

UFRN IMD BTI

Professor: Wellington Souza

Introdução às Técnicas de Programação – Turma 03 – 2020.6

Grupo: André Augusto Fernandes - Matrícula: 20200050162

TRABALHO FINAL

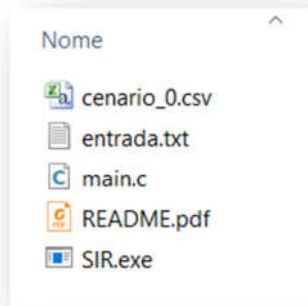
Implementação do Modelo Epidemiológico Computacional SIR em C

CHECKPOINT 1 – 01/12/2020

1. Introdução

Nesta etapa está sendo entregue uma versão preliminar do código, com todos os comandos definidos dentro do main e sem uso de módulos e cabeçalhos externos. Um uso mais extensivo dos conceitos aprendidos durante o curso será apresentado nas próximas entregas.

2. Conteúdo do ZIP



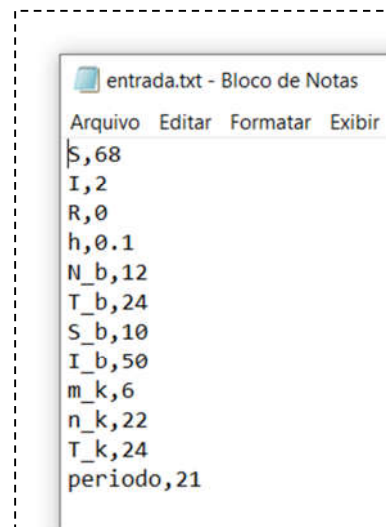
- cenario_0.csv: arquivo CSV com tabela gerada pelo modelo SIR;
- entrada.txt: arquivo TXT com os parâmetros iniciais de simulação;
- main.c: código;
- README.pdf: arquivo PDF com orientações sobre a atividade;
- SIR.exe: executável.

3. Parâmetro Iniciais

Os parâmetros iniciais deverão de fornecidos ao programa através de um arquivo texto formatado da seguinte forma:

"identificação da variável","valor da variável"
"identificação da variável","valor da variável"
"identificação da variável","valor da variável"
.
.
.

Exemplo:



3.2. Descrição dos Parâmetros

S: número de indivíduos suscetíveis (que ainda não estão contaminados);

I: número de indivíduos infectados (capazes de infectar indivíduos S);

R: número de indivíduos removidos (que se recuperaram, tornaram-se imunes ou faleceram);

h: pequeno intervalo de tempo entre cálculos da projeção (em horas);

N_b: número de pessoas suscetíveis que se infectaram em uma observação b (comportamento de contágio) de um período já decorrido;

T_b: intervalo de tempo da observação b (em horas);

S_b: número de pessoas suscetíveis no início da observação b;

I_b: número de pessoas infectadas no início da observação b;

T_k: intervalo de tempo de observação k (comportamento de recuperação) de um período já decorrido (em horas);

n_k: indivíduos infectados em um intervalo de tempo T_k;

m_k: indivíduos recuperados de um total n_k após intervalo de tempo t_k;

periodo: intervalo de tempo da modelagem (em dias).

4. Compilação no Terminal

Usando um terminal ou prompt de comando, dirija-se à pasta onde foi descompactado o ZIP e digite:

```
gcc -o nome_do_executável main.c
```

Exemplo:

```
gcc -o SIR main.c
```

Será criado um arquivo executável com o nome fornecido pelo usuário. No caso do exemplo, foi criado SIR.exe.

5. Executando o programa

Para executar o programa, ainda usando o terminal, digite:

```
./nome_do_executável código_do_cenário arquivo_de_entrada
```

As opções de **cenários** possíveis são **0** (padrão), **1** (distanciamento/máscaras), **2** (melhoria nos protocolos de atendimento) ou **3** para todos os cenários.

OBS: esta versão do código só possui o cenário 0 implementado.

Exemplo:

```
./SIR 0 entrada.txt
```

5.2. Comportamento esperado

O programa deverá imprimir os resultados no terminal e em um arquivo chamado `cenario_0.csv`.

6. Modificações para o próximo Checkpoint

Nas próximas entregas serão implementados:

- Structs para armazenar as variáveis utilizadas no modelo;
- Alocação dinâmica de ponteiro para struct;
- Funções para realizar as operações atualmente contidas no main;
- Modularização: fragmentação do código em módulos, cabeçalhos e main.

Segue abaixo uma prévia dos protótipos a serem implementados:

```
typedef struct {  
    /*campos*/  
} cenarios;  
  
void calcula_b_k(cenarios *v, int indice);  
  
void calcula_sir(cenarios *v, int indice);  
  
void le_arquivo(cenarios *v, FILE *f);  
  
void imprime_arquivo(FILE *f, cenarios *v, int indice);
```

7. Limitações do programa

Tive um problema com a função `setlocale(LC_ALL, "Portuguese")`. Ela funcionou pra imprimir o idioma de forma correta, mas provocou um comportamento adverso no restante do programa, então desativei. A solução que utilizei foi fazer a conversão de decimal pra caractere na hora da impressão.