

Programação de Periféricos

Componentes

Edson Moreno

edson.moreno@pucrs.br

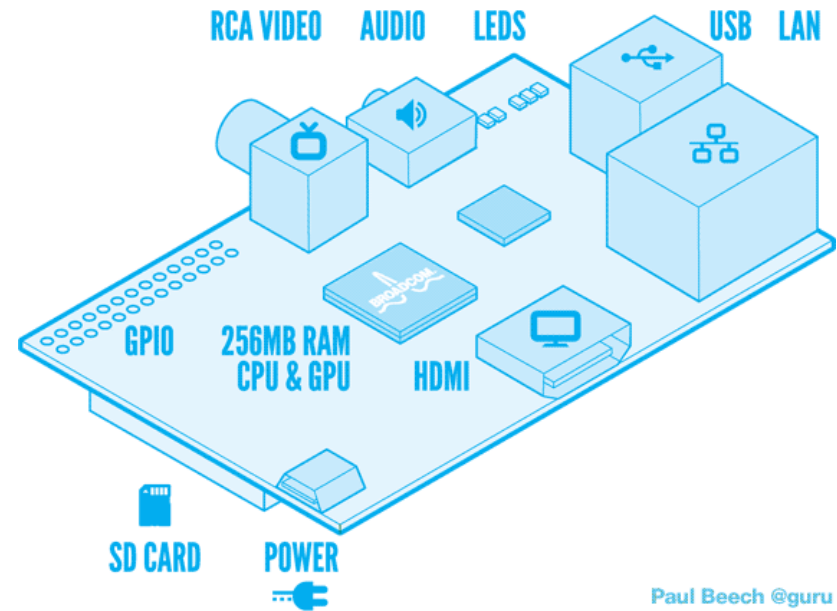
<http://www.inf.pucrs.br/~emoreno>

Apresentação

- ***Raspberry pi***
- Sistema Operacional embarcado
- Atividade

Raspberry pi

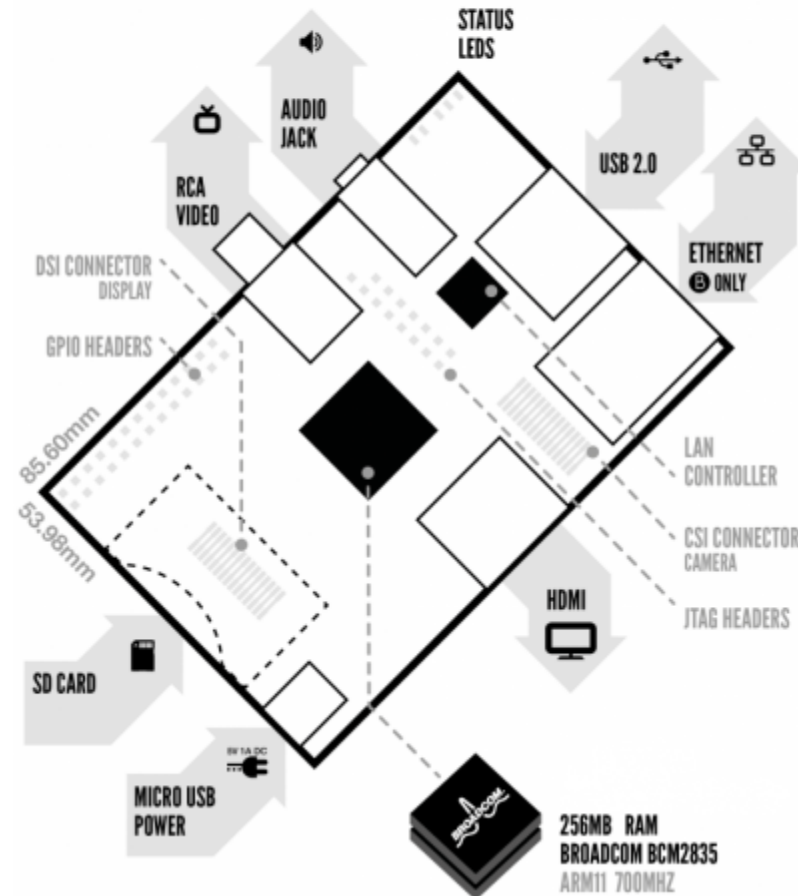
- Sistema computacional
 - Processador + memória + IO
 - Requer SD card para o boot
- Aplicações distintas
 - Computador do tamanho de um cartão de crédito
 - Empregável em sistemas eletrônicos específicos
 - etc





Raspberry pi

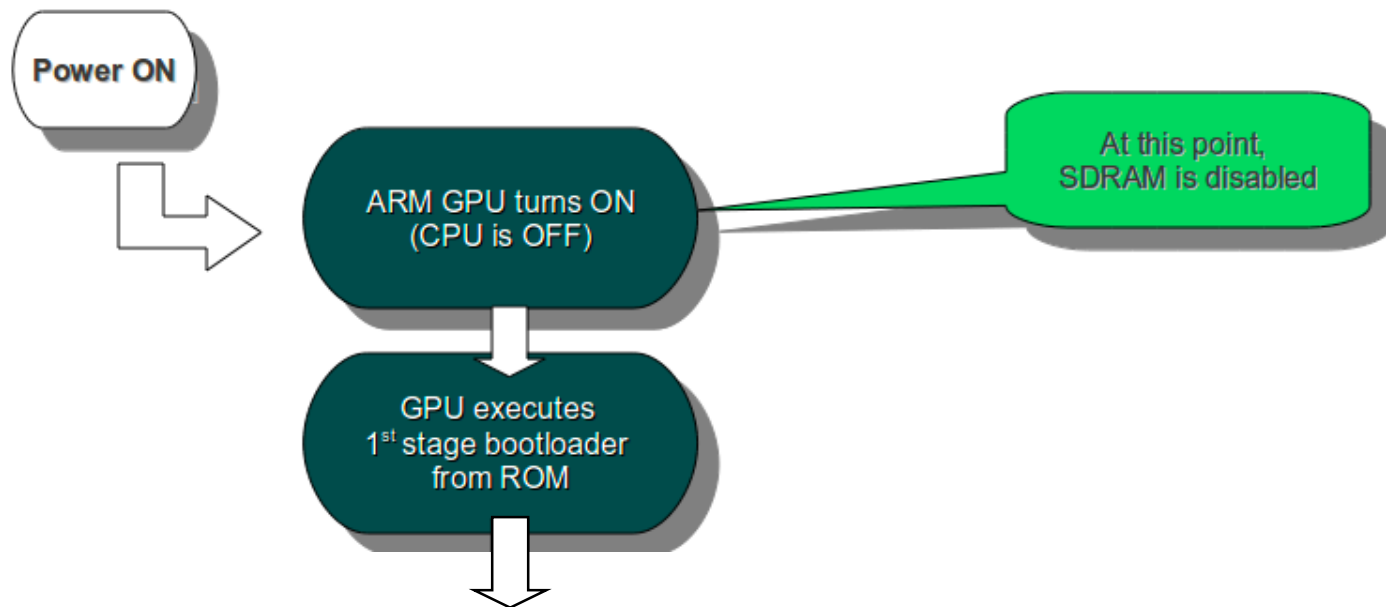
- Componentes disponíveis
 - O SoC é um Broadcom BCM2835
 - Contém um ARM1176JZFS
 - Com FPU
 - Opera a 700MHz
 - Um Videocore 4 GPU.
 - Capaz de reproduzir BluRay, usando H.264 a 40Mbps.
- Mas o que é um SoC?
 - Proposta de construção de sistemas completos em um único *die*.



Como funciona o Raspberry pi

- Arquivos para o boot devem estar numa partição FAT32
- O Raspberry Pi requer um SD card instalado para bootar
- Três estágios para o boot
 - Estágio 1
 - Estágio 2
 - Estágio 3

Primeiro estágio de boot



Como funciona o Raspberry pi

- Três estágios para o boot
 - Estágio 1
 - Quando ligado, executa o first-stage boot
 - Código presente na ROM interna
 - Monta uma partição de boot em FAT 32 no SDCard
 - Preparação para o segundo estágio de boot
 - Estágio 2
 - Estágio 3

Como funciona o Raspberry pi

- Três estágios para o boot
 - Estágio 1
 - Estágio 2
 - A CPU ainda não está ativa, nem a RAM
 - Estágio ainda executado pela GPU
 - Ação realizada durante este estágio
 - Bootloader.bin carregado na cache L2 da GPU
 - Arquivo estava armazenado na SD Card
 - Resultado deste estágio
 - Habilita a RAM
 - Carrega o arquivo start.elf da SD Card
 - Estágio 3

Como funciona o Raspberry pi

- Três estágios para o boot
 - Estágio 1
 - Estágio 2
 - Estágio 3
 - Execução do start.elf caracteriza o bootloader do terceiro estágio
 - GPU continua ativa e CPU inativa
 - start.elf é o firmware da GPU
 - Carrega as configurações básicas do sistema (config.txt)
 - Ao final deste estágio
 - O kernel.img é carregado (Kernel.img = binário do SO)
 - Libera a execução da CPU
 - Inicia o boot do kernel

E o kit?

- element14.com
 - Raspberry pi
 - Interfaces:
 - 26 pinos de GPIO dedicados
 - Uma interface UART, uma interface i2c, uma interface SPI com dois modos, audio i2s, e alimentação (3v3, 5v, gnd)
 - USBs, Ethernet, audio, HDMI,
 - Placa de expansão
 - SD Card
 - Cabos
 - USB (alimentação)

Apresentação

- Raspberry pi
- ***Sistema Operacional embarcado***
- Atividade

Linux embarcado

- Hardware requerido
 - Dá suporte a uma grande variedade de arquitetura de processadores
 - x86 e x86-64, ARM, PowerPC , etc
 - De 32/64 bits
 - Com e sem MMU
 - Não suportado para pequenos microcontroladores
 - Com exceção do toolchain, bootloader e kernel, todos outros componentes são normalmente independentes de arquitetura
 - Estes são recursos que serão abordado ao longo da disciplina

Linux embarcado

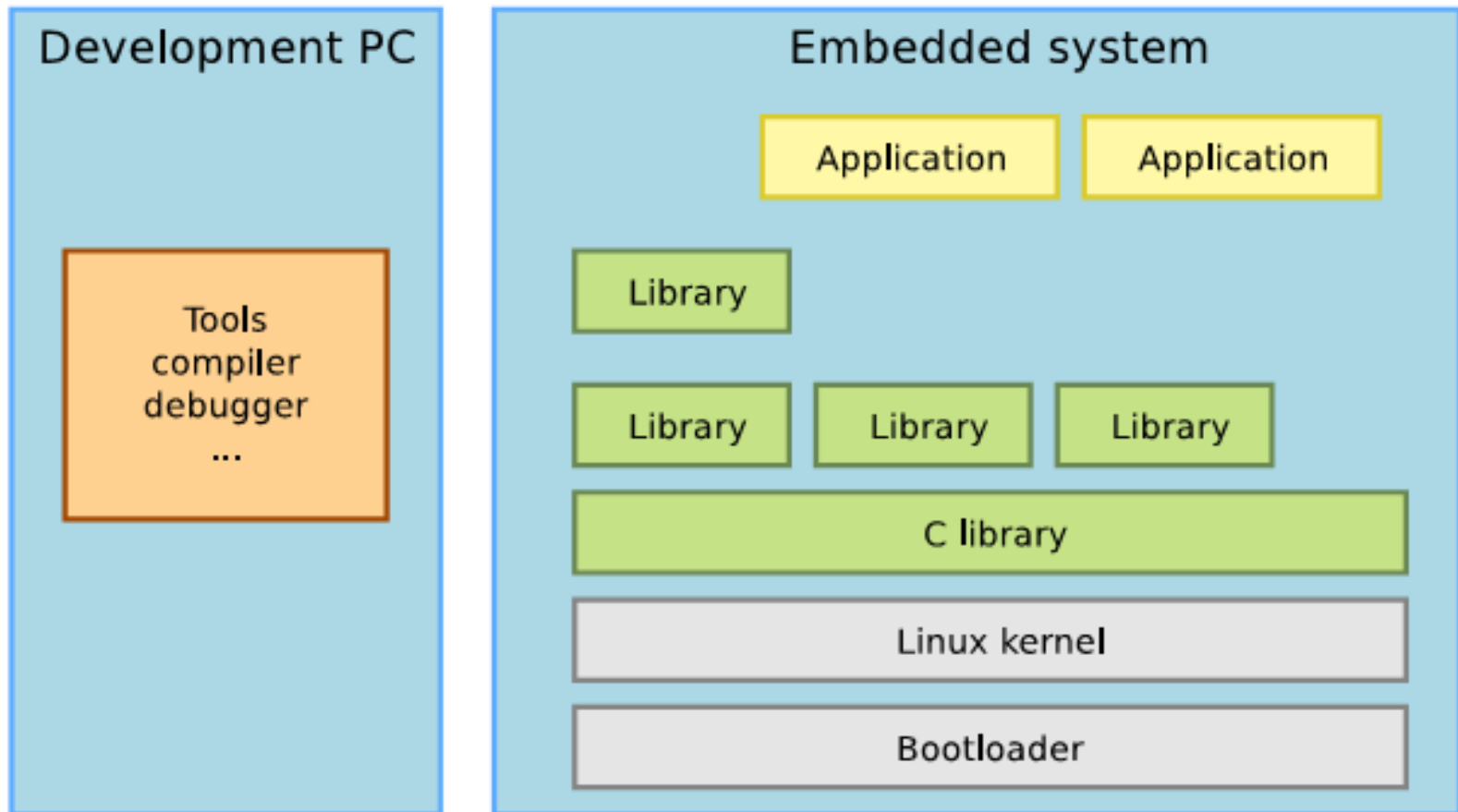
- Hardware requerido
 - Memória RAM
 - Pequenas distribuição podem requerer 8MB
 - Distribuições mais realistas vão requerer ao menos 32MB
 - Memória de armazenamento
 - Um sistema linux básico vai requerer ao menos 4MB
 - Suporta
 - Memória flash
 - Cartões SD/MMC

Linux embarcado

- Requisitos de comunicação
 - Dá suporte a vários tipos de interfaces
 - I2C
 - SPI
 - CAN
 - 1-Wire
 - USB
 - E ainda
 - Ethernet, wifi, bluetooth
 - IPV4, IPV6, TCP, UDP
 - etc

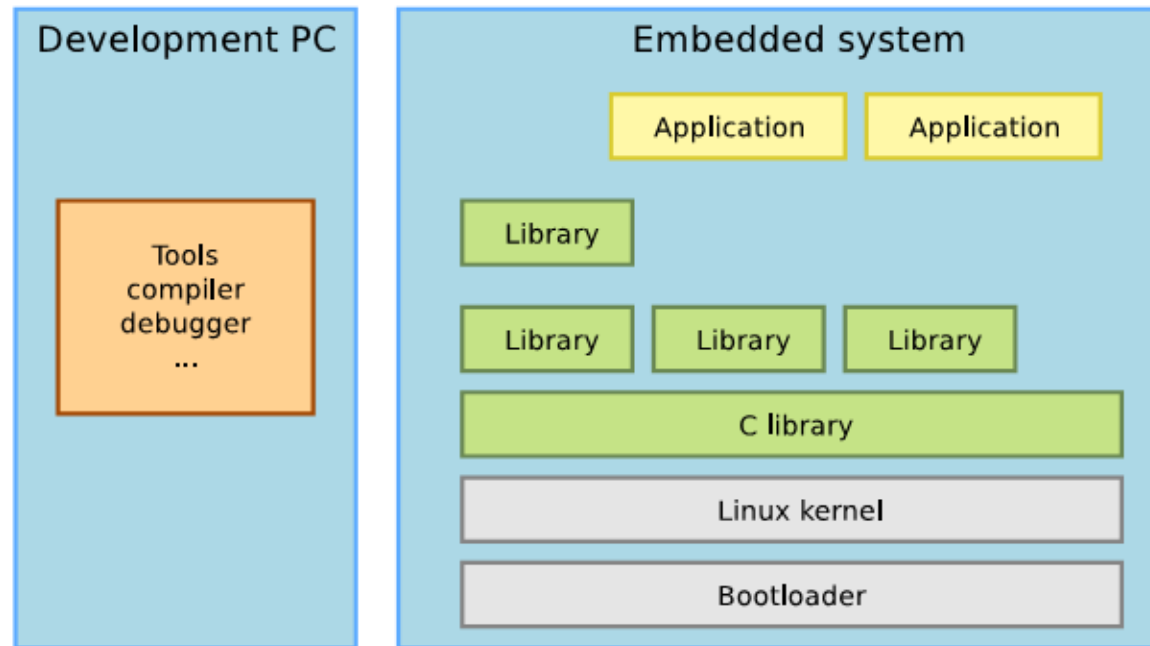
Linux embarcado

- Arquitetura de um sistema de linux embarcado



Linux embarcado

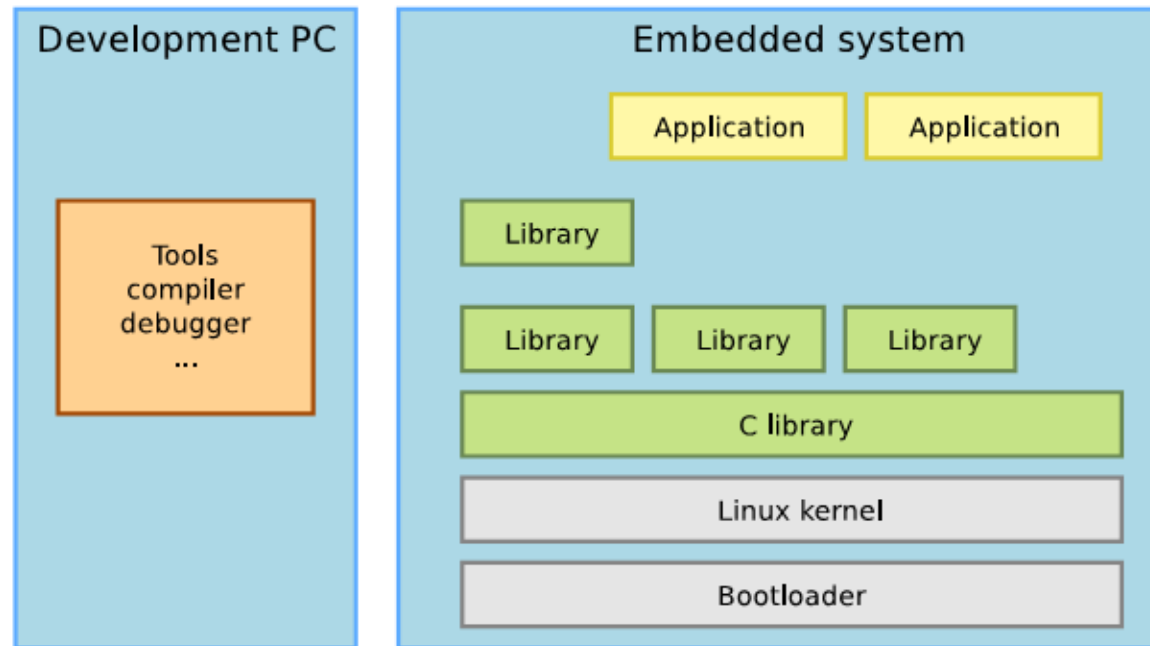
- Arquitetura de um sistema de linux embarcado



- Cross-compilation
 - Compilador usado para gerar o código executável da plataforma alvo

Linux embarcado

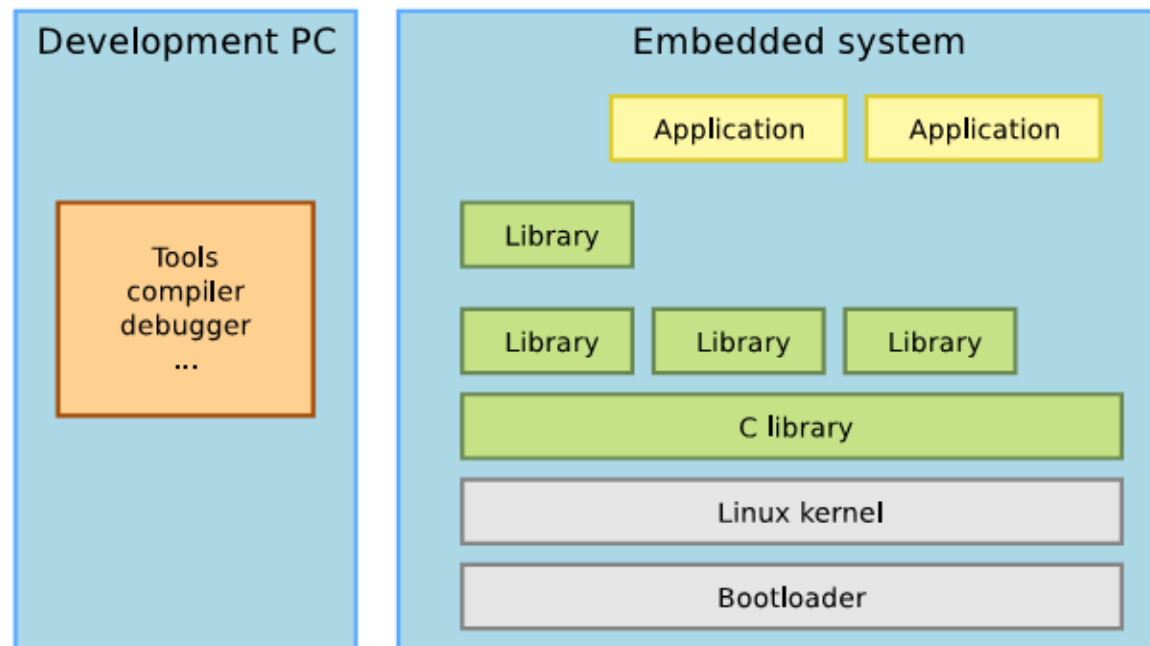
- Arquitetura de um sistema de linux embarcado



- Bootloader
 - Iniciado pelo hardware, responsável pela configuração inicial, carga e execução do kernel
 - No raspberry pi este módulo está no SoC

Linux embarcado

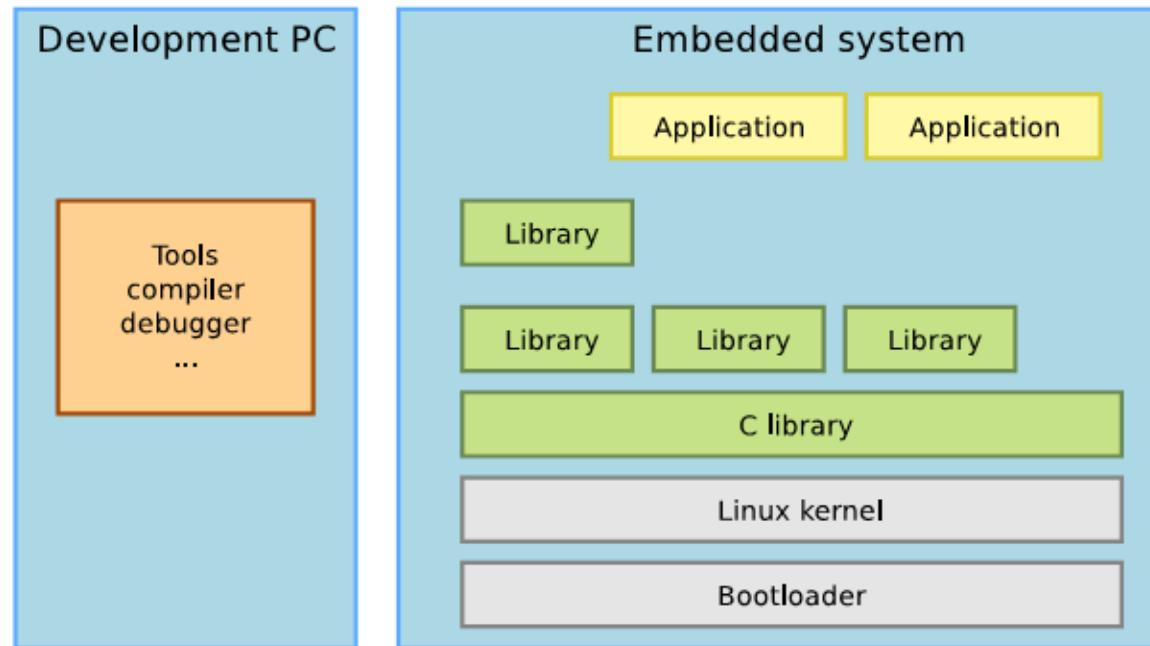
- Arquitetura de um sistema de linux embarcado



- Linux Kernel
 - Contem os recursos de gerenciamento de processo e memória, pilha de rede, device drivers, além de prover recursos para o nível de usuário

Linux embarcado

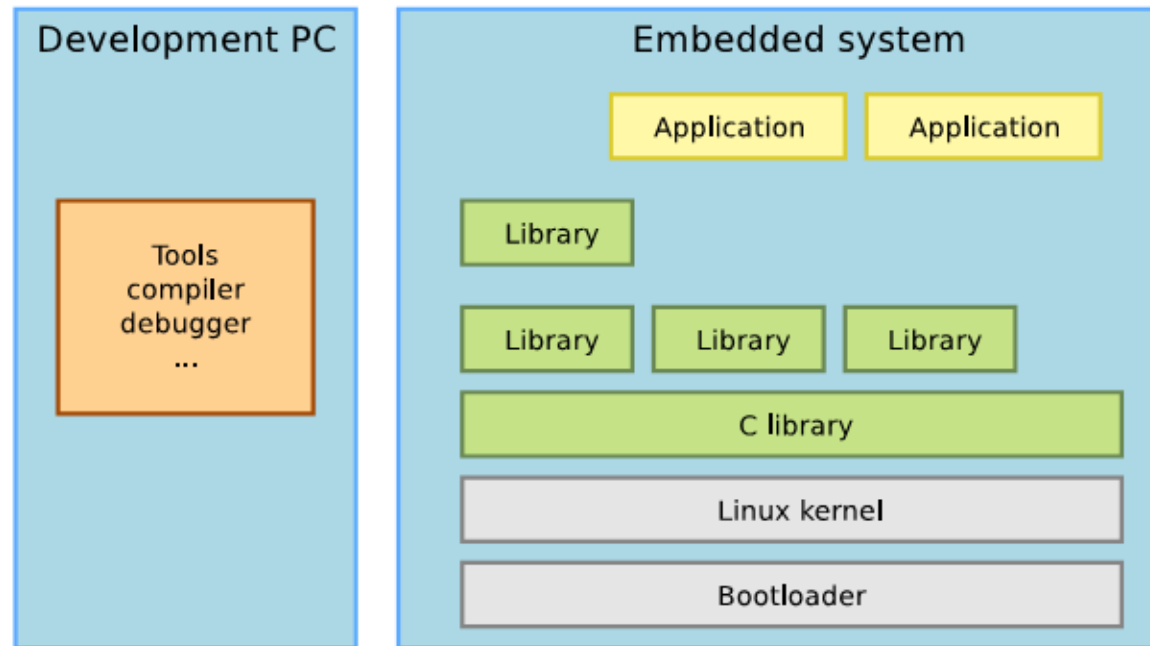
- Arquitetura de um sistema de linux embarcado



- Biblioteca C
 - Interface entre as aplicações de kernel e do espaço do usuário

Linux embarcado

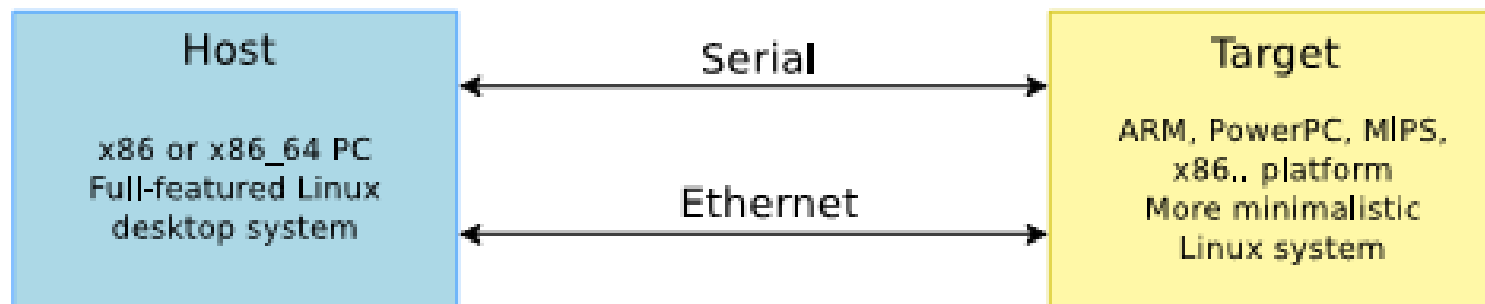
- Arquitetura de um sistema de linux embarcado



- Bibliotecas e aplicações
 - Desenvolvidas pelo usuário ou por outros fornecedores

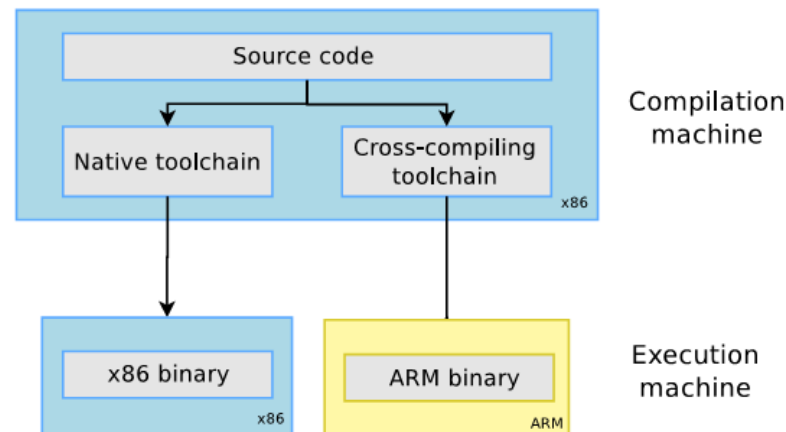
Linux embarcado

- Forma de desenvolvimento embarcado
 - Dividido entre
 - O computador que contem o conjunto de ferramentas de compilação
 - O dispositivo alvo do desenvolvimento
 - Normalmente são conectados por
 - Interface serial para propósito de depuração
 - Ethernet/JTAG para depurações de baixo nível



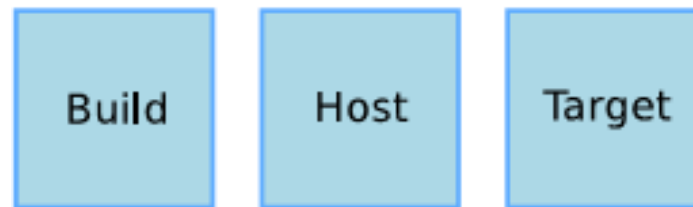
Cross compiler

- Computador
 - Normalmente possui ferramental nativo para geração de código executável pela própria arquitetura
- Cross compiler
 - Utilizados para gerar arquivo executavel para a plataforma alvo
 - Podem ser executados em um computador
 - Código gerado deve ser transferido para o dispositivo alvo



Cross compiler

- Denominações distintas para a criação de ferramental necessário
 - Máquina build: onde será gerado o ferramental
 - Máquina host: onde o ferramental será executado
 - Máquina target: onde os binários gerados serão executados
- Configurações possíveis

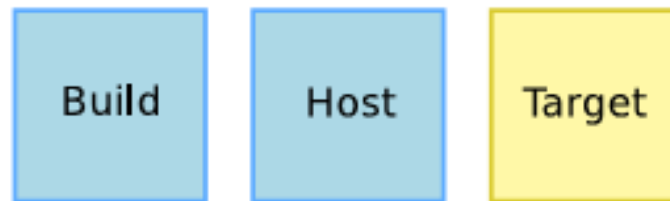


Native build

used to build the normal gcc
of a workstation

Cross compiler

- Denominações distintas para a criação de ferramental necessário
 - Máquina build: onde será gerado o ferramental
 - Máquina host: onde o ferramental será executado
 - Máquina target: onde os binários gerados serão executados
- Configurações possíveis

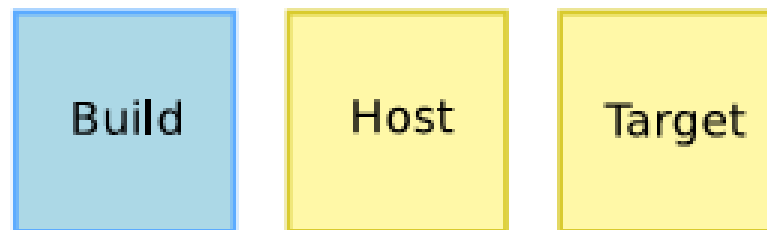


Cross build

used to build a toolchain that runs
on your workstation but generates
binaries for the target

Cross compiler

- Denominações distintas para a criação de ferramental necessário
 - Máquina build: onde será gerado o ferramental
 - Máquina host: onde o ferramental será executado
 - Máquina target: onde os binários gerados serão executados
- Configurações possíveis

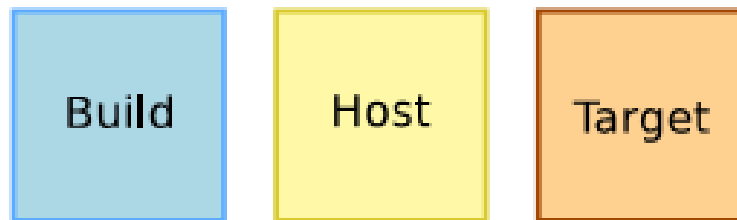


Cross-native build

used to build a toolchain that runs on your target and generates binaries for the target

Cross compiler

- Denominações distintas para a criação de ferramental necessário
 - Máquina build: onde será gerado o ferramental
 - Máquina host: onde o ferramental será executado
 - Máquina target: onde os binários gerados serão executados
- Configurações possíveis



Canadian build

used to build on architecture A a
toolchain that runs on architecture B
and generates binaries for architecture C

Apresentação

- Raspberry pi
- Sistema Operacional embarcado
- ***Atividade***

Atividade

- Iniciar o desenvolvimento do TP1
 - Iniciar o uso do kit
 - Entender o fluxo envolvido na geração de um ukernel
 - Validar o ukernel sobre a plataforma