

## EP3: Bibliotecas estática e dinâmica em linguagem C

Ana Livia Rüegger Saldanha  
Nº USP: 8586691

11 de dezembro de 2023

### Resumo

Este relatório apresenta uma breve análise de desempenho, em termos de tempo de execução, de cada uma das funções das bibliotecas desenvolvidas em linguagem C para o Exercício Programa 3 da disciplina MAC0216.

---

## 1 INTRODUÇÃO

O projeto a que este relatório se refere, realizado como Exercício Programa para a disciplina MAC0216 do IME-USP, apresenta implementações e interfaces de duas bibliotecas em C:

1. Biblioteca estática 'hashliza', com uma série de funções correspondentes a uma sequência de passos para geração de um hash em hexadecimal para a string fornecida. Em conjunto, os passos/funções executam o mesmo algoritmo implementado no EP1 de MAC0216;
2. Biblioteca dinâmica 'shannon', contendo uma função que calcula a Entropia de Shannon para uma dada string, com a base de logaritmo especificada.

Ainda, o projeto inclui um programa de teste automatizado implementado no arquivo `testa.c` e um script para compilação das bibliotecas e desse programa, além da geração de uma documentação no padrão Doxygen. Também no `testa.c`, foi implementada uma função que mede os tempos de execução para 10 chamadas de cada uma das funções contidas em cada biblioteca. A seguir, serão apresentados os resultados dessas medições.

## 2 DESEMPENHO

Nesta seção, apresentamos os dados obtidos experimentalmente a fim de demonstrar o desempenho das funções.

### 2.1 CONFIGURAÇÕES

Antes de apresentarmos os resultados das medições, cabe especificar quais são as configurações do computador no qual os experimentos foram executados, ressaltando que todas as medições foram realizadas sob as mesmas condições. O computador em questão possui processador AMD Ryzen 5 5500U, 8GB de RAM e sistema operacional GNU/Linux x86\_64 (distribuição Linux Mint 20.3; kernel Linux 5.8.0-63-generic).

## 2.2 DESEMPENHO POR FUNÇÃO

As medições apresentadas abaixo (Tabela 1), de tempo médio, mínimo e máximo de execução, foram tomadas de um conjunto de 10 execuções para cada função. Para medição do tempo, foi utilizada a função `clock()` da biblioteca `time.h`, registrando os momentos (`clock_t`) imediatamente antes à chamada da função e imediatamente depois, e, por fim, calculando o tempo de execução como a diferença entre esses dois instantes dividida pela constante `CLOCKS_PER_SEC` (também da biblioteca `time.h`). Cabe também ressaltar que as chamadas das funções foram feitas sempre com strings de mesmo tamanho.

Função	Média	Mínimo	Máximo
ep1Passo1Preenche	0.0000003	0.0000000	0.0000010
ep1Passo2XOR	0.0000019	0.0000010	0.0000020
ep1Passo3Comprime	0.0000562	0.0000560	0.0000570
ep1Passo4Hash	0.0000008	0.0000000	0.0000010
ep1Passo4HashEmHexa	0.0000044	0.0000040	0.0000050
ep3CriaVetorMagico	0.0000109	0.0000100	0.0000110
ep3CalculaEntropiaShannon	0.0000046	0.0000040	0.0000050

Tabela 1: Tempos de execução médio, mínimo e máximo, em segundos, para cada função, considerando 10 execuções.

## 3 CONCLUSÃO

Como pode ser observado na Tabela 1, as variações de tempo de execução para diferentes funções correspondem ao esperado: funções mais complexas, como a função `ep1Passo3Comprime` que executa diversos loops aninhados, têm maior tempo de execução; funções que executam menos operações, como a `ep1Passo1Preenche` e a `ep1Passo4Hash`, apresentam menor tempo de execução. Em relação às execuções de uma mesma função, é possível observar alguma variação quando comparamos tempos mínimos e máximos, mas essa variação não ultrapassa 0.000001 segundo em nenhum caso (o que é esperado, visto que as execuções foram realizadas para strings do mesmo tamanho).