

## Sinais

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 trabalho de programação até à data da submissão;
- 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 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. Cada mensagem enviada pelo programa para o ecrã deverá conter informação sobre o PID do processo que a emitiu (e.g. "PID: 1234").
2. Para testar o programa deverá ser utilizado o ficheiro disponibilizado anteriormente e intitulado PL\_Programacao\_00\_DB.csv
3. O programa realizado na ficha de trabalho anterior deverá ser atualizado de forma a ter um menu principal com opções para a execução de ambas as fichas de trabalho. O menu da 3ª ficha de trabalho deverá ter o seguinte conteúdo:

**Ficha 3 – Sinais**

- a) Cálculo da média e variância
- b) Apagar ficheiros de resultados
- c) Sair

4. Ao escolher a opção "**Ficha 3 – Sinais**" o processo pai deverá:
  - a. apresentar o submenu da Ficha 3;
  - b. criar três filhos (processo A, B e C) que ficarão a aguardar o envio de sinais por parte do processo pai;
  - c. escrever no ecrã "*Sou o processo pai (PID xxxx), e criei os processos A (PID xxxx), B (PID xxxx) e C (PID xxxx)*".

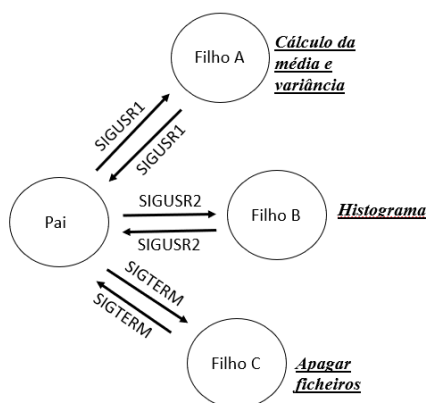


Figura 1 – Relação entre o processo pai e os 3 filhos

5. Ao escolher a opção "**Cálculo da média e variância**" do submenu da Ficha 3:
  - a. o **processo pai** deverá:
    - i. enviar (`kill()`) o sinal SIGUSR1 ao processo filho A;
    - ii. escrever no ecrã "*Sinal SIGUSR1 enviado pelo processo pai (PID xxxxx) ao processo filho A (PID xxxxx)*";
    - iii. aguardar pela receção de um sinal;

- b. o **processo filho A** ao receber (`signal()`) o sinal SIGUSR1 deverá:
- escrever no ecrã *"Sinal SIGUSR1 recebido pelo processo com PID xxxxx"*;
  - questionar o utilizador sobre:
    - o nome do ficheiro a abrir;
    - a existência de uma linha cabeçalho no ficheiro;
    - a coluna de dados a analisar.
  - calcular a média e variância da coluna de dados escolhida pelo utilizador, e escrever os mesmos na forma de linha no ficheiro `ficha03_media.dat`. A informação deve ser adicionada caso o ficheiro já exista;
  - criar o ficheiro `ficha03_sucesso.dat` com uma linha contendo informação da data e indicação do sucesso da operação. A informação deve ser adicionada caso o ficheiro já exista;
  - criar o ficheiro `ficha03_coluna.dat` com uma linha contendo os dados da coluna escolhida pelo utilizador e. A informação deve ser substituir a informação existente caso o ficheiro já exista;
  - indicação do sucesso da operação (e.g. *"Operações realizadas pelo processo filho A com sucesso!"*) e solicitar prosseguir;
- c. seguidamente o processo filho A deverá:
- enviar o sinal SIGUSR1 ao processo pai;
  - escrever no ecrã *"Sinal SIGUSR1 enviado pelo processo filho A (PID xxxxx) ao processo pai (PID xxxxx)"*;
  - ficar a aguardar por um novo sinal.
- d. o **processo pai** ao receber o sinal SIGUSR1 deverá:
- escrever no ecrã *"Sinal SIGUSR1 recebido pelo processo pai (PID xxxxx), vou enviar de seguida o sinal SIGUSR2 ao processo filho B (PID xxxxx). Pressione uma tecla para continuar..."*;
  - enviar sinal SIGUSR2 ao filho B;
  - escrever no ecrã *"Sinal SIGUSR2 enviado pelo processo pai (PID xxxxx) ao processo filho B (PID xxxxx)"*;
  - ficar a aguardar por um novo sinal.
- e. o **processo filho B** ao receber (`signal()`) o sinal SIGUSR2 deverá:
- escrever no ecrã *"Sinal SIGUSR2 recebido pelo processo B com PID xxxxx"*;
  - solicitar ao utilizador para prosseguir;
  - apresentar no ecrã o conteúdo dos ficheiros `ficha03_media.dat` e `ficha03_sucesso.dat`
  - solicitar ao utilizador para prosseguir;
  - apresentar o histograma do conteúdo presente no ficheiro `ficha03_coluna.dat`
  - solicitar ao utilizador para prosseguir;
  - enviar o sinal SIGUSR2 ao processo pai;
  - escrever no ecrã *"Sinal SIGUSR2 enviado pelo processo filho B (PID xxxxx) ao processo pai (PID xxxxx)"*;
  - ficar a aguardar por um novo sinal.
- f. o processo pai ao receber o sinal SIGUSR2 deverá:
- escrever no ecrã *"Sinal SIGUSR2 recebido pelo processo pai (PID xxxxx), retornando para o submenu da ficha 3. Pressione uma tecla para continuar..."*.
- g. apresentar o submenu da ficha 3.
6. Ao escolher a opção **"Apagar ficheiros de resultados"** do submenu da ficha 3:

- a. o **processo pai** deverá:
    - i. enviar o sinal SIGTERM ao processo filho C;
    - ii. escrever no ecrã *"Sinal SIGTERM enviado pelo processo pai (PID xxxxx) ao processo C (PID xxxxx)"*;
    - iii. aguardar pela receção de um sinal;
  - b. o **processo filho C** ao receber o sinal SIGTERM deverá:
    - i. escrever no ecrã *"Sinal SIGTERM recebido pelo processo C (PID xxxxx). Irei de seguida apagar os ficheiros .dat. Pressione uma tecla para continuar..."*;
    - ii. apagar o ficheiro ficha02\_mediana.dat;
    - iii. apagar o ficheiro ficha02\_sucesso.dat;
    - iv. apagar o ficheiro ficha02\_coluna.dat;
    - v. informar o utilizador que a tarefa foi executada com sucesso, e solicitar prosseguir;
  - c. em seguida o processo filho C deverá:
    - i. enviar o sinal SIGTERM ao processo pai (xxxx);
    - ii. escrever no ecrã *"Sinal SIGTERM enviado pelo processo filho C (PID xxxxx) ao processo pai (PID xxxxx)"*;
    - iii. ficar a aguardar por um novo sinal;
  - d. Ao receber o sinal SIGTERM o **processo pai** deverá:
    - i. escrever no ecrã *"Sinal SIGTERM recebido pelo pai (PID xxxxx), retornando para o submenu da ficha 3. Pressione uma tecla para continuar..."*.
  - e. apresentar o submenu da ficha 3
7. Ao escolher a opção **"Sair"** do submenu da Ficha 3:
- a. o **processo pai** deverá enviar um sinal SIGKILL a todos os filhos;
  - b. o processo pai deverá escrever no ecrã *"Sinal SIGKILL enviado aos filhos com PIDs xxxxx, xxxxx e xxxx, retornando para o menu principal. Pressione uma tecla para continuar..."*.
  - c. O processo pai deverá então apresentar o menu principal.

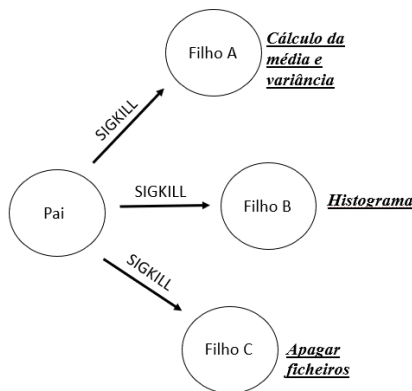


Figura 2 – Término dos processos filhos pelo processo pai

8. Quando se encontrar no submenu da ficha 3, utilize um segundo terminal de comandos para executar as opções deste submenu enviando sinais através da linha de comando. Tal deverá ser demonstrado na defesa.
9. O menu final do programa deverá ser o seguinte:
  - a) Ficha 1 – Processos
    - i. Listagem de Processos
    - ii. Consultar PIDs
    - iii. Estatística Descritiva (Mediana e Desvio Padrão)
    - iv. Estatística Descritiva (Mediana e Desvio Padrão) com awk
    - v. Sair

- b) Ficha 2 - Criação de Processos
  - i. Cálculo da média e variância
  - ii. Apagar ficheiros de resultados
  - iii. Sair
- c) Ficha 3 - Sinais
  - i. Cálculo da média e variância
  - ii. Apagar ficheiros de resultados
  - iii. Sair
- d) Sair

**Nota:** Caso seja necessário, a função `sprintf()` poderá ser usada para criar strings a partir de dados “formatados”.

**Example:**

```
#include<stdio.h>
int main()
{
    char buffer[40];
    int a = 2, b = 4, c;
    c = a * b;
    sprintf(buffer, "Multiplicação de %d e %d é %d", a, b, c);

    // A string "Multiplicação de 2 e 4 é 8" é armazenada
    // na string "buffer" em vez de ser enviada para o stdout

    printf("%s", buffer);

    return 0;
}
```

Output:

Multiplicação de 2 e 4 é 8

**Bibliografia:**

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