

Processos

A resolução da presente ficha de trabalho deverá fazer parte integrante de um único relatório que deverá:

- conter a resolução de TODAS as fichas de Programação até à data de entrega;
- ser submetido em formato pdf, através da plataforma Nónio, dentro do prazo indicado nessa plataforma;
- incluir separadamente, num único ficheiro .c, o CÓDIGO FONTE relativo a todas as fichas de trabalho;
- identificar o aluno e ano letivo nos printscreens realizados do terminal, recorrendo à variável de shell PS1 para introduzir o primeiro e último nome no prompt do terminal;
- seguir o modelo disponível na plataforma Nónio;
- ser realizado individualmente;
- incluir uma análise SWOT;
- ser assinado digitalmente.

1. Relativamente ao comando `ps`:
 - a. Descreva os seus parâmetros mais importantes, nomeadamente o parâmetro que permite obter a prioridade de um processo.
 - b. Descreva as várias colunas que são observáveis através do comando `ps aux`.
 - c. Realize a comparação deste comando com os comandos `top` e o `pstree`.
2. Identifique a função do comando `grep`. Exemplifique a sua utilização conjuntamente com o comando `ps`.
3. Descreva a utilização das funções `system()` e `exec()`. Descreva a maior diferença entre elas.
4. Desenvolva o programa `SO_2324_ficha01_NomeAluno.c` que apresente um submenu onde se poderão escolher as seguintes ações:
 - a) Listagem de Processos
 - i. Listagem de Processos (`ps aux`)
 - ii. Listagem de Processos com parâmetros definidos pelo utilizador (`ps`)
 - iii. Listagem de Processo à escolha do utilizador (`ps aux | grep`)
 - iv. Listagem de Processos em tempo real (`top`)

A execução do comando `top` implicará a destruição do programa inicial. Após a execução dos restantes comandos o programa retornará ao menu inicial.

5. Acrescente ao programa `SO_2324_ficha01_NomeAluno.c` um submenu que permita escolher as seguintes ações:
 - b) Consultar PIDs
 - i. PID do processo actual (`getpid()`)
 - ii. PID do processo pai (`getppid()`)
6. **Adicione ao programa `SO_2324_ficha01_NomeAlunos.c` uma opção designada por “Estatística Descritiva (Média e Variância)” . Quando esta opção for seleccionada o utilizador deverá:**
 - a. introduzir o nome do ficheiro a analisar;
 - b. informar sobre a existência (ou não) de uma linha cabeçalho no ficheiro;
 - c. a coluna de dados a analisar.

Para testar deverá ser utilizado o ficheiro disponibilizado no site da disciplina intitulado

`PL_Programacao_DB.csv`. Após a obtenção do nome do ficheiro, o processo deverá calcular a média e variância da coluna escolhida.

7. Descreva sucintamente em que consiste o filtro `awk`, indicando em particular como extrair linhas e colunas de um ficheiro de texto usando esta ferramenta.
8. **Adicione ao programa `SO_2324_ficha01_NomeAlunos.c` uma opção designada por “Estatística Descritiva (Média e Variância) com `awk`” que deverá realizar a implementação da alínea anterior recorrendo à utilização do `awk`.**

O menu final do programa deverá ser o seguinte:

- a) Listagem de Processos
 - i. Listagem de Processos (**ps aux**)
 - ii. Listagem de Processos com parâmetros definidos pelo utilizador (**ps**)
 - iii. Listagem de Processo à escolha do utilizador (**ps aux | grep**)
 - iv. Listagem de Processos em tempo real (**top**) *
 - b) Consultar PIDs
 - i. PID do processo actual (**getpid()**)
 - ii. PID do processo pai (**getppid()**)
 - c) Estatística Descritiva (Média e Variância)
 - d) Estatística Descritiva (Média e Variância) com awk
 - e) Sair
- * com autodestruição do processo

Nota: o `awk` é uma ferramenta poderosa que permitir manipular o conteúdo de ficheiros

e.g.

```
system("awk '{print $2}' gastototalsemreciclagem.dat > awk_custosr1.dat");  
// Obtém a 2ª coluna do ficheiro gastototalsemreciclagem.dat
```

Bibliografia:

- N. Matthew, R. Stones, *"Beginning Linux Programming"*
- <http://www.thegeekstuff.com/2010/01/awk-introduction-tutorial-7-awk-print-examples>