Atividade 3 - Desenvolvimento de Código Otimizado

Allan Garcia Cavalcante e Silva - 13731222
Eduarda Fritzen Neumann - 12556973
Lucas Eduardo Gulka Pulcinelli - 12547336
Teo Sobrino Alves - 12557192

Novembro 2023

1 Introdução

No presente trabalho faremos uma comparação de tamanho e tempo de execução utilizando diferentes flags de otimização parra programas disponibilizados no site *benchmark games*. Os programas utilizados foram fannkuch-redux e n-body.

2 Desenvolvimento

Os níveis de otimização utilizados foram `-O0` (sem otimização) até `-O3` (nível máximo de otimização que garante a corretude do binário gerado) e `-Os`, que otimiza o tamanho do binário gerado. A diretiva de compilação utilizada foi `gcc -Wall -Ox -o binario` e eventuais flags para bibliotecas necessárias para os programas.

Para a análise do tempo foram realizadas 10 repetições e calculada a média, excluindo o menor e maior tempo de execução, utilizando a função time do Linux, para a entrada foram utilizados os valores sugeridos no site. Para a medição do tamanho, foi realizada a verificação do tamanho em bytes do binário gerado através do comando `ls -l`.

O código foi executado numa CPU intel Core i5-8265U. Não foram utilizadas flags de otimização voltadas para arquitetura (ex: `-march=native`).

3 Resultados

tabela 1: Resultados para o programa fannkuch-redux com o valor 12:

nível de otimização	-O0	-O1	-O2	-O3	-Os
tempo de compilação (s)	0.20	0.16	0.19	0.29	0.19
tempo de execução (s)	26.6	4.18	4.50	4.30	4.43
tamanho (bytes)	24496	16312	16312	16312	16312

tabela 2: Resultados para o programa n-body com o valor 50000:

nível de otimização	-O0	-O1	-O2	-O3	-Os
tempo de compilação (s)	0.20	0.28	0.29	0.28	0.29
tempo de execução (s)	14.89	2.22	2.26	1.74	2.45
tamanho (bytes)	20288	16120	16120	16120	16096

4 Análise dos Resultados

Como podemos ver as diretivas de otimização de performance sempre obtiveram vantagens no tempo sobre o código sem otimização e também possuem tamanho menor. O código gerado com a otimização de `-Os` oferece o mesmo nível de otimização que `-O2`, porém visando não utilizar funções *inline* e outros fatores que podem causar aumentos no número de instruções e assim pode reduzir o tamanho do código, como no caso do segundo programa, as otimizações quase sempre trazem um aumento no tempo de compilação.

O nível de otimização mais alto nem sempre produz o código mais rápido, o que se deve provavelmente à especificidades da arquitetura, já que o binário gerado poderá ter tempo de execução diferente devido ao tempo de execução de instruções específicas.

5 Conclusão

O uso das otimizações sempre trouxe benefícios em relação ao tempo de execução e tamanho do código em comparação com nenhuma otimização, sempre com pequeno aumento no tempo de compilação se comparado com o ganho de tempo na execução.