UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
Curso de Graduação em Engenharia Mecatrônica

Sistemas Digitais para Mecatrônica

RELATÓRIO DA SEMANA 10 DE SISTEMAS DIGITAIS PARA MECATRÔNICA

Gabriel Augusto de Morais Batista 11421EMT007

Uberlândia

Março de 2022

Uma toolchain de compilação cruzada é um conjunto de ferramentas que possibilita a construção de um código fonte em código binário, para uma plataforma diferente daquela em que é realizada a construção do código, sendo que essa diferença pode ocorrer por diferenças de arquitetura da CPU, ABI, sistema operacional ou biblioteca C, por exemplo. Neste processo, são envolvidas 3 máquinas, que são:

- Máquina de construção (build machine): é onde acontece a construção do código;
- Máquina hospedeira (host machine): é onde acontece a execução do código;
- Máquina de destino (target machine): é a máquina para a qual os programas geram código;

Na toolchain nativa, os 3 papéis de máquina são realizados pela mesma máquina, enquanto que na toolchain de compilação cruzada, a máquina de construção e a máquina hospedeira são a mesma, porém a máquina de destino é diferente destas, e tais máquinas são definidas pelos parâmetros –build, --host e –target no script de configuração, que por meio do autoconf, por padrão, entende que sejam todos a máquina atual.

O autoconf define o conceito de definição de sistema, representados como tuplas, contendo a arquitetura de CPU, o sistema operacional, o vendedor, ABI e a biblioteca C, com o seguinte formato: <arch>-<vendor>-<os>-<libc/abi>. O campo de sistema operacional possui dois valores principais, que são none para cadeias de ferramentas “bare-metal” e linux para cadeias de ferramentas Linux. O valor none é utilizado em desenvolvimentos sem sistema operacional definido, a biblioteca C usada geralmente é a newlib, tem serviços de biblioteca C que não faz requisição de sistema operacional, permite fornecer chamadas básicas de sistema para diferentes hardwares de máquinas de destino, entre outros usos, enquanto que o valor Linux é usado para desenvolvimento em sistema operacional Linux, as bibliotecas C utilizadas são específicas do Linux e suporta as chamadas de sistema para o Linux.

Existem quatro componentes principais em uma toolchain de compilação cruzada do Linux, que são binutils, gcc, headers do kernel do Linux e a biblioteca C.

O binutils é algo como uma “coleção de ferramentas binárias”, sendo as principais o Id que vincula vários arquivos de objeto em uma biblioteca compartilhada, um executável ou outro arquivo objeto, e também o as, que recebe o código assembler específico da arquitetura em forma de texto e produz um arquivo objeto correspondente com código binário. Possui também ferramentas de depuração ou análise, como addr2line, ar, c++filt e

readelf, precisa ser configurada para cada arquitetura de CPU diferente e tem compilação cruzada bastante simples, por não necessitar de dependências especiais.

O gcc é a sigla para GNU Compiler Collection, e serve de front-end para muitas linguagens de origem, como C, C++, Fortran, mas também serve de back-end para muitas arquiteturas de CPU. O gcc fornece o próprio compilador, cc1 para C, cc1plus para C++, só gera código assembly em formato de texto, o driver do compilador, gcc, g++, que comanda o próprio compilador, mas também o binutils montador e linker, as bibliotecas de destino libgcc (tempo de execução do gcc), libstdc++ (a biblioteca C++), libgfortran (tempo de execução do Fortran) e arquivos header para a biblioteca C++ padrão. Construir o gcc é um pouco mais complicado do que construir o binutils, pois são necessários duas etapas.

Os headers do kernel do Linux são necessários para a construção de uma biblioteca C e são definições de números de chamada do sistema, vários tipos de estrutura e definições. No kernel, eles estão divididos em headers visíveis no espaço de usuário e headers internos do kernel. A versão do kernel usada para os headers, deve ser a mesma ou mais antiga do que a versão rodando na máquina de destino, para evitar que sejam usadas chamadas do sistema que não estão contidas naquele kernel.

A biblioteca C possibilita a implementação das funções padrão do POSIX, assim como vários outros padrões e extensões, e é baseada nas chamadas de sistema do Linux. Várias implementações estão disponíveis, como glibc, uClibc-ng, musl, bionic (para sistemas android) e algumas para propósitos especiais, como newlib, dietlibc e klibc. Após a compilação e instalação, fornece o linker dinâmico, a própria biblioteca C, as bibliotecas associadas a ela e as headers dela.

O processo de compilação de uma toolchain de compilação cruzada regular do Linux é, na verdade, fácil, e segue a seguinte sequência:

- Construção de binutils;
- Construção das dependências do gcc: mpfr, gmp, mpc;
- Instalação dos headers do kernel Linux;
- Construção de um gcc de primeiro estágio, sem suporte a biblioteca C e com suporte apenas a linkagem estática;
- Construção da biblioteca C à partir do gcc de primeiro estágio;
- Construção do gcc final, com suporte a biblioteca C e linkagem dinâmica.