SSC0951 – Desenvolvimento de Código Otimizado

Otimizações do compilador

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Otimizações do compilador

Opções:

-O0

No optimization (the default); generates unoptimized code but has the fastest compilation time.

Note that many other compilers do fairly extensive optimization even if “no optimization” is specified.

With gcc, it is very unusual to use -O0 for production if execution time is of any concern, since -O0 really does mean no optimization at all. This difference between gcc and other compilers should be kept in mind when doing performance comparisons.

-O1

Moderate optimization; optimizes reasonably well but does not degrade compilation time significantly.

-O2

Full optimization; generates highly optimized code and has the slowest compilation time.

-O3

Full optimization as in -O2; also uses more aggressive automatic inlining of subprograms within a unit and attempts to vectorize loops.

-Os

Optimize space usage (code and data) of resulting program.

Otimizações do compilador

option	optimization level	execution time	code size	memory usage	compile time
-O0	optimization for compilation time (default)	+	+	-	-
-O1 or -O	optimization for code size and execution time	-	-	+	+
-O2	optimization more for code size and execution time	--		+	++
-O3	optimization more for code size and execution time	---		+	+++
-Os	optimization for code size		--		++
-Ofast	O3 with fast none accurate math calculations	---		+	+++

+increase ++increase more +++increase even more -reduce --reduce more ---reduce even more

Otimizações do compilador

Algumas flags importantes (vistas na Aula 5):

-finline-functions

-ftree-loop-vectorizer

-fif-conversion

Outras opções que afetam desempenho:

-march

-mtune e -mcpu

Lista completa:

https://gcc.gnu.org/onlinedocs/gcc-7.2.0/gcc/Optimize-Options.html

Qual conjunto de flags trará o melhor desempenho para um determinado código?

Otimizações do compilador

Estudo:

Finding Best Compiler Options for Critical Software Using Parallel Algorithms

International Symposium on Intelligent and Distributed Computing, 2018

Utiliza:

The Computer Language Benchmarks Game (CLBG)

https://benchmarksgame-team.pages.debian.net/benchmarksgame/

Otimizações do compilador

Otimização depende também do compilador

Opções de compilador:

- gcc
- clang
- icc (Intel)

Otimizações do compilador

Atividade:

- Escolher uma arquitetura (pode ser Intel, AMD, ARM) e compilar com a diretiva da arquitetura que está executando (-march)
- Escolher um dos programas do CLBG
- Compilar com todas as opções de otimização (sem otimização, O1, O2 e O3)
- Compilar com as flags encontradas no artigo para o programa escolhido para a análise considerando as flags que aparecerem em 80% ou mais das soluções com melhor desempenho
- Comparar os resultados