Nome: Beatriz Viana Costa NUSP: 13673214

Relatório EP1 – MAC0216 – Técnicas de Programação:

1. **Especificações do o hardware utilizado nos testes:**

i. Sistema:

Kernel: 5.4.0-91-generic x86_64 bits: 64 Compilador: gcc v: 9.3.0

Desktop: Cinnamon 5.2.7 Distro: Linux Mint 20.3 Una

Base: Ubuntu 20.04 focal

ii. Máquina:

Laptop Dell Inspiron 5458

iii. CPU:

Intel Core i5-5200U, Dual Core

2. **Tabela com o tempo de execução em cada linguagem:**

ENTRADA	MÉDIA C	MÉDIA ASSEMBLY
10000000	0m 0.42s	0m 0.47s
110111120	0m 0.46s	0m 0.47s
210222220	0m 0.82s	0m 0.91s
310333358	0m 1.27s	0m 1.34s
410444432	0m 1.60s	0m 1.81s
510555560	0m 2.02s	0m 2.23s
610666660	0m 2.36s	0m 2.77s
710777961	0m 2.81s	0m 3.16s
810888972	0m 3.20s	0m 3.51s
999999936	0m 3.89s	0m 4.33s

Tabela 1: Tempo de execução em segundos (média de dez execuções para cada entrada).

3. **Tabela com o tamanho de cada executável:**

C	ASSEMBLY
16.904 bytes	11.672 bytes

Tabela 2: Tamanhos dos executáveis em bytes.

4. **Conclusão:**

Os resultados obtidos não foram exatamente como o esperado, pelo tamanho dos programas – em assembly com 246 linhas e em *C* com 58 – esperava que o programa em C fosse menor, contudo, o executável em *C* se mostrou maior.

Além disso, acreditava que o tempo de execução do programa em *assembly* fosse ser menor do que o em linguagem *C*, porém em nenhum dos testes o tempo de execução em *assembly* foi menor.

Provavelmente tem relação com o fato de que no código em *assembly* houveram várias movimentações de valores para registradores e funções a mais, como no caso do *input* e *output*, que na linguagem *C* já podem ser realizadas de maneira automática com a biblioteca *stdio.h*, precisando utilizar somente as funções *scanf()* e *printf()*, e em assembly é necessário ler/imprimir a *string* fazendo a conversão de cada número de *ASCII* para inteiro e vice-versa.