

# Processo de compilação cruzada

Levy G. S. Galvão

O tradicional processo de compilação cruzada em Linux segue os passos abaixo:

1. Compilar **binutils**;

Esta constitui-se como uma coleção de ferramentas binárias, como o vinculador **ld** ou o assembler **as**. Também existem ferramentas de análise e depuração. As ferramentas devem ser configuradas para cada arquitetura de CPU.

2. Compilar as dependências do **gcc**: **mpfr**, **gmp**, **mpc**;

A biblioteca **mpfr** (*multiple-precision floating-point computations*) é utilizada para substituir chamadas às funções matemáticas em tempo de compilação. **gmp** é uma dependência do **mpfr**. Já a **mpc** é utilizada em operações matemáticas que envolvem números complexos.

3. Instalar cabeçalhos do kernel do Linux;

Estes cabeçalhos se constituem de definições numéricas de chamadas do sistema, várias estruturas e definições.

4. Compilar o primeiro estágio do **gcc**: neste passo permitindo suporte à vinculação estática e sem suporte a biblioteca C;

O **gcc** é o *GNU compiler collection* que serve de frente a várias linguagens como C, C++, Fortran, etc; e suporta várias arquiteturas de CPU. Esta fornece os compiladores, os motores de compilação, **binutils**, o assembler e o vinculador. Além disso também provê várias bibliotecas essenciais.

5. Compilar a biblioteca C usando o primeiro estágio do **gcc**;

A biblioteca C é a *GNU C library* e em sua completude possui as bibliotecas padrões do Linux C e utilizada amplamente em *desktops* e servidores. Suporta várias arquiteturas e sistemas operacionais.

6. Compilação final com o **gcc**, biblioteca C e suporte à vinculação dinâmica;

Ao final, se necessária a vinculação de bibliotecas dinâmicas, estas são feitas nessa etapa.