



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



## **Tarefa Semanal 11 – Sistemas Embarcados II**

### **Projeto Yocto e Buildroot**

Fernando Rabelo Fernandes Junior - 11611EMT020

Uberlândia, março de 2022

## PROJETO YOCTO

O projeto Yocto é um projeto de colaboração de software livre que fornece modelos, ferramentas e métodos que suportam sistemas customizados baseados em Linux para produtos integrados, independentemente da arquitetura de hardware. Para os que estão se perguntando o que o nome significa, o termo *yocto* designa a menor unidade do SI. Como prefixo, *yocto* indica 10 elevado a -24.

Dois componentes importantes do projeto Yocto são mantidos em conjunto com o projeto OpenEmbedded: BitBake, o mecanismo de criação, e OpenEmbedded-Core, o conjunto principal de receitas usadas para executar o processo de criação.

Como um projeto de colaboração, também conhecido como projeto principal, o Yocto incorpora diversas partes heterogêneas do processo de desenvolvimento. Essas partes são conhecidas como *projetos* dentro do projeto Yocto geral e incluem ferramentas de criação, metadados de instruções de criação chamados *receitas*, bibliotecas, utilitários e interfaces gráficas com o usuário (GUIs).

Poky é um sistema de desenvolvimento de referência para o projeto Yocto. Inclui o BitBake, o OpenEmbedded-Core, um pacote de suporte à placa (BSP) e outros pacotes ou camadas incorporadas à criação. O nome Poky também designa a distribuição padrão do Linux resultante do uso do sistema de criação de referência, que pode ser mínima (*core-image-minimal*) ou um sistema Linux completo com uma GUI (*core-image-sato*).

É possível considerar o sistema de criação Poky como um sistema de referência para todo o projeto — um exemplo funcional do processo em ação. Ao fazer o download do projeto Yocto, na verdade você obtém uma instância dessas ferramentas, utilitários, bibliotecas, cadeia de ferramentas e metadados que podem ser usados para criar o sistema padrão, como se descreve aqui. O sistema de referência e a distribuição de referência que ele cria se chamam Poky. Também é possível usar isso como um ponto de partida para criar a sua própria distribuição — que pode ter o nome que você quiser, evidentemente.

A cadeia de ferramentas — um compilador, assembler, vinculador e outros utilitários binários necessários para criar arquivos executáveis binários para uma determinada arquitetura — é um item necessário para todos os sistemas de criação. O Poky usa a GNU Compiler Collection (GCC), mas é possível especificar outras cadeias de ferramentas. O Poky usa uma técnica conhecida como *compilação cruzada*: o uso de uma cadeia de ferramentas em uma arquitetura para desenvolver arquivos executáveis binários para outra arquitetura (por exemplo, desenvolver uma distribuição ARM em um sistema baseado em x86). Os desenvolvedores frequentemente usam a compilação cruzada no desenvolvimento de sistemas embarcados para aproveitar o desempenho mais alto do sistema host.

O BitBake é um mecanismo de criação. Ele lê as receitas e as segue, buscando pacotes, construindo-os e incorporando os resultados em imagens inicializáveis. O BitBake é mantido conjuntamente pelos projetos Yocto e OpenEmbedded.

Na tentativa de facilitar o desenvolvimento em Linux integrado, o projeto Yocto fornece alguns métodos diferentes para trabalhar de forma gráfica. Um acréscimo relativamente novo ao projeto é o Hob, que fornece um frontend gráfico para o BitBake e o processo de criação. Os dois estão continuamente em desenvolvimento, completos com estudos dos usuários da comunidade.

A Embedded GLIBC (EGLIBC) é uma variante da GNU C Library (GLIBC), projetada para funcionar bem em sistemas embarcados. Os objetivos da EGLIBC envolvem área de cobertura reduzida, componentes configuráveis e suporte melhor para configuração cruzada e testes cruzados. A EGLIBC faz parte do projeto Yocto, mas é mantida dentro da sua própria estrutura de controle.

Estas são outras ferramentas sob a bandeira do projeto Yocto:

- Autobuilder: cria construções automatizadas contínuas das ferramentas do projeto Yocto, permitindo atividades automatizadas de garantia de qualidade (QA).
- Cross-Prelink: fornece pré-vinculação para ambientes de desenvolvimento de compilação cruzada, melhorando o desempenho.
- Pseudo: emula o acesso do root, uma parte essencial da criação de uma imagem final inicializável.
- Swabber: detecta quando uma criação de compilação cruzada foi contaminada por componentes do host.
- Build Appliance: é uma máquina virtual que executa o Hob, permitindo as pessoas que usam hosts de criação que não são de Linux vejam o processo do projeto Yocto em primeira mão. (Observação: no momento, as ferramentas de desenvolvimento do projeto Yocto só são suportadas no Linux.)

## PROJETO BUILDROOT

Buildroot é uma ferramenta que auxilia e automatiza a criação de distribuições Linux Embarcado. Ele faz a cross-compilação do código para a arquitetura da placa para a qual se quer construir a distribuição (mais sobre Cross-compilação neste artigo).

A documentação do Buildroot é bem completa (embora um pouco difícil de entender à primeira vista) e existe bastante referência online e na comunidade, como o site do Sérgio Prado e da Bootlin (antiga Free Electrons).

Instalar o Buildroot é, basicamente, clonar ou baixar o repositório da ferramenta. Porém, para que possa ser usada, alguns pacotes são necessários, como explicado neste outro artigo.

Você pode desenvolver qualquer aplicação que desejar. Por simplicidade, recomendo o uso de Python caso você não tenha familiaridade com makefiles e cross-compilação de programas em C.

Para o Buildroot, importa que o código esteja em algum lugar acessível para que a ferramenta possa fazer o download e compilação (se necessário). Por exemplo, criando um repositório no seu Github.