

Signals

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.

Nota:

Cada mensagem enviada para o ecrã deverá conter informação sobre o PID do processo que a emitiu (e.g. "PID: 1234").

Para testar o programa deverá ser utilizado o ficheiro disponibilizado anteriormente e intitulado PL_Programacao_00_DB.csv

1. O programa realizado na ficha de trabalho anterior deverá ser atualizado de forma a ser adicionado ao menu principal a opção para a execução da presente ficha de trabalho. O submenu desta ficha de trabalho deverá ter o seguinte conteúdo:

Ficha 3 – Sinais

- a) Cálculo da mediana e desvio padrão
- b) Apagar ficheiros de resultados
- c) Sair

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, C) que ficarão a aguardar o envio de sinais por parte do processo pai;
 - i. Cada processo filho deverá escrever:
"PROCESSO FILHO (PID xxxx): aguarda receção de sinal!"
- c. escrever no ecrã:
*"PROCESSO PAI (PID xxxx): criei os processos A (PID xxxx), B (PID xxxx) e C (PID xxxx).
Por favor introduza a opção pretendida!"*
- d. aguardar pela escolha de uma opção por parte do utilizador.

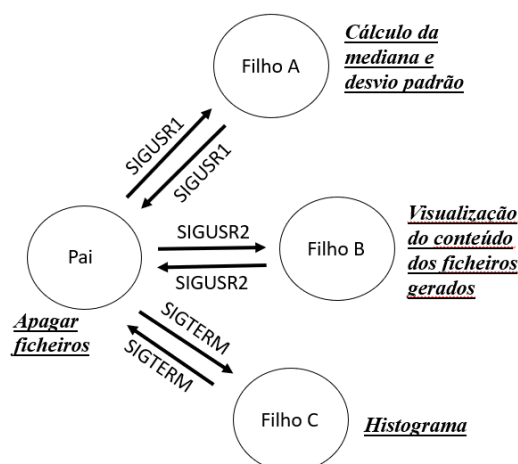


Figura 1 – Relação entre o processo pai e os 3 filhos

2. Ao escolher a opção **“Cálculo da mediana e desvio padrão”** do submenu da Ficha 3:

- a. o **processo pai** deverá:
 - i. enviar o sinal **SIGUSR1** (`kill()`) ao processo filho A;
 - ii. escrever no ecrã:
“PROCESSO PAI (PID xxxxx): sinal SIGUSR1 enviado pelo 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:
 - i. escrever no ecrã:
“PROCESSO FILHO A (PID xxxx): sinal SIGUSR1 recebido!”;
 - ii. questionar o utilizador sobre:
 1. o nome do ficheiro a abrir;
 2. a existência de uma linha cabeçalho no ficheiro;
 3. a coluna de dados a analisar.
 - iii. calcular a mediana e desvio padrão da coluna de dados escolhida pelo utilizador, e escrever os mesmos na forma de linha no ficheiro **ficha03_mediana.dat**. A informação deve ser acrescentada caso o ficheiro já exista. Escrever no ecrã:
“PROCESSO FILHO A (PID xxxx): ficheiro ficha03_mediana.dat atualizado com sucesso!”;
 - iv. criar o ficheiro **ficha03_coluna.dat** com uma linha contendo os dados da coluna escolhida pelo utilizador. A informação deve ser substituir a informação existente caso o ficheiro já exista. Escrever no ecrã:
“PROCESSO FILHO A (PID xxxx): ficheiro ficha03_coluna.dat atualizado com sucesso!”;
 - v. indicar o sucesso da operação e solicitar prosseguir:
*“PROCESSO FILHO A (PID xxxx): Operações realizadas com sucesso!
Pressione uma tecla para continuar...”*.
- c. seguidamente o processo filho A deverá:
 - i. enviar o sinal **SIGUSR1** ao processo pai;
 - ii. escrever no ecrã:
“PROCESSO FILHO A (PID xxxx): sinal SIGUSR1 enviado ao processo pai (PID xxxxx)”;
 - iii. ficar a aguardar por um novo sinal.
- d. o **processo pai** ao receber o sinal **SIGUSR1** deverá:
 - i. escrever no ecrã:
*“PROCESSO PAI (PID xxxxx): sinal SIGUSR1 recebido do processo filho A (PID xxxx). Sinal SIGUSR2 será de seguida enviado ao processo filho B (PID xxxxx) para apresentação do conteúdo dos ficheiros gerados.
Pressione uma tecla para continuar...”*;
 - ii. enviar sinal **SIGUSR2** ao filho B (`kill()`);
 - iii. escrever no ecrã:
“PROCESSO PAI (PID xxxxx): Sinal SIGUSR2 enviado ao processo filho B (PID xxxxx)”;
 - iv. ficar a aguardar por um novo sinal.
- e. o **processo filho B** ao receber (`signal()`) o sinal **SIGUSR2** deverá:
 - i. escrever no ecrã:
*“PROCESSO FILHO B (PID xxxx): Sinal SIGUSR2 recebido.
Pressione uma tecla para continuar...”*;
 - ii. apresentar no ecrã o conteúdo dos ficheiros **ficha03_mediana.dat** e **ficha03_coluna.dat**
 - iii. solicitar ao utilizador para prosseguir;
 - iv. enviar o sinal **SIGUSR2** ao processo pai;
 - v. escrever no ecrã:
“PROCESSO FILHO B (PID xxxx): Sinal SIGUSR2 enviado ao processo pai (PID xxxxx)”;
 - vi. ficar a aguardar por um novo sinal.

- f. o **processo pai** ao receber o sinal **SIGUSR2** deverá:
- i. escrever no ecrã:
“PROCESSO PAI (PID xxxxx): sinal SIGUSR2 recebido do processo filho B (PID xxxx). Sinal SIGTERM será de seguida enviado ao processo filho C (PID xxxxx) para geração do histograma.
Pressione uma tecla para continuar...”;
 - ii. enviar sinal **SIGTERM** ao filho C;
 - iii. escrever no ecrã:
“PROCESSO PAI (PID xxxxx): Sinal SIGTERM enviado ao processo filho C (PID xxxxx)”;
 - iv. ficar a aguardar por um novo sinal.
- g. o **processo filho C** ao receber (`signal()`) o sinal **SIGTERM** deverá:
- i. escrever no ecrã:
“PROCESSO FILHO C (PID xxxx): Sinal SIGTERM recebido.
Pressione uma tecla para continuar...”;
 - ii. apresentar o histograma do conteúdo presente no ficheiro **ficha03_coluna.dat**
 - iii. solicitar ao utilizador para prosseguir;
 - iv. enviar o sinal **SIGTERM** ao processo pai;
 - v. escrever no ecrã:
“PROCESSO FILHO C (PID xxxx): Sinal SIGTERM enviado ao processo pai (PID xxxxx)”;
 - vi. ficar a aguardar por um novo sinal.
- h. o **processo pai** ao receber o sinal **SIGTERM** deverá:
- i. escrever no ecrã:
“PROCESSO PAI (PID xxxxx): Sinal SIGTERM recebido do processo filho C (PID xxxxx).
Retornando para o submenu da ficha 3.
Pressione uma tecla para continuar...”.
 - ii. apresentar o submenu da ficha 3.
3. Ao escolher a opção **“Apagar ficheiros de resultados”** do submenu da ficha 3:
- a. o **processo pai** deverá:
- i. apagar o ficheiro **ficha03_mediana.dat**;
 - ii. apagar o ficheiro **ficha03_coluna.dat**;
 - iii. indicar o sucesso da operação e solicitar prosseguir:
“PROCESSO PAI (PID xxxxx): Eliminação com sucesso dos ficheiros .dat.
Pressione uma tecla para continuar...”.
4. 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:
“PROCESSO PAI (PID xxxxx): Sinal SIGKILL enviado aos filhos com PIDs xxxxx, xxxx, e xxxx.
Retornando para o menu principal.
Pressione uma tecla para continuar...”.
- c. O processo pai deverá então apresentar o menu principal.

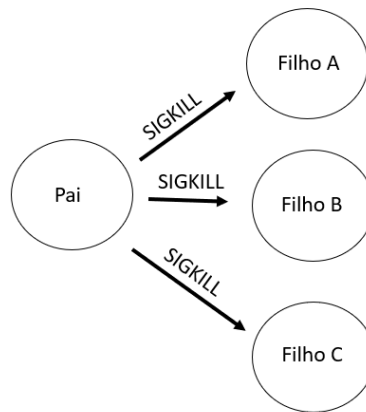


Figura 2 – Término dos processos filhos pelo processo pai

5. 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 igualmente ser demonstrado na defesa.
6. O menu final do programa deverá ser o seguinte:
 - a) Ficha 1 – Processos
 - i. Listagem de Processos
 - ii. Consultar PIDs
 - iii. Estatística Descritiva (Média e variância)
 - iv. Estatística Descritiva (Média e variância) com awk
 - v. Sair
 - b) Ficha 2 – Criação de Processos
 - i. Cálculo da mediana e desvio padrão
 - ii. Apagar ficheiros de resultados
 - iii. Sair
 - c) Ficha 3 – Sinais
 - i. Cálculo da mediana e desvio padrão
 - 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>