

Universidade Federal de Uberlândia  
Engenharia de Controle e Automação / Engenharia Mecatrônica  
Sistemas Embarcados II / Sistemas Digitais para Mecatrônica  
Prof. Éder Alves de Moura  
Semana 10 – Compilação Cruzada



23/02/2022

### Roteiro de Atividades

Crie uma pasta em seu repositório GitHub, denominada 'Semana10'. Nela colocará um documento pdf com as atividades dessa semana.

1. Faça um resumo do processo de compilação cruzada descrito no vídeo:

- Anatomy of Cross-Compilation Toolchains

[https://www.youtube.com/watch?v=Z1RJmh\\_OqeA](https://www.youtube.com/watch?v=Z1RJmh_OqeA)

O processo de compilação, desenvolvido na atividade da semana 9, é feito usando Flask no Python. A aplicação app é definida como um objeto Flask com parâmetro o nome `__name__` do programa usado por python, no código chamado app.py. O endereço da aplicação Flask é definida com a função `@app.route('/')`. Depois disso, cabe definir o programa principal numa função denominada `index()` e lançar o programa usando o método `run(debug=True)` sobre o objeto de classe Flask. A função app.py vai compilar os códigos html e css usado no programa index.html.

2. Faça um roteiro de explicação e instalação, ao estilo tutorial, do processo das ferramentas de toolchain para a compilação cruzada para sistemas embarcados, a partir dos dois primeiros vídeos da lista:

- Embedded Linux | Embedded Linux ToolChain | Beginners

[https://www.youtube.com/watch?v=9CmZXuHdCUM&list=PLMB3ddm5Yvh1H\\_e4lqXyFH25efrNYEDnP&index=1](https://www.youtube.com/watch?v=9CmZXuHdCUM&list=PLMB3ddm5Yvh1H_e4lqXyFH25efrNYEDnP&index=1)

### LINUX EMBARCADO

#### O que é um toolchain?

Um toolchain é um conjunto de pacotes, que podem ser baixados ou construídos usando um gerador de toolchain, precisos para criar um aplicativo para um aparelho específico, como um sistema embarcado Linux. No caso de um sistema embarcado Linux, o toolchain é usado para gerar a ligação do sistema, o sistema kernel e o sistema de gerenciamento dos documentos raízes.

Geralmente, um toolchain pode ser baseado em um GNU ou em uma máquina virtual de baixo nível e o compilador Clang. Ele pode ser desenvolvido junto com o sistema (nativo) ou a parte (cross).

Um GNU toolchain contém um debugger e compiladores C, C++, Assembly, Java, ... . O GNU contém também uns Binutils usados para converter códigos Assembly em códigos binários, e uma biblioteca C que pode implementar APIs POSIX para comunicar com o kernel.

### Como construir um toolchain?

Para construir um toolchain, é preciso seguir as especificações da máquina como o CPU, a arquitetura do ponto flutuante, o Big/Little Endian, e a interface de aplicação binária. Um toolchain segue uns standards para ser projetado. Ele é sempre usado para um CPU ARM, MIPS ou x86\_64, para o sistema kernel de Linux e um sistema operacional usando bibliotecas C e ABI. Para ter as informações sobre esses dados, podemos usar o comando 'gcc -dumpmachine' na janela de comando da máquina.

### Quais são as bibliotecas em C usadas?

As bibliotecas em C são usadas para fazer chamadas de sistema no sistema kernel de maneira simplificada. As diferentes bibliotecas são:

- glibc: usada para a implementação do API POSIX.
- eglibc: uma biblioteca que foi criada a partir da biblioteca glibc para sistemas embarcados.
- musl libc: usada para sistemas com uma RAM e/ou um armazenamento baixa/o
- uClibc-ng: uma biblioteca que vem da biblioteca Clibc, usada em sistemas embarcados e aparelhos móveis usando  $\mu$ Linux.

### Como escolher uma toolchain?

Podemos uma toolchain que já foi pre-criada, uma pre-criada antes da instalação ou uma gerada usando uma ferramenta de criação de toolchain.

Uma toolchain pre-criada é uma opção mais simples mas é menos flexível dependendo das aplicações. Uma boa toolchain é Debian-based, Yocto Project, Mentor Graphics, TimeSys, MontaVista, Linaro, mas tem mais. Uma boa toolchain tem que vir com uma biblioteca C da sua preferência e deve ser fácil de atualizar.

Já existem projetos de toolchains criadas a partir de pouco que podem ser achadas em sites como <https://trac.clfs.org> ou <http://crosstool-ng.github.io>.

Para criar uma toolchain usando uma ferramenta de criação de toolchain, podemos instalar o Crosstool-NG.

### Instalação do Crosstool-NG:

- 1) Pegar a ferramenta disponível em <https://github.com/crosstool-ng/crosstool-ng.git>
- 2) Ir na pasta \*/testing/docker/\*
- 3) Achar sua pasta de distribuição e entrar nela
- 4) Abrir o documento 'Dockerfile' com o bloco de notas e procurar a lista de dependências

Exemplo:

```
cd 'testing/docker/'
cd ubuntu18.10/

vim Dockerfile
```

Resultado:

```
RUN apt-get install -y gcc g++ gperf bison flex texinfo help2man make libncurses5-dev \
python3-dev autoconf automake libtool libtool-bin gawk wget bzip2 xz-utils unzip \
patch libstdc++6
```

- 5) Instalar todas as dependências da lista de dependências
- 6) Mover para a pasta 'crosstool-ng'
- 7) Usar os comandos seguintes:

```
git checkout crosstool-ng-1.24.0^C
./bootstrap
./configure --enable-local
make
make install
./ct-ng
```

#### Construção do toolchain:

- 1) Primeiro, verificamos que as pastas criadas são limpas e não tem coisas que ficam de criações anteriores.

```
./ct-ng distclean
```

- 2) Agora, pegamos a lista das configurações que são usadas para construir, e escolher uma:

```
./ct-ng list-samples
./ct-ng <configuration-name>
```

- 3) Abrir o menu de configuração:

```
./ct-ng menuconfig
```

- 4) Tirar a opção "Render the toolchain read-only" em "Paths and misc options"
- 5) Começar a construção:

```
./ct-ng build
```

A toolchain esta localizada em "/x-tools/<configuration name>/bin" !