

Relatório - Exercício Programa #03

MAC0216 - Técnicas de Programação I

Professor Daniel Macêdo Batista

Instituto de Matemática e Estatística - IME USP

Aluno: Francisco Nassif Membrive

Data de Entrega: 11.12.2023

1. Programa

Este exercício-programa, o terceiro da disciplina de MAC0216, Técnicas de Programação I, corresponde a duas bibliotecas implementadas em C. Uma biblioteca estática, hashliza.h, que possui as funções necessárias para criptografar uma string em um hash de 32 caracteres hexadecimais. E uma biblioteca dinâmica, shannon.h, que possui uma única função capaz de calcular a entropia de Shannon de uma string a partir da string e de uma base para ser usada no cálculo do logaritmo. Foram utilizadas na implementação as bibliotecas padrão <stdio.h>, <stdlib.h>, <string.h> e <math.h>. As duas bibliotecas são utilizadas para calcular a entropia de Shannon e gerar o hash de 10 strings diferentes com o mesmo tamanho: 100 caracteres. Isso foi feito em um programa testa.c que registrou o tempo máximo, mínimo e médio de execução de cada função a partir da biblioteca <sys/time.h>, que serão exibidos na tabela na seção seguinte. A compilação das bibliotecas, bem como do programa teste.c e a geração da documentação html via doxygen pode ser feita através do script compila.sh, entregue em conjunto com os códigos.

2. Desempenho de cada função em microssegundos

Função	Média	Mínimo	Máximo
ep1Passo1Preenche	0.2 μ s	0.0 μ s	1.0 μ s
ep1Passo2XOR	2.3 μ s	2.0 μ s	3.0 μ s
ep1Passo3Comprime	28.0 μ s	27.0 μ s	30.0 μ s
ep1Passo4Hash	0.2 μ s	0.0 μ s	1.0 μ s
ep1Passo4HashEmHexa	0.6 μ s	0.0 μ s	1.0 μ s
ep3CriaVetorMagico	34.0 μ s	4.0 μ s	290.0 μ s
ep3CalculaEntropiaShannon	2.6 μ s	1.0 μ s	10.0 μ s

3. Sistema e conclusões

O código foi escrito utilizando o Visual Studio Code e rodado no Ubuntu 22.04 (Máquina Virtual) de 64 bits, com processador Ryzen 5 3600 utilizando 1 núcleo de processamento através da Oracle VM VirtualBox e 16GB de memória RAM. Os tempos medidos foram condizentes com o tamanho das strings dos testes. Com o passo três sendo o mais demorado por ter mais laços aninhados. A função mais demorada foi a que cria os vetores mágicos por ter que percorrer dois laços de

tamanho 256. Como as demais funções dependem do tamanho da string, seus tempos superariam o de execução de `ep3CriaVetorMagico` caso fosse executado o teste com strings maiores (`ep3CriaVetorMagico` tem tempo de execucao constante). Além disso, as funções do passo também tem tempo esperado constante, já que operam apenas strings de tamanho 48 e 16, respectivamente. A primeira parte do passo quatro foi a mais rápida pois sua única operação é copiar 16 caracteres de uma string para outra, tudo conforme o esperado.