Pedro Cruz Martins 11965859

Relatório EP1 - Técnicas de Programação 1

Programa que calcula um hashing de uma string de entrada até 100000 caractéres. O programa conta com duas implementações em linguagens diferentes, em Python, linguagem interpretada de alto nível e em Assembly ,linguagem de montagem de baixo nível.

Resultados:

Entrada	**Média (Python)**	**Média (Assembly)**
texto100000.txt	0.343629818s	0,01s
texto10000.txt	0.035553928s	0,05s
texto1000.txt	0.004046368s	0.00s
texto100.txt	0.00s	0.00s
texto10.txt	0.00s	0.00s

obs: só era possível ver até duas casas decimais a performance do tempo de execução do código em assembly

Tamanhos:

Arquivo Python	Arquivo Assembly
3412 bytes	14199 bytes

De forma inesperada, uma vez que Python é uma linguagem de nível maior que assembly, o arquivo .s é menor do que o arquivo .py.

Rodado com: AMD Ryzen 7 5800H with Radeon Graphics, com 2,78GB de memória, no terminal do WSL com Ubuntu 22.04.3.

Conclusão:

Como podemos ver pelos tempos de execução, o código em assembly parece não aumentar tanto o tempo conforme os vetores de teste vão aumentando de tamanho, a diferença entre a performance dos códigos para o “texto100000.txt” é enorme, porém, para casos pequenos como “texto10.txt” ou “texto100.txt” os tempos de execução são muito semelhantes, havendo pouca diferença entre os códigos.

Observações:

- O programa parece não estar funcionando corretamente para os testes de 10000 chars e 100000 chars, pelo observado parece um problema de limpeza de memória, pois toda vez que é rodado o executável temos uma saída diferente.

- O código Python está imbutido com a biblioteca time para a medição do seu tempo de execução