências | linhas] da palavra_3: <número>

ficheiro: <nome>

...

Processo: ...

Observações: (1) caso a opção `-p` e `-w` não sejam ativadas, o valor `n` e `s` serão zero; (2) a vermelho significa que a linha do histórico sobre a contagem é definida de acordo com o tipo da contagem, ou seja, aparecerá “número de ocorrências” ou “número de linhas”. O número destas linhas terá de aparecer de acordo com o número de palavras a pesquisar.

Ficheiros de teste

Os mesmos disponibilizados na 1ª fase do projeto. Os alunos terão de descarregar os ficheiros para a sua máquina. Não os deverão abrir no moodle, principalmente o ficheiro `file1.txt` por este ser grande.

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 `pgrepwc.py` e `hpgrepwc.py` do projeto numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (por exemplo `grupo01` ou `grupo23`). Juntamente com os dois 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: (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 **12 de dezembro de 2021 (domingo) às 23:59h.**

Avaliação dos Trabalhos

As avaliações dos trabalhos serão realizadas remotamente na **entre 13 e 17 de dezembro de 2021**. 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.

Divulgação dos resultados

A data prevista da divulgação dos resultados da avaliação dos trabalhos é 31 de dezembro de 2021.