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

FEELT49081 – SISTEMAS DIGITAIS PARA MECATRÔNICA

Prática de Sistemas Digitais para Mecatrônica

Prof. Éder Alves de Moura

SEMANA 11:

DISTRIBUIÇÕES CUSTOMIZADAS

MATHEUS ALVES DE PAULA 11521EMT008

Uberlândia

Março de 2022

1. ROTEIRO DE ATIVIDADES

A atividade desta semana consiste em criar uma pasta chamada ‘Semana11’ no repositório do Github e incluir o presente relatório com o desenvolvimento das atividades propostas no roteiro.

1.1. ATIVIDADE 1

A anatomia de um software embarcado a partir de uma distribuição Linux é uma série de blocos básicos, conforme visualizado na Figura (1).

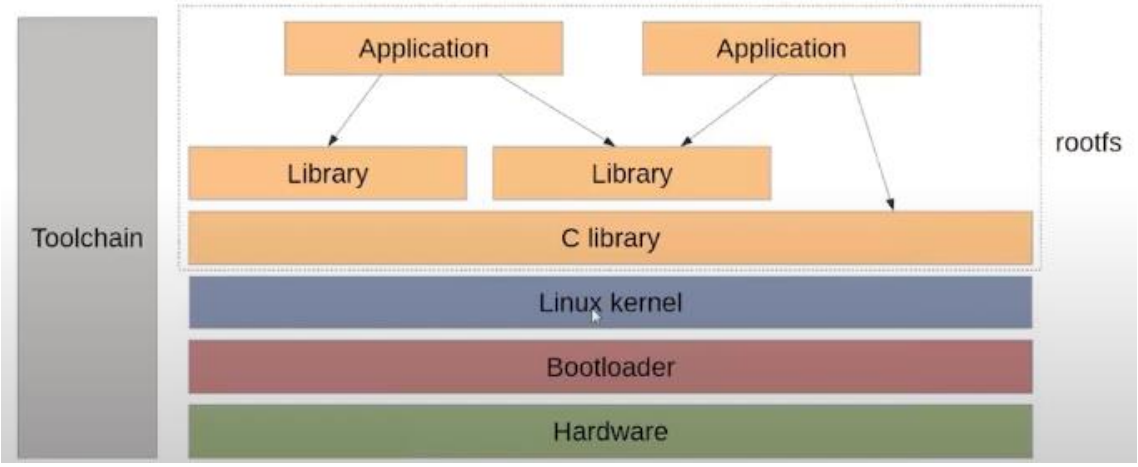

Figura 1 - Anatomia de um software embarcado

O Hardware, já conhecemos da arquitetura de computadores, é o seu produto;

O Bootloader é iniciado pelo hardware, responsável pela inicialização básica, carregamento e execução do kernel Linux;

O Kernel Linux é o núcleo do sistema operacional. Gerencia CPU, memória e I/O, exportando serviços para as aplicações do usuário;

O Rootfs é o sistema de arquivos principal. Possui as bibliotecas do sistema para uso dos serviços exportados pelo kernel, além das bibliotecas e aplicações do usuário;

A Toolchain, como visto anteriormente, é o conjunto de ferramentas para gerar os artefatos de software do sistema.

O Yocto Project é um sistema de build composto por diversas ferramentas para criação de distribuições Linux embarcado customizadas e tem suporte dos principais

fabricantes de semicondutores. A Figura (2) apresenta uma comparação do Yocto com outros projetos.

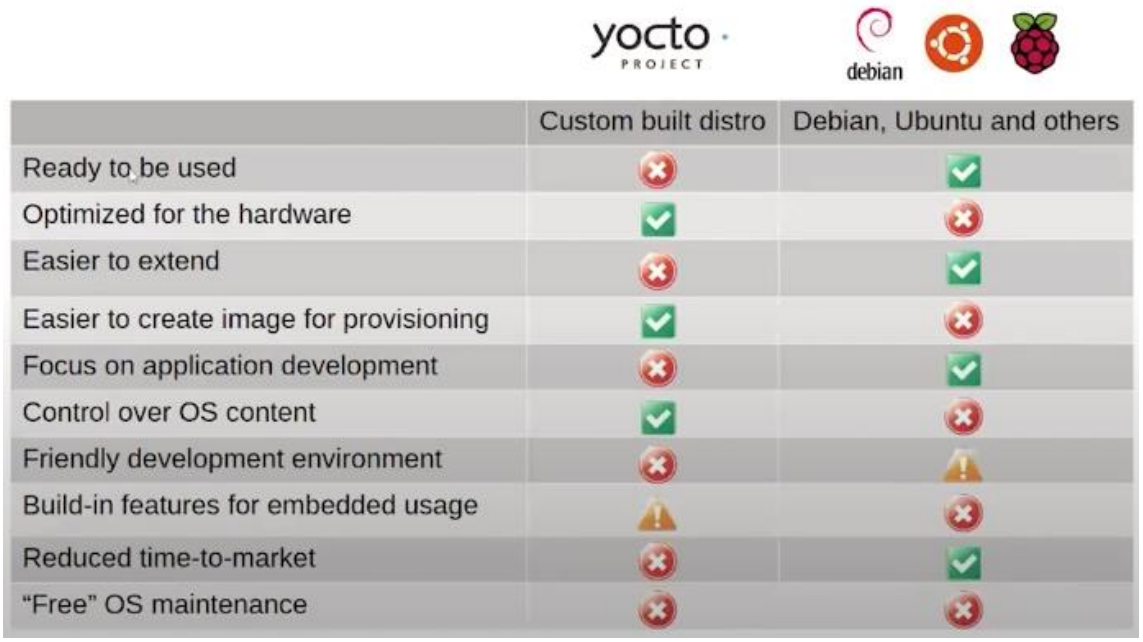

Figura 2 - Comparação

Assim, podemos definir algumas vantagens e desvantagens, conforme apresentado a seguir.

- Prós:
  1. Total controle do que existe "debaixo do capô" e das licenças;
  2. Alta Flexibilidade;
  3. Fácil Escalabilidade;
  4. Facilidade no controle de mudanças e updates;
  5. Otimizado de acordo com os requisitos de projeto e hardware;
  6. Ótimo para etapas de Bring-up, Testes e Validação.

- Contras:
  1. Ramp up pode ser demorado;
  2. Nem todos os pacotes necessários podem estar disponíveis para cross compilação;
  3. Suporte e mão de obra especializada;
  4. Necessidade de infra-estrutura para geração das imagens.

As etapas de criação de uma aplicação customizada com o projeto Yocto são:

1. Configurar o ambiente hospedeiro para desenvolvimento do Yocto;
2. Criar uma copia local dos arquivos no sistema hospedeiro;
3. Estabelecer um repositorio de metadados no sistema;
4. Criar uma BSP layer usando o script yocto-bsp;
5. Efetuar mudanças de configurações para a nova BSP layer;
6. Modificar as recipes para a nova BSP layer;
7. Preparar o build;
8. Construir imagem.

1.2. ATIVIDADE 2

O Buildroot é um conjunto de makefiles e patches, sendo possível gerar: toolchain, rootfs, kernel e bootloader. É independente de fornecedores e suporta várias arquiteturas, como por exemplo, ARM, PowerPC, x86, etc.

Os comandos básicos para o download do buildroot é apresentado na Figura (3).

```
$ mkdir ~/bb_lab
$ cd ~/bb_lab
$ wget https://buildroot.org/downloads/buildroot-2021.02.6.tar.gz
$ tar xvf buildroot-2021.02.6.tar.gz
$ cd buildroot-2021.02.6/
$ make menuconfig
```

Figura 3 – Comandos básicos

Após executar os comandos, será aberta a tela de configuração, conforme apresentado na Figura (4).

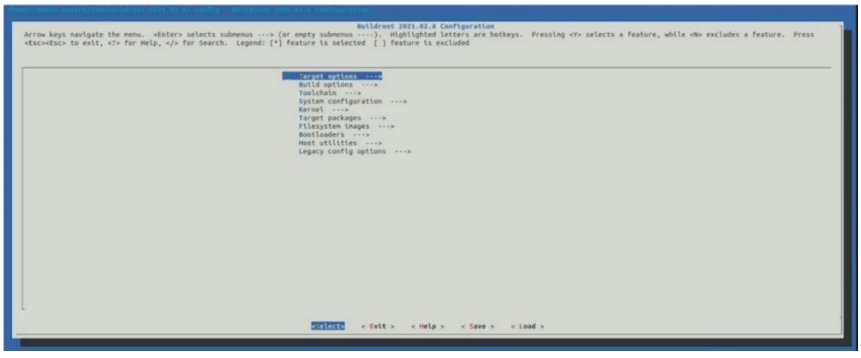

Figura 4 - Configuração

O Buildroot pode construir automaticamente a cadeia de ferramentas de compilação cruzada necessária, criar um sistema de arquivos raiz, compilar uma imagem do kernel Linux e gerar um carregador de inicialização para o sistema embarcado de destino ou pode executar qualquer combinação independente dessas etapas.