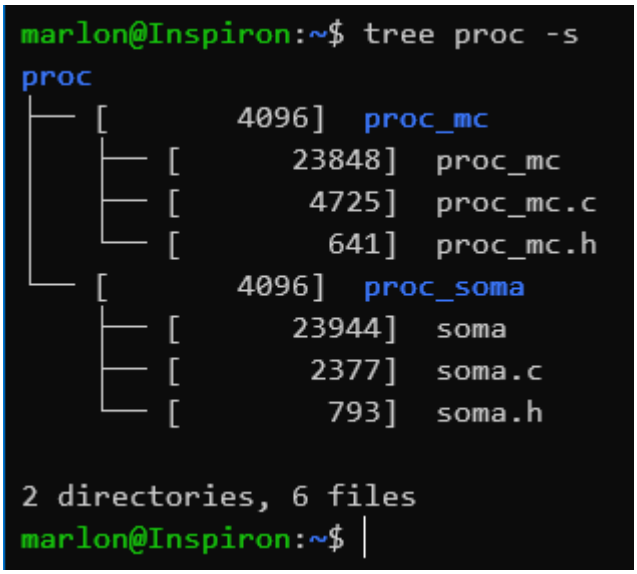
2º TP
Infodir - IPC e Multithread

Sexta-feira, 25 de novembro de 2022.

Desenvolva um programa chamado infodir para calcular o número total de arquivos, subdiretórios e o tamanho total, em *bytes*, dos arquivos de um diretório, incluindo todos os arquivos dos subdiretórios. Considerando o exemplo da figura abaixo, se o diretório informado pelo usuário for proc, o programa deve exibir como resultado o seguinte:

```
Arquivos = 6
Subdiretórios = 2
Tamanho do diretório = 56.328 bytes
```



Os nomes na cor azul representam os subdiretórios do diretório proc. Observe que todos possuem o mesmo tamanho de 4.096 *bytes*, o que não representa o tamanho real do diretório, porque ele não corresponde ao somatório dos tamanhos de cada arquivo do diretório. Por exemplo, o subdiretório proc_mc possui três arquivos (proc_mc, proc_mc.c e proc_mc.h), que somados os seus tamanhos obtém-se o tamanho real desse subdiretório como 29.214 *bytes*. O diretório proc_soma possui três arquivos (soma, soma.c e soma.h) que juntos totalizam 27.114 *bytes*. Somando esses valores obtém-se os 56.328 *bytes* do diretório proc.

O funcionamento do programa deve realizar os seguintes passos:

1. O usuário digita na linha de comando do sistema operacional o nome do programa infodir e passa como argumento o diretório que ele deseja obter os dados, como no exemplo abaixo.

```
marlon@Inspiron:~$ ./infodir proc
```

2. Para cada subdiretório que o programa infodir (processo pai) encontrar no diretório deve-se criar um processo filho para calcular o número total de arquivos, subdiretórios e o tamanho total dos arquivos do diretório, incluindo todos os arquivos dos subdiretórios. É função do infodir (processo pai) apenas ler o conteúdo do diretório informado pelo usuário, criar os processos filhos e exibir os relatórios abaixo com os resultados do programa. Se o diretório não possuir nenhum subdiretório, apenas arquivos, o infodir deve criar um único processo filho para realizar os cálculos.

3. O processo pai deve compartilhar uma região de memória com os processos filhos. Essa área de memória será usada pelos processos filhos escreverem o número total de arquivos, subdiretórios e o tamanho total calculado dos subdiretórios. O processo pai deve acessar essa

área de memória, após a conclusão dos processos filhos, para somar todos esses valores de cada subdiretório de modo a obter a quantidade total de arquivos, diretórios e o tamanho do diretório.

4. O infodir antes de acessar o diretório para ler o seu conteúdo e de iniciar a criação do(s) processo(s) filho(s), deve obter o tempo inicial e após calcular a quantidade total de arquivos, diretórios e o tamanho do diretório, exibir o tempo final e o tempo gasto para realizar esses cálculos. O resultado desse processamento deve ser exibido em um relatório com o seguinte leiaute:

```
- Método: IPC - Interprocess Communication
- Diretório: /home/marlon/proc

- Conteúdo do diretório
    Arquivos = 6
    Subdiretórios = 2
    Tamanho do diretório = 56.328 bytes

- Tempo usando IPC
    Início.....: 14:25:35
    Término: 14:25:59
    Duração: 24 segundos
```

   O programa deve exibir o tempo de início e término no formato hh:mm:ss e a duração em segundos.

5. Após exibir o relatório usando o método de comunicação entre processos (memória compartilhada), o infodir deve repetir os passos 2 a 4 mas usando outro método para calcular o número total de arquivos, subdiretórios e o tamanho total dos arquivos de um diretório. Esse método é a criação de múltiplas *threads* (*multithread*). Para cada subdiretório que o programa encontrar no diretório deve-se criar uma *thread*. Uma região de memória do processo deve ser compartilhada com suas *threads*. Essa área de memória será usada pelas *threads* para escreverem o número total de arquivos, subdiretórios e o tamanho total calculado dos subdiretórios. O processo deve acessar essa área de memória, após a conclusão das *threads*, para somar todos esses valores de cada subdiretório de modo a obter a quantidade total de arquivos, diretórios e o tamanho do diretório.

   ☞ **Atenção:** Não será considerado o fluxo de execução principal do processo como *thread*, somente as *threads* criadas com a função thrd_create[1] da biblioteca de *threads* da Linguagem C. Portanto, toda a implementação *multithread* do infodir deve usar apenas as funções da biblioteca de *threads* introduzidas no C11, sendo assim, não use as funções *Pthreads* (POSIX Threads).

   Como as *threads* são consideradas processos leves, espera-se que o tempo gasto no processamento *multithread* seja menor do que no IPC, o que será verificado quando o infodir exibir o relatório usando o leiaute abaixo.

[1] Documentação da biblioteca de *threads* da Linguagem C padrão ISO: https://en.cppreference.com/w/c/thread

- Método: *Multithread*
- Diretório: /home/marlon/proc

- Conteúdo do diretório
  Arquivos = 6
  Subdiretórios = 2
  Tamanho do diretório = 56.328 bytes

- Tempo usando *Multithread*
  Início.....: 14:26:02
  Término: 14:26:12
  Duração: 10 segundos

- Critérios de avaliação

O trabalho será avaliado considerando:

1. Estrutura da solução:

   a. A validação dos dados fornecidos pelo usuário.
   b. A lógica empregada na solução do problema.
   c. O funcionamento do programa.
   d. Escrever funções específicas, ou seja, com atribuição clara e objetiva.
   e. Não escrever código redundante.
   f. Código-fonte sem erros e sem advertências do compilador.
   g. Código-fonte legível, indentado, organizado e comentado.
   h. Identificadores significativos para aprimorar a inteligibilidade do código-fonte.

2. O programa deve ser desenvolvido no sistema operacional Linux usando apenas a Linguagem C padrão ISO[2] e a GNU C *Library*[3], que inclui a API POSIX. Programas desenvolvidos em outras linguagens, mesmo que parcialmente, receberão nota zero.

   O Capítulo 14. *File System Interface* do Manual de Referência da Biblioteca da Linguagem C do projeto GNU, disponível em https://www.gnu.org/software/libc/manual, é uma ótima referência para entender, por exemplo, como acessar o conteúdo de um diretório, pois há exemplo de programa e uma descrição detalhada das funções e estruturas usadas.

3. Para que o programa seja avaliado, o código deve executar com sucesso. Programas que apresentarem erros de compilação, ligação ou *segmentation fault* receberão nota zero.

4. Trabalhos com plágio, ou seja, programas com código fonte copiados de outra pessoa (cópia integral ou parcial) receberão nota zero.

5. O desenvolvimento do trabalho é individual.

[2] Documentação da Linguagem C padrão ISO: https://en.cppreference.com/w/c
[3] Documentação da GNU C *Library*: https://www.gnu.org/software/libc/

6. O programa deve ser composto de um único arquivo de cabeçalho (infodir.h) e um único código-fonte (infodir.c).

7. O código-fonte (.c) e o arquivo de cabeçalho (.h) devem ser codificados como UTF-8.

8. É proibido modificar os nomes de arquivos, identificadores, os protótipos de função, as declarações e/ou definições de funções fornecidos neste texto ou em anexo.

- Instruções para entrega do trabalho

1. Compacte os dois arquivos para criar um arquivo 7z com o seu nome e sobrenome, por exemplo: NelsonMandela. Use o software livre 7-Zip, que está disponível em https://www.7-zip.org/download.html.

2. Envie o arquivo 7z para o e-mail marlon.silva@ifsudestemg.edu.br.

- Data de entrega

Segunda-feira, 5 de dezembro de 2022 até às 23:59.

- Valor do trabalho

10,0 pontos

Prof. Márlon Oliveira da Silva
marlon.silva@ifsudestemg.edu.br