

O que é um cross compiling toolchain?

É um conjunto de ferramentas que permite construir código fonte em código binário para uma plataforma diferente daquela em que a construção foi feita.

E além de uma plataforma diferente, este conjunto de ferramentas necessita de:

- Arquitetura de CPU diferente
- ABI diferente
- Sistema operacional diferente
- Biblioteca C diferente

Este processo envolve três máquinas diferentes:

- **Build:** Lugar da construção
- **Host:** Onde a execução fica
- **Target:** Para cada programa gerar código

O toolchain nativo tem a seguinte sequência lógica:

build == host == target

Porém o processo estudado aqui possui a sequência da compilação cruzada que possui uma lógica diferente onde:

build == host != target

Como conseguir uma cross compilation toolchain?

Primeiramente um target chamado *autoconf* define as configurações gerais do sistema em tuplas divididas em tags que contém strings que descrevem componentes. As tuplas são geralmente estruturadas assim:

Arquitetura-Vendor-Toolchain-Biblioteca

Os toolchains aqui são especificados para definir se o desenvolvimento é sem sistema operacional ou para o Linux. Se o processo é para Linux, se possui 4 componentes já pré-estabelecidos: gcc, binutils, Cabeçalhos do kernel do Linux, Biblioteca C. Cada um vai ser detalhado abaixo:

- **Binutils:** Estes são a coleção de ferramentas binárias onde se montam e ligam os arquivos da arquitetura e bibliotecas
- **GCC:** Provém a compilação em todos os níveis e as bibliotecas do target
- **Cabeçalhos do kernel do Linux:** Definem números de system calls, vários tipos de estruturas e definições.
- **Biblioteca C:** Provém a implementação de funções padrão do POSIX além de outros padrões e extensões. É baseado nas system calls do Linux.

Assim definidos os componentes para o processo no Linux, pode-se estabelecer como funciona o processo de compilação cruzada para toolchain utilizando este sistema operacional, o processo é o seguinte.

- Construir binutils
- Construir dependências gcc, mpfr, gmp e mpc
- Instalar os cabeçalhos do kernel do Linux
- Construir o primeiro estágio do gcc
- Construir a biblioteca C utilizando o processo anterior
- Construir o gcc final

Assim definido o passo a passo podemos definir como obter este processo, para isso é necessário ou obter uma distribuição Ubuntu e Debian que já possuem vários compiladores que utilizam deste processo, ou construir sua própria utilizando ferramentas de Crosstool-NG ou de sistemas de construção integrada do Linux.