

# Distribuições Linux customizadas - projeto Yocto e Buildroot

Levy G. S. Galvão

## 1 Anatomia de um software embarcado a partir de uma distribuição Linux

Uma distribuição Linux embarcada difere de uma tradicional de *desktop* em relação ao tamanho reduzido, menor quantidade de módulos, menor processamento, etc., consequentemente sendo adequada para aplicações dedicadas. A arquitetura básica de *software* deste tipo de distribuição pode ser dividida nas seguintes partes, caminhando do nível mais baixo até o topo:

1. *Hardware*: apesar de não constituir a arquitetura de *software*, esta é a peça essencial para qual o sistema deve ser compilado para ser executado na arquitetura específica da CPU;
2. *Bootloader*: primeiro programa a ser executado durante a inicialização do *hardware*, sendo este responsável por toda a inicialização básica do sistema, incluindo carregamento e execução do *kernel* Linux;
3. *Kernel* Linux: núcleo do sistema operacional e é responsável pelo gerenciamento *hardware*, como: CPU, memória, dispositivos de entrada e saída (I/O); e exporta os serviços para as aplicações do usuário, comumente conhecido por realizar a interface entre o sistema operacional e o *hardware*;
4. *Rootfs*: o *root file system* é o sistema de arquivos principal que possui as bibliotecas do sistema para uso dos serviços exportados pelo *kernel* e bibliotecas de aplicações do usuário;
5. *Toolchain*: apesar de não constituir a arquitetura de *software* do sistema em execução, esta é essencial ser delineada, pois é um conjunto de ferramentas utilizadas em uma máquina hospedeira (*host*) para gerar os artefatos de *software* do sistema para uma máquina alvo (*target*), atendendo as especificadas de cada arquitetura alvo e as necessidades do usuário;

## 2 Projeto Yocto

O projeto Yocto é um projeto de código aberto composto por várias ferramentas que buscam a criação de uma distribuição Linux embarcado customizada. O projeto tem como filosofia que: “O projeto Yocto não é uma distribuição Linux

embarcada, ele cria uma customizada para você”.

Este se diferencia dos demais sistemas de compilação devido sua completude, com comunidade ativa e receptiva, suporta os principais fabricantes de semicondutores e é liderado pela *The Linux Foundation*.

Considerando os prós e contras do projeto Yocto, pode-se considerar:

- Prós:
  - Muito esforço foi colocado no projeto pela comunidade;
  - A compilação do sistema pode ser feita por interface por linha de comando ou interface gráfica;
  - Possui vários pacotes;
  - Facilidade de uso, incluindo na construção do *rootfs*, sendo mais fácil que o *buildroot*;
- Contras:
  - Exige maior esforço para encontrar erros;
  - Leva muito tempo e esforço para que seja corretamente configurado corretamente;
  - Sua terminologia pode ser confusa a princípio;

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

O passo-a-passo para a geração de uma distribuição utilizando o projeto Yocto é bem mais complicado que para o *buildroot*, portanto possuindo uma curva de aprendizado mais íngreme e que pode assustar principiantes. Porém seu fluxo de trabalho pode ser exemplificado por:

1. Baixar o código fonte;
2. Aplicar *patches*;
3. Configuração e compilação;
4. Analisar o resultado e trabalhar na divisão das dependências a partir do conceito de camada;
5. Gerar pacotes;
6. Realização de testes para garantir a qualidade dos pacotes gerados;
7. Gerar *feed* de pacotes;
8. Gerar a imagem final do *rootfs*

## 3 *Buildroot*

O *buildroot* é uma alternativa ao uso do projeto Yocto (e muitos outros) para auxiliar e automatizar a criação de distribuições Linux embarcado. Este possui bastante referências na comunidade com vários exemplos.

Considerando os prós e contras do *buildroot*, pode-se considerar:

- Prós:

- Ótima documentação e página da *wiki*;
- Grande seletude de pacotes;
- Ótima escalabilidade, permitindo compilação de grandes ou pequenos projetos embarcados;
- O *rootfs* pode ser criado com gerenciadores de pacote, porém toma tempo;
- Contras:
  - Muitos problemas de compilação e depuração;
  - Apesar de poder ser instalado em vários dispositivos, só funciona bem com algumas distribuições de ponta;

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

Inicialmente o *buildroot* deve ser instalado na máquina hospedeira rodando alguma distribuição Linux. Os pacotes essenciais de instalação são:

- `build-essencial`;
- `ncurses5`;
- `bazaar`;
- `cvs`;
- `git`;
- `mercurial`;
- `rsync`;
- `scp`;
- `subversion`.

Após a instalação dos pacotes essenciais, o *buildroot* pode ser baixado diretamente do seu site oficial.

O passo seguinte é realizar as configurações necessárias, prezando-se por:

- Especificar uma plataforma alvo para a compilação (ou compilação cruzada);
- Escolha da *toolchain*;
- Escolha do *bootloader*;
- Escolha da versão do *kernel* Linux;
- Customizações para o *rootfs*;

Após essas configurações, a etapa posterior é a geração das imagens. Inicialmente o local de *download* dos pacotes é definido, permitindo reaproveitamento dos arquivos baixados para outro processo de compilação. O tipo do sistema de arquivos também deve ser definido, e.g. FAT32, EXT4. A porta serial de comunicação também deve ser definida. Por fim a geração da imagem pode ser feita.

Após a geração das imagens, estas devem ser gravadas em um dispositivo de armazenamento a ser conectado ao sistema de destino para executar as imagens.