



## Semana 11 – Distribuições Customizadas

Professor: Éder Alves de Moura

Alunos: Wander Victor Verçosa Mares – 11811EAU010

**Uberlândia**

**06/03/2022**

# Distribuições Customizadas no Linux – Projeto Yocto e Buildroot

## 1. Anatomia de software embarcado a partir de uma distribuição em Linux.

Como distribuição em Linux, podemos definir um como uma aplicação embarcada diferente das embarcadas em desktops tradicionais, devido ao tamanho menor, quantidade inferior de módulos, pouco processamento, entre outros, logo, esta aplicação é adequada para ocasiões de sistemas dedicados. O software para esse tipo de distribuição tem a arquitetura básica dividida nas seguintes partes, do nível mais baixo até o topo:

- 1.1. **Hardware:** mesmo este não fazendo parte da arquitetura do software, ainda se torna uma parte importante para o sistema que irá compilar o código, o que irá definir também a especificação do CPU;
- 1.2. **Bootloader:** no processo de inicialização do hardware, a priori é executado o bootloader, este que é responsável por toda a inicialização básica do sistema, isto por sua vez, inclui o kernel Linux;
- 1.3. **Kernel Linux:** é meio central do sistema operacional, este é responsável pelo gerenciamento do hardware, isto significa: CPU, memória, periféricos de entrada e saída, entre outros; além disso, exporta os serviços para aplicação do usuário. Realiza também, a interface entre o hardware e o sistema operacional;
- 1.4. **Rootfs:** o root file system é representado como o sistema de arquivos principal, este possui as bibliotecas do sistema (destinadas ao uso dos serviços exportados pelo kernel e bibliotecas de aplicações do usuário);
- 1.5. **Toolchain:** este não constitui a arquitetura de software do sistema em execução, porém é essencial ser delineada, visto que o conjunto de ferramentas utilizadas em uma máquina “host” para gerar artefatos de software para o sistema em uma máquina target (destino), atendendo as especificidades de cada arquitetura e/ou as necessidades do usuário.

## 2. Projeto Yocto

É um projeto de código aberto, composto por várias ferramentas, estas por sua vez buscam o desenvolvimento de uma distribuição Linux embarcada customizada. A filosofia do projeto é definida como: projeto Yocto não é uma distribuição Linux embarcada, ela cria uma customizada para você”.

Em sua forma completa que é possível diferenciar o mesmo dos demais sistemas de compilação, visto que tem uma comunidade ativa e receptiva, com suporte aos principais fabricantes de semicondutores e é liderado pela The Linux Foundation.

A seguir serão destacados os pontos positivos e negativos do Projeto Yocto:

➤ Positivos:

- Alto esforço gerado pela comunidade;
- É possível compilar por interface, por linha de comando ou interface gráfica;
- Possui vasta gama de pacotes;
- Utilização mais fácil que o buildroot, como exemplo, na construção de rootfs.

➤ Negativos:

- Alto esforço na localização de erros;
- Tempo e esforço elevado para à correta configuração;
- Terminologia confusa a princípio;

### 2.1. Como gerar uma distribuição customizada com o Projeto Yocto.

Em forma de tutorial, para a geração de distribuição em projeto Yocto, este se torna mais complicado em contraste com o buildroot, dessa forma, o aprendizado a longo prazo pode assustar os iniciantes. A seguir o passo-a-passo simplificado:

- 1° Download do código fonte;
- 2° Aplicação de *patches*;
- 3° Configuração e compilação;
- 4° Analise dos resultados e trabalhar na divisão das dependências a partir do conceito de camada;
- 5° Geração de pacotes;
- 6° Realização de teste para garantir a qualidade dos pacotes gerados;
- 7° Geração de *feed* de pacotes;
- 8° Geração de imagem final do rootfs.

### 3. Buildroot

É uma alternativa ao Projeto Yocto, como forma de auxiliar e automatizar a criação de distribuições Linux embarcado. Em comparação, este possui maior quantidade de referências na comunidade. A seguir serão explícitos os pontos positivos e negativos do buildroot:

➤ Positivos:

- Ótima documentação e página da wiki;
- Grande seleção de pacotes;
- Alta escalabilidade, esta permite uma compilação vasta em grandes e pequenos projetos embarcados;
- O rootfs podem ser estabelecidos como gerenciadores de pacotes, no entanto, isso leva tempo;

➤ Negativos:

- Grande número de problemas de compilação e depuração;
- Em teoria é possível se instalar em vários dispositivos, no entanto, só possui um bom funcionamento em dispositivos mais robustos.

#### 3.1. Como gerar uma distribuição customizada com o Buildroot.

A priori, faz-se necessário o buildroot completamente instalado na máquina host que esteja “rodando” alguma distribuição Linux. Os pacotes essenciais a serem instalados são:

- build-essencial;
- neurses5;
- bazar;
- evs;
- git;
- mercurial;

- rsync;
- sep;
- subversion.

Finalizada a instalação dos pacotes essenciais, o buildroot pode ser baixado do site oficial. A seguir, deve-se realizar as configurações necessárias, levando em consideração:

- Especificação da plataforma target para a compilação cruzada;
- Escolha da toolchain;
- Escolha do bootloader;
- Escolha da versão do kernel Linux;
- Customização para o rootfs.

Após seguir o passo-a-passo anterior, posteriormente se faz necessário a geração das imagens. Portanto, define-se o local de download dos pacotes, de início é permitido o reaproveitamento dos arquivos baixados para outro processo de compilação. Outro ponto importante, é a definição do sistema de arquivos. A porta serial de comunicação também deve ser definida. De tal forma, a geração da imagem será sucedida. Dessa forma, as imagens serão geradas, as mesmas devem ser gravadas em um dispositivo de armazenamento a ser conectado ao sistema de destino para execução das imagens.