

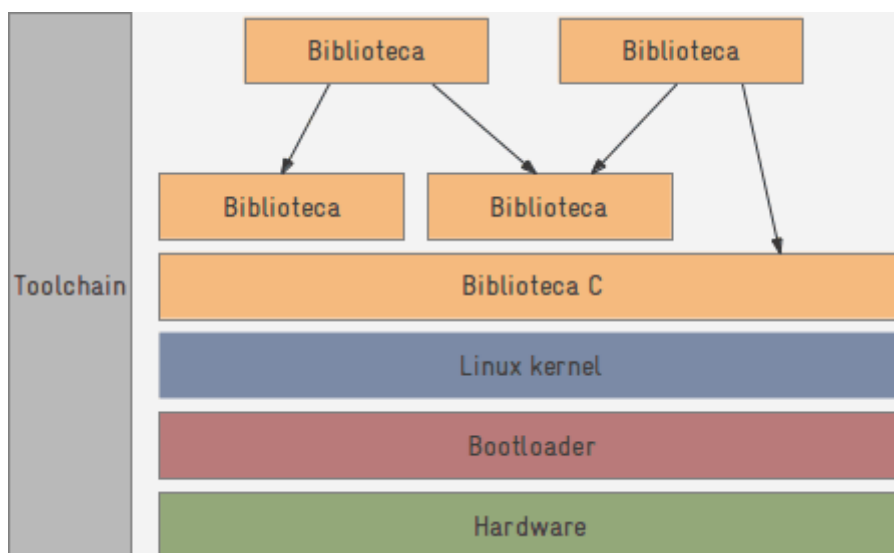
Atividades da semana 11

Baltazar Alic Borges da Silva – 11711EMT022

1)a)

Um sistema Linux embarcado não se difere, no quesito sistêmico, de um sistema Linux desktop. A mesma estrutura e conceitos são aplicados em ambos os domínios. A principal diferença está nos requisitos de processamento, armazenamento, consumo de energia e confiabilidade. Na maioria dos sistemas Linux embarcado, os recursos disponíveis são limitados e muitas vezes a interface com usuário é bastante limitada ou simplesmente não existe.

Na arquitetura básica de um sistema Linux estão cinco componentes básicos:



O Hardware: é o produto; Bootloader: iniciado pelo hardware, responsável pela inicialização básica, carregamento e execução do kernel Linux; Kernel Linux: núcleo do sistema operacional. Gerencia CPU, memória e I/O, exportando serviços para as aplicações do usuário; Rootfs: sistema de arquivos principal. Possui as bibliotecas do sistema para uso dos serviços exportados pelo kernel, além das bibliotecas e aplicações do usuário; Toolchain: conjunto de ferramentas para gerar os artefatos de software do sistema.

b)

O projeto Yocto é um sistema de Build o qual é capaz de Fornecer Toolchain para cross-compiling e debug; Construir dos artefatos de software; Gerenciar cadeias de dependências entre os pacotes e construir imagens.

O Yocto Project, além de ser um Sistema de Build, ele endereça questões críticas no desenvolvimento de produtos como: redução do time-to-market, redução de custos de desenvolvimento e manutenção.

c)

Primeiro, instale todos os pacotes de desenvolvimento necessários a partir do repositório de software do seu sistema host. Se você já desenvolve software no seu host, é possível que a maioria deles já esteja instalada. As Listagens Instalando pré-requisitos no Ubuntu, Instalando pré-requisitos no Fedora, Instalando pré-requisitos no CentOS e Instalando pré-requisitos no openSUSE mostram como instalar os pacotes necessários em várias distribuições.

Examine o arquivo de configuração principal (conf/local.conf). Por padrão, o arquivo de configuração é configurado para criar uma imagem para qemu86 ou uma instância de QEMU emulando um processador x86 de 32 bits. Se você tem um host de multiprocessador, é muito recomendável remover os comentários das seguintes opções de paralelismo para acelerar a sua criação. Por enquanto, configure esses dois valores como 2x o número de núcleos do seu processador (por exemplo, 8 para um processador de 4 núcleos).

```
BB_NUMBER_THREADS = "8"
```

```
PARALLEL_MAKE = "-j 8"
```

Execute um desenvolvimento inicial

O desenvolvedor inicial usa o compilador do host para desenvolver a cadeia de ferramentas de várias compilações e quaisquer outras necessárias.

Inicialize a nova imagem

Quando a criação for concluída, será possível inicializá-la por meio do QEMU emulando uma máquina x86 de 32 bits. Para fazer isso, basta executar:

```
runqemu qemu86
```

É possível efetuar login na máquina com o nome de usuário root, sem senha.

É possível fazer muitas customizações na imagem final, nos pacotes que a formam e no processo em si. O sistema de desenvolvimento é escrito em Python e está bem documentado.

Test drive do Hob

O Hob é uma GUI que ajuda a automatizar alguns dos processos descritos anteriormente.

Para executar o Hob, basta digitar hob a partir do seu diretório de desenvolvimento para a tela principal aparecer. Comece escolhendo uma arquitetura (nesse caso, qemu86). Em seguida, o Hob analisa as receitas disponíveis, como mostra a Configuração da imagem do Hob.

Configuração da imagem do Hob

Depois, é possível escolher uma imagem de base (o exemplo anterior construiu `core-image-minimal` — portanto, esses binários já existem. Escolha `core-image-minimal` e, a seguir, clique em `View recipes` ou `View packages`. A tela na Pacotes do Hob mostra as listas de rolagem em `View packages`. É possível selecionar ou desmarcar essas caixas ou removê-las da criação, respectivamente. Clique em um cabeçalho de coluna para classificar por essa coluna.

Pacotes do Hob

Printscreen do Hob na área de pacotes com todos selecionados.

Na guia `Packages`, observe que o Python não está incluído na criação mínima. É possível incluí-lo simplesmente marcando a caixa de seleção referente a `python-2.7.2-r2.14`, como mostra a Incluindo um pacote no Hob. Todas as dependências e subpacotes também são incluídos imediatamente.

Incluindo um pacote no Hob

Printscreen do Hob na área de pacotes demonstrando a inclusão de um pacote.

Clique em `Build image` para ver o Hob desenvolver uma nova imagem com as suas opções. Em seguida, é possível executar a imagem resultante em um emulador do QEMU simplesmente clicando em `Run Image`, como mostra a Detalhes da imagem do Hob.

Verificando o acréscimo

Printscreen do QEMU desmenbrando a verificação de acréscimo.

Há mais informações, tutoriais e vídeos disponíveis no website do projeto Yocto.

2)b)

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

c)

Preparar o ambiente Linux

Abrir um terminal

Usar o menu convencional ou shortcut: `ctrl + alt + t`

Atualizar sistema - rode como root

```
sudo apt-get install build-essential
```

o Build essential é um pacote que contém as referências para vários outros pacotes necessários para construir softwares em geral:

2. Criar um diretório de trabalho:

3. Ex: `mkdir ~/buildroot`

4. Ir para o diretório de trabalho:

Ex: `cd ~/buildroot`

5. Baixar o BuildRoot do fornecedor:

Ex: `wget http://buildroot.uclibc.org/downloads/buildroot-2015.02.tar.gz`

Para agilizar este processo, o mesmo arquivo está disponível em

http://www.inf.pucrs.br/emoreno/undergraduate/CC/progperif/common/buildroot/br_15.02

/buildroot-2015.02.tar.gz. Para baixa-lo, basta utilizar o comando `wget` no terminal com este

endereço.

6. Extrair:

Ex: `tar -xvzf buildroot-2015.02.tar.gz`

7. Ir para o diretório:

Ex: `cd buildroot-2015.02`

8. Preparar para rodar o build

Ex: `make raspberrypi_defconfig`

9. Executar o BuildRoot

Ex: `make menuconfig`

Menuconfig é um recurso gráfico para a elaboração de tela em um terminal. Nem sempre

tal recurso está disponível, e para instalá-lo deve-se utilizar o seguinte comando

`sudo apt-get install libncurses5-dev`