

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA**

**RODRIGO DE CASTRO MICHELASSI
NUSP: 13672703**

Relatório EP1 - Operações sobre primos

**SÃO PAULO
2022**

1. Introdução

Nesse EP, a ideia era que fosse desenvolvido um algoritmo eficiente, nas linguagens C e Assembly x86_64, que trabalhasse certas operações envolvendo números primos, as quais deveriam, de acordo com o modo de entrada do usuário, imprimir na tela:

- I. O próximo número primo, a partir do número inserido na entrada.
- II. Caso o número de entrada seja múltiplo de dois números primos, imprimir esses números.

A proposta central do EP era desenvolver as habilidades aprendidas em sala do uso da linguagem Assembly e testemunhar a diferença do peso, em bytes, entre o mesmo programa em linguagem de máquina e em uma linguagem de alto nível.

A máquina usada para os testes analisados a seguir possui as seguintes especificações:

Sistema Operacional: Ubuntu 21.10

Processador: Intel(R) Core(TM) i7-8565U CPU @ 1.80GH

Memória: 8GB (2*4GB SODIMM DDR4 Synchronous 2400 MHz) RAM

2. Tempos de Execução

Entrada	Média (C)	Média (Assembly)
10000000	0m 2.43s	0m 2.41s
110111120	0m 0.76s	0m 2.58s
210222220	0m 0.94s	0m 1.86s
310333358	0m 0.71s	0m 1.90s
410444432	0m 0.70s	0m 1.44s
510555560	0m 0.76s	0m 1.40s
610666660	0m 0.70s	0m 1.82s
710777961	0m 0.48s	0m 1.37s
810888972	0m 0.72s	0m 1.38s
999999936	0m 0.38s	0m 1.22s

Tabela 1: Tempo de execução médio em segundos

3. Tamanhos dos Executáveis

C	Assembly
18.136 bytes	8.560 bytes

Tabela 2: Tamanhos dos executáveis em bytes

4. Considerações Finais

Por fim, os resultados obtidos não ficaram dentro do esperado, mesmo com os algoritmos implementados em ambas linguagens seguirem a mesma lógica. Nota-se que o programa em Assembly, sendo mais leve que o programa em C (cerca de 10.000 bytes mais leve), mesmo assim passou por um tempo de execução médio maior que o programa em C, quando ele, por ser muito mais leve, deveria ter levado um tempo menor para a execução.

Nota-se também que o programa em C não faz as conversões de entrada String para Binário, e para a saída, de Binário para String, motivo o qual pode ter sido o causador dessa divergência nos tempos de execução.

Dessa forma, embora os resultados obtidos tenham sido fora do esperado, ambos os programas funcionam ainda em uma velocidade alta para entradas grandes, e retornam os resultados esperados para as entradas inseridas.