

Este resumo aborda os projetos YOCTO E BUILDROOT que são duas das ferramentas mais adotadas desenvolvidas para facilitar o trabalho de quem precisa gerar uma distribuição Linux customizada. Aqui encontram-se as características de ambos os projetos contendo: sua função, elementos necessários para sua configuração, principais diferenças e recomendação de aplicação.

## YOCTO

O projeto Yocto é definido como “um projeto de colaboração de código aberto que fornece modelos, ferramentas e métodos para ajudá-lo a criar sistemas personalizados baseados em Linux para produtos embarcados, independentemente da arquitetura de hardware.”

Yocto usa Openembedded como seu sistema de construção. A saída de uma construção de projeto Yocto consiste amplamente em três componentes:

- Target run-time binaries: Incluem o carregador de inicialização, kernel, módulos do kernel, imagem do sistema de arquivos raiz e quaisquer outros arquivos necessários para implementação do Linux.
- Package feed: É a coleção de pacotes de software disponíveis para instalação.
- Target SDK: É a coleção de bibliotecas e arquivos de cabeçalho que representam o software instalado no destino.

O projeto Yocto é amplamente utilizado na indústria e tem o apoio de muitas empresas influentes além de uma grande comunidade de desenvolvedores, esta combinação ajuda no impulsionamento do projeto.

Yocto é facilmente expandido por meio de camadas, que podem ser publicadas independentemente para adicionar funcionalidade adicional, para plataformas de destino não disponíveis nas versões do projeto ou para armazenar personalizações exclusivas para seu sistema. Camadas podem ser adicionadas à sua configuração para adicionar recursos exclusivos que não estão especificamente incluídos nas versões de estoque.

Yocto tem o suporte de dispositivo mais amplo devido a diversos fabricantes de semicondutores e placas, é provável que o Yocto ofereça suporte a qualquer plataforma de destino que seja escolhida. É ainda extremamente flexível e personalizável, estas personalizações podem ser armazenadas em camadas.

Entretanto, a maior desvantagem do projeto Yocto é a curva de aprendizado, exigindo muito tempo e esforço para aprender e realmente entender o sistema. O número de pacotes que precisam ser construídos, incluindo o conjunto de ferramentas, kernel e todos os componentes de tempo de execução de destino é significativo tornando o tempo e recursos de build de desenvolvimento bastante altos. Além disso, as estações de trabalho de desenvolvimento para Yocto tendem a ser sistemas grandes, tornando inviável a utilização de notebooks compactos.

As ferramentas de utilização do projeto Yocto requerem um host de construção com uma CPU de arquitetura x86. O sistema de construção do projeto Yocto usa processamento paralelo sempre que possível. Portanto, um host de construção com várias CPUs ou uma CPU com vários núcleos reduz significativamente o tempo de construção.

Existem várias maneiras de obter as ferramentas do projeto Yocto, ou mais precisamente, a distribuição de referência do projeto Yocto, Poky. Acesse <https://www.yoctoproject.org/downloads> e clique na versão mais recente do Poky. Baixar a versão coloca um arquivo compactado da distribuição de referência Poky denominado poky-<codename> - <release>.tar.bz2 em seu sistema.

A configuração do host de construção requer a instalação de pacotes de software adicionais. Todas as quatro distribuições Linux principais têm esses pacotes disponíveis em seus repositórios de pacotes. Depois de instalar os pacotes adicionais, você precisa descompactar o arquivo Poky, que inclui todos os dados de configuração necessários, receitas, scripts de conveniência e BitBake. O BitBake requer Python com uma versão principal 2.6 ou 2.7. BitBake atualmente não suporta o novo Python 3, que introduziu mudanças na sintaxe da linguagem e novas bibliotecas quebrando a compatibilidade com versões anteriores.

O comando a ser usado e os pacotes adicionais a instalar dependem da distribuição do Linux que você instalou no host de construção. Para Ubuntu:

```
user@ubuntu:~$ sudo apt-get install gawk wget git-core diffstat \
unzip texinfo gcc-multilib build-essential chrpath socat \
libSDL1.2-dev xterm
```

A instalação do Poky requer apenas a descompactação do arquivo baixado do site do projeto Yocto anteriormente. Recomenda-se a criação de um subdiretório no diretório inicial para todas as coisas relacionadas às compilações do seu projeto Yocto.

## **BUILDROOT**

O projeto Buildroot é definido como “uma ferramenta simples, eficiente e fácil de usar para gerar sistemas Linux embarcados por meio de compilação cruzada”. Este projeto é focado na simplicidade e no minimalismo. Em geral, Buildroot desabilitará todas as configurações opcionais de tempo de compilação para todos os pacotes (com algumas exceções notáveis), resultando no menor sistema possível. Cabe ao designer do sistema habilitar as configurações apropriadas para um determinado dispositivo.

Buildroot cria todos os componentes da fonte, mas não oferece suporte ao gerenciamento de pacotes no destino. Os aplicativos podem atualizar o sistema de arquivos de destino, mas não há mecanismo para instalar novos pacotes em um sistema em execução.

A saída de uma construção de projeto Buildroot consiste amplamente em três componentes:

- A imagem do sistema de arquivos raiz e quaisquer outros arquivos auxiliares necessários para implantar o Linux na plataforma de destino.
- O kernel, boot-loader e módulos de kernel apropriados para o hardware alvo.
- O conjunto de ferramentas usado para construir todos os binários de destino.

Por ser focado em simplicidade, em geral, Buildroot é mais fácil de aprender do que Yocto. O sistema de construção principal é curto o suficiente para permitir que um desenvolvedor entenda todo o sistema enquanto é expansível o suficiente para atender às necessidades dos desenvolvedores Linux embarcados.

Devido ao objetivo do design de desabilitar todas as configurações opcionais de tempo de construção, Buildroot geralmente produzirá as menores imagens possíveis usando a configuração pronta para uso. Os tempos de construção e os recursos do host de construção também serão menores, em geral, do que os do projeto Yocto.

O foco na simplicidade e nas opções de construção habilitadas mínimas implicam que você pode precisar fazer uma customização significativa para configurar uma construção Buildroot para seu aplicativo. Além disso, todas as opções de configuração são armazenadas em um único arquivo, o que significa que se você tiver várias plataformas de hardware, precisará fazer cada uma das alterações de personalização para cada plataforma.

Qualquer mudança no arquivo de configuração do sistema requer uma reconstrução completa de todos os pacotes o que pode resultar em compilações longas. Embora a primeira compilação possa ser mais curta que uma compilação Yocto, as compilações subsequentes podem exigir a reconstrução de muitos componentes.

A instalação do Buildroot é feita através do download na página do projeto <https://buildroot.org/download.html> em seguida descompacte-o em um diretório qualquer na máquina. É necessário também o kernel para a mini2440.

## **CONCLUSÃO**

O projeto Yocto tem evoluído bastante nos últimos releases, e isso inclui melhorias nos tempos de compilação e no tamanho das imagens geradas. Sendo bem mais flexível e completo que o Buildroot, Yocto dá um controle maior ao desenvolvedor de distribuições Linux. O amplo suporte da indústria, a comunidade ativa e o amplo suporte de plataforma tornam esta opção uma boa escolha para designers.

O projeto Buildroot é bem mais simples e intuitivo que Yocto, entretanto, se o seu projeto requer vários tipos de hardware ou outras diferenças, a utilização de Buildroot pode ser reconsiderada devido à complexidade de sincronizar várias configurações, já para um sistema que consiste em uma única configuração, Buildroot é uma boa escolha.