



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA



Tarefa Semanal 10 – Sistemas Embarcados II

Compilação Cruzada

Fernando Rabelo Fernandes Junior - 11611EMT020

Uberlândia, fevereiro de 2022

Compilação Cruzada

A compilação cruzada ocorre quando um dispositivo compila um código fonte para uma plataforma diferente daquela que compilou o código, geralmente temos um ambiente de desenvolvimento que está em uma arquitetura diferente da arquitetura da placa em que vamos rodar nosso programa. Para que nossos notebooks ou desktops possam gerar um binário, programa que rode em outra arquitetura, acontece o Cross Compile (Compilação Cruzada, da tradução direta do inglês). Por exemplo, em nosso caso, um computador com Linux Ubuntu, utilizando um processador Intel ou AMD com arquitetura x86, irá compilar um código para o computador embarcado rodando um sistema Linux adaptado rodando em uma arquitetura completamente diferente.

Utilizar a compilação cruzada para trabalhar com sistemas embarcados é muito comum, principalmente quando estes não possuem capacidade de processamento para suportar um compilador. Para nós seria possível compilar no próprio sistema embarcado, porém, apesar de os recursos não serem tão limitados assim, não queremos desperdiçá-los. Além disso programar em um computador regular com a disposição de diversos tipos de IDE diferentes é muito melhor do que programar em um computador embarcado com recursos de interface limitadas.

Para realizarmos esse processo de compilação cruzada precisamos primeiro obter um software development kit, também conhecido como Kit de desenvolvimento de software, SDK ou devkit. O SDK nada mais é que um conjunto de ferramentas de desenvolvimento de software que possibilita a criação e compilação dos softwares para um sistema diferente.

Na área de sistemas embarcados, tanto micro-processados como micro-controlados, geralmente temos um ambiente de desenvolvimento que está em uma arquitetura diferente da arquitetura da placa em que vamos rodar nosso programa, como nossos notebooks ou desktops, que geralmente possuem processadores Intel ou AMD, arquitetura x86. Em contrapartida a Raspberry Pi é controlada por um processador com arquitetura ARM. O Arduino é controlado por um microcontrolador de arquitetura AVR e assim por diante. Para que nossos notebooks ou desktops Intel possam gerar um binário, programa que rode em outra arquitetura, acontece o “Cross Compile” (Compilação Cruzada da tradução direta do inglês). Toda vez que você realiza o upload de um Sketch pela Arduino IDE acontece uma compilação cruzada.

Exemplificando: Pense em duas pessoas, uma é brasileira, ou seja, sua língua nativa é o português, e uma é norte americana que normalmente fala inglês, lê e entende essa língua nativamente. Agora pense que eu tenho um livro escrito em grego, é em grego pra dificultar as coisas mesmo, e eu traduzo esse livro para português e entrego essa tradução para o brasileiro e pro norte americano. O brasileiro vai conseguir ler a tradução normalmente sem problemas, já o nosso amigo norte americano vai continuar não entendendo nada do que está escrito em português.

O mesmo aconteceu no nosso exemplo acima em que tentamos rodar o binário na Raspberry Pi. Eu tenho um programa escrito em C, mas meu processador não entende C. Então eu tenho que “traduzir”, ou seja, compilar, esse programa em instruções que o meu

processador entenda. Quando eu faço isso no meu computador Intel ele compila instruções que processadores Intel entendam. Agora se eu tentar rodar essas mesmas instruções na Raspberry Pi, que tem um processador ARM, ele não vai entender e por consequência retornar com “Exec format error”.

É isso que a compilação cruzada resolve, nós vamos ter um compilador que entende as instruções Intel para gerar instruções ARM, o chamado compilador cruzado (do inglês cross compiler). Então primeiramente precisamos de um compilador cruzado.

Para o desenvolvimento de aplicações embarcadas, a utilização da compilação cruzada é uma técnica bem comum. Como dito no começo do artigo, geralmente sempre temos um ambiente de desenvolvimento que está em uma arquitetura diferente da arquitetura da placa em que vamos rodar nosso programa. Mesmo que por de baixo dos panos, no caso do Arduino IDE que utiliza o avr-gcc para realizar a compilação cruzada para o ATmega328P do Arduino UNO.