



嵌入式Linux开发



www.farsight.com.cn

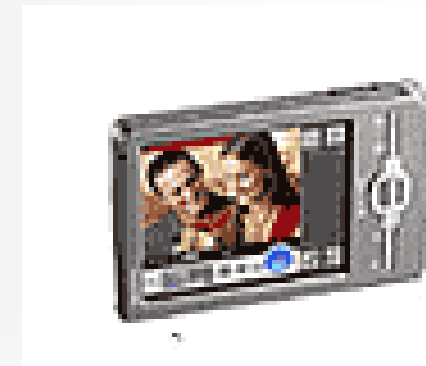
- Ø Linux简介及嵌入式Linux开发初步
- Ø U-boot开发初步
- Ø Kernel开发
- Ø Linux下文件系统开发初步
- Ø Linux下驱动开发要点
- Ø 多线程应用开发
- Ø 网络应用开发
- Ø Linux下图形界面开发
- Ø 华清远见linux培训课程介绍

- ✓ Linus Torvalds于1991编写
- ✓ Linux是一个Unix兼容的系统，大部分通用的Unix工具和程序都可以在Linux系统下运行
- ✓ 使用GNU工具开发：
 - ∅ gcc, glibc, binutils, make等
- ✓ GNU = GNU is Not Unix
- ✓ GPL = General Public License

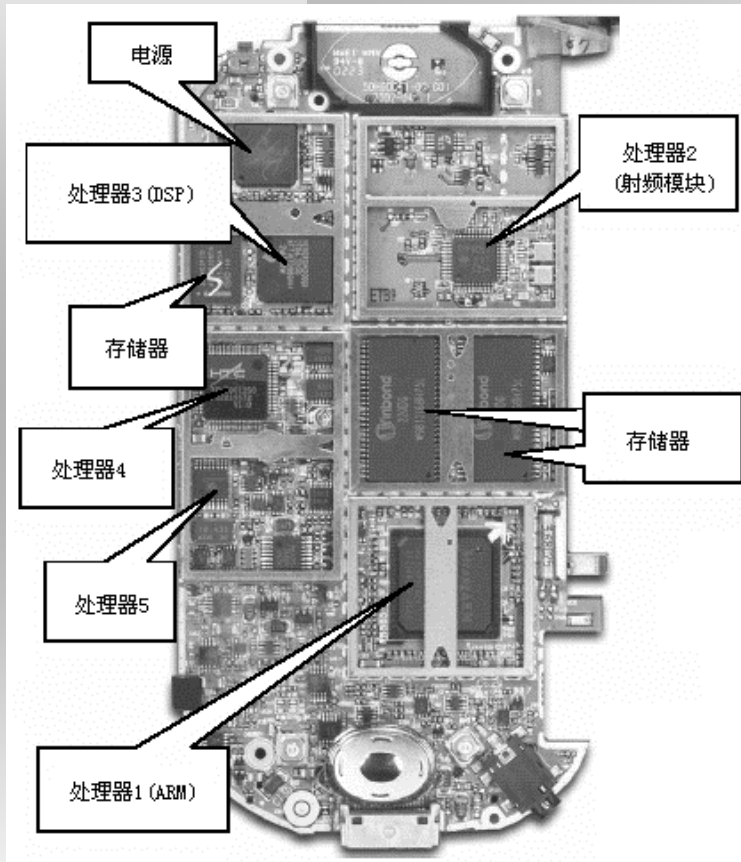
- ✓ 提供了先进的网络支持
- ✓ 多任务、多用户
- ✓ 符合IEEE POSIX标准
- ✓ 核心能仿真FPU
- ✓ 支持数十种文件系统格式
- ✓ 完全运行于**保护模式**
- ✓ 开放源代码
- ✓ 采用先进的内存管理机制，更加有效地利用物理内存

嵌入式系统开发

√ 嵌入式系统是以应用为中心，以计算机技术为基础，并且**软硬件可裁剪**，适用于应用系统对**功能、可靠性、成本、体积、功耗**有严格要求的专用计算机系统。



嵌入式系统与PC机



• 软件 (操作系统/驱动程序/应用程序)

• 存储器^序

• 输入

• 处理器

• 输出

嵌入式系统与PC

嵌入式系统

PC机

引导代码

Bootloader引导，针对不同电路板进行移植

主板的BIOS引导，无须改动

操作系统

WinCE、VxWorks、Linux等，需要移植

Windows、Linux等，不需要移植

驱动程序

每个设备驱动都必须针对电路板进行重新开发或移植，一般不能直接下载使用

操作系统含有大多数驱动程序，或从网上下载直接使用

协议栈

需要移植

操作系统包括，或第三方提供

开发环境

借助服务器进行交叉编译

在本机就可开发调试

仿真器

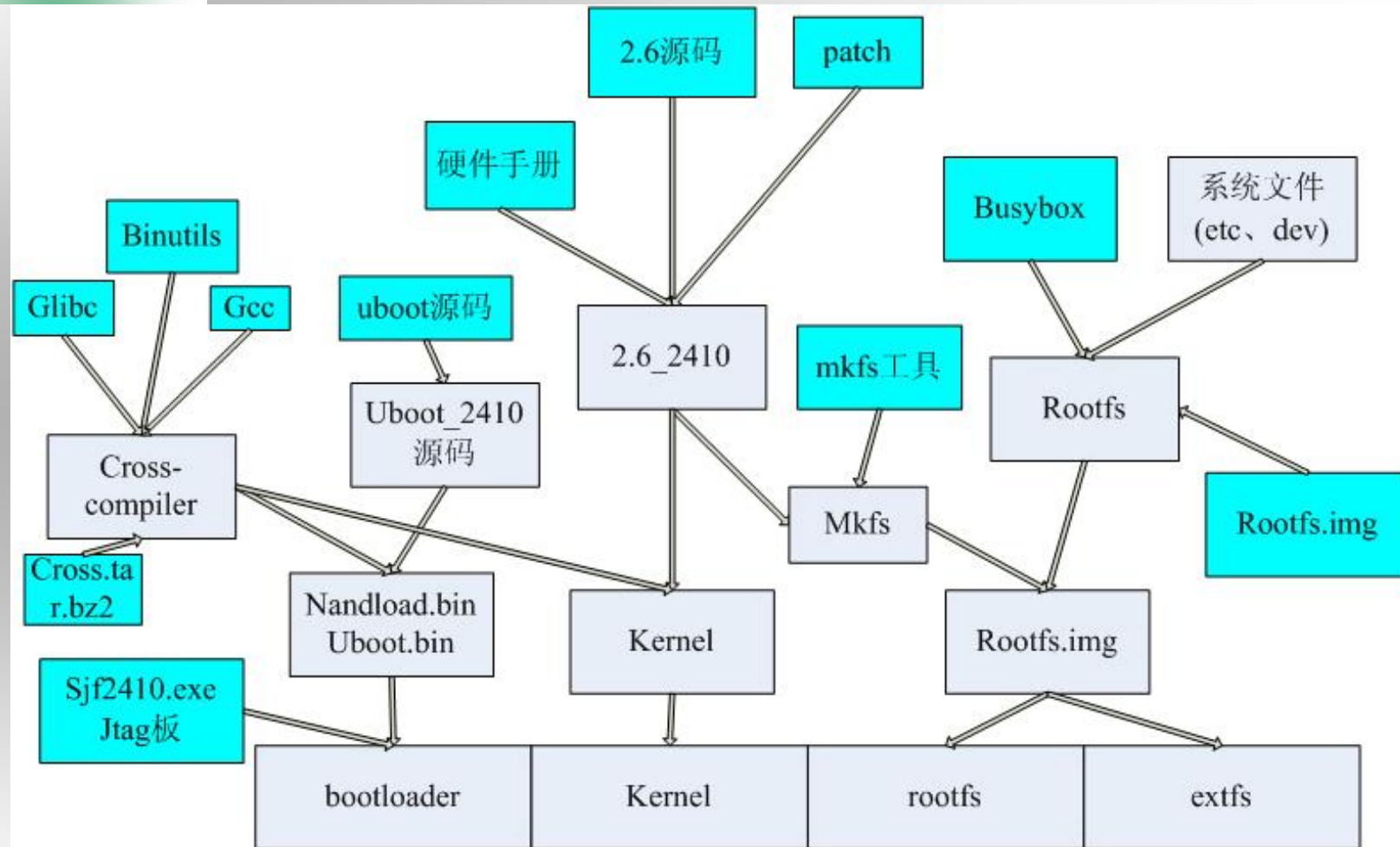
需要

不需要

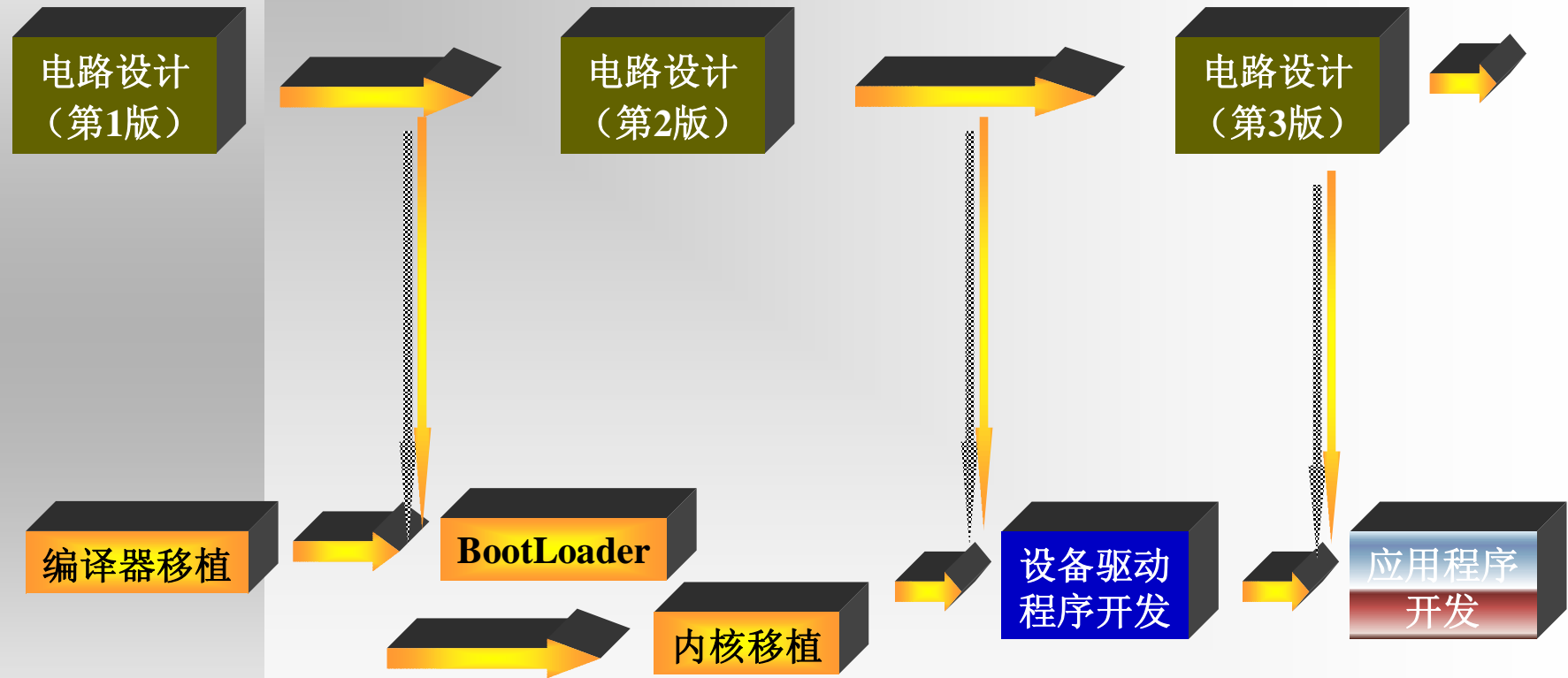
嵌入式系统的特点

- ✓ 1) 嵌入式系统低功耗、体积小，专用性强。嵌入式系统与PC机的最大不同就是嵌入式CPU大多工作在为特定用户群设计的系统中，能够把PC机中许多由板卡完成的任务集成在芯片内部，从而有利于嵌入式系统设计趋于小型化。
- ✓ 2) 为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存贮于磁盘等载体中。
- ✓ 3) 嵌入式系统的硬件和软件都必须高效率地设计，系统要精简。操作系统一般和应用软件集成在一起。
- ✓ 4) 对软件代码质量要求很高。应该尽最大可能避免死机的情况发生。
- ✓ 5) 嵌入式系统开发需要专门的开发工具和开发环境。

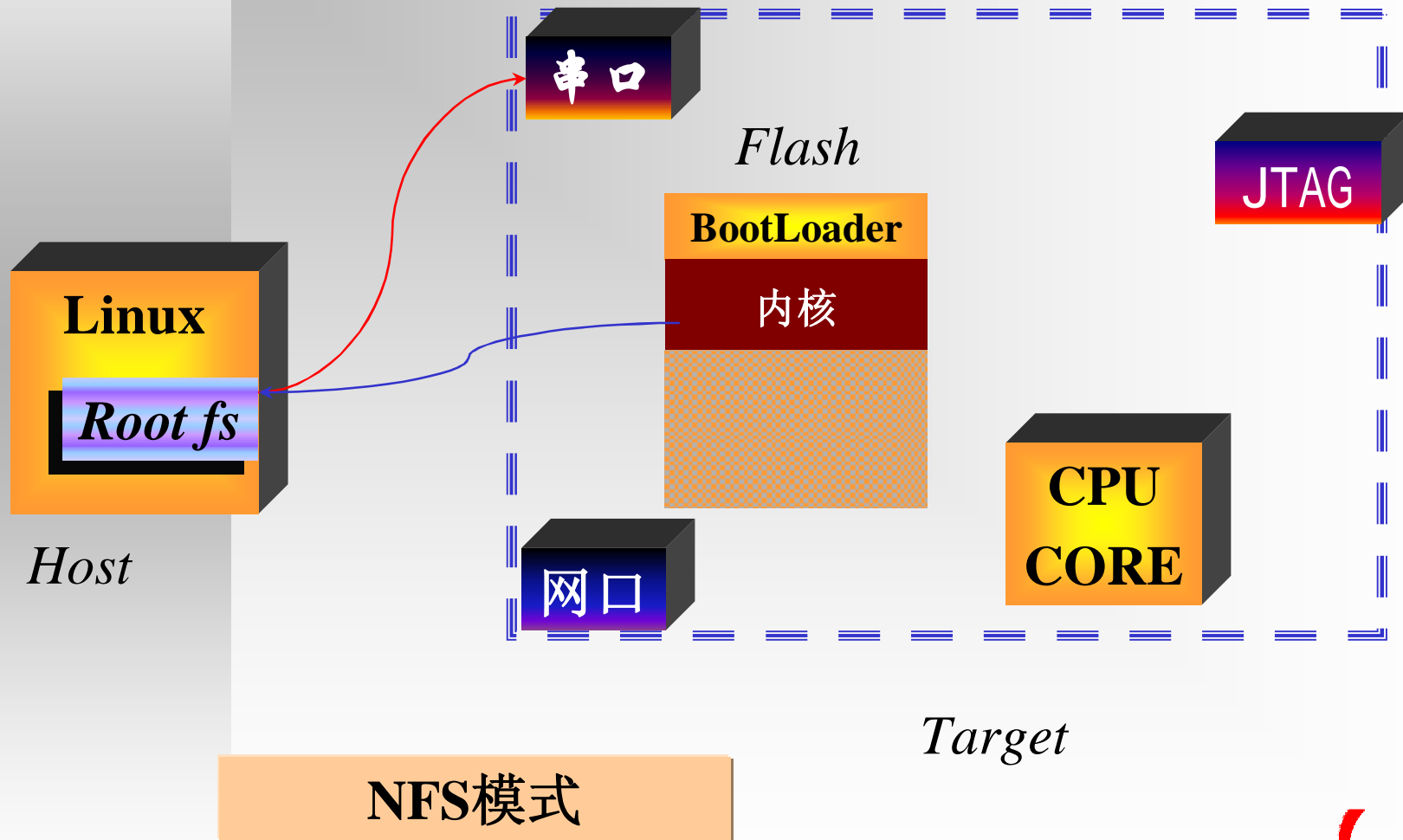
嵌入式系统开发的内容



嵌入式Linux开发流程



嵌入式Linux开发模式

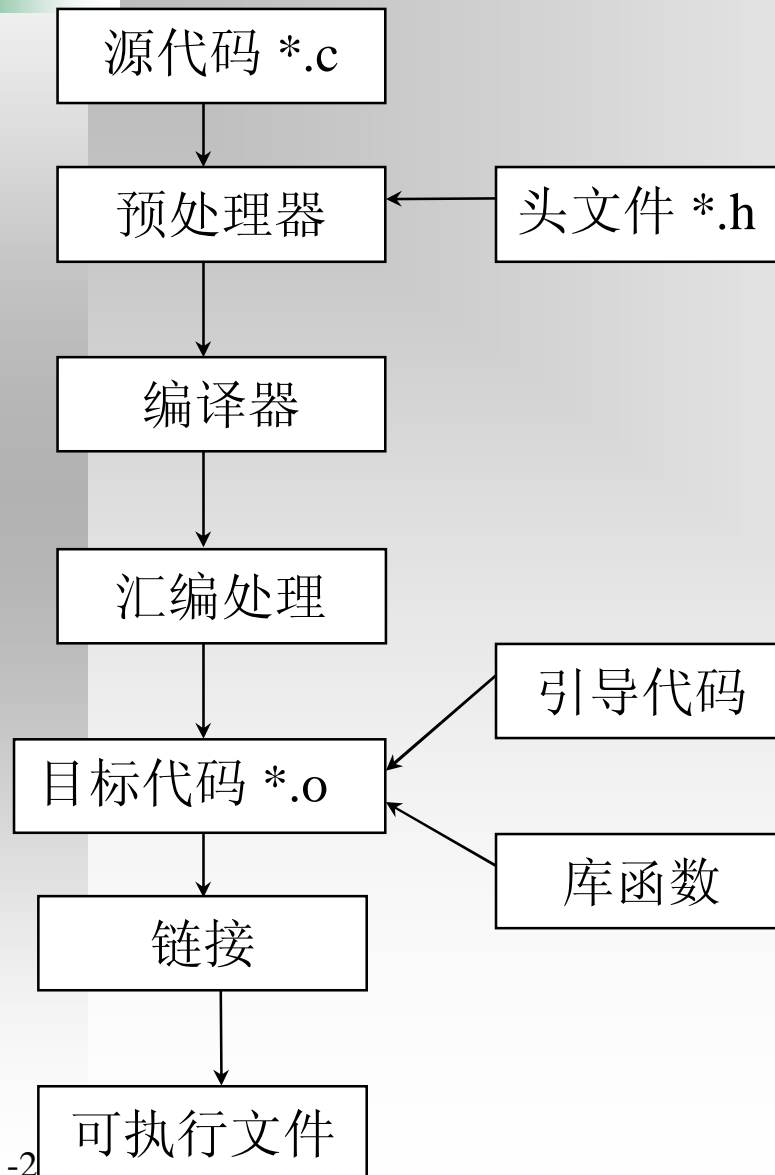


- √ 全称为GNU CC，GNU项目中符合ANSI C标准的编译系统
- √ 编译如C、C++、Object C、Java、Fortran、Pascal、Modula-3和Ada等多种语言
- √ 一个交叉平台编译器，适合在嵌入式领域的开发编译

编译器的作用

√ Gcc的编译流程分为四个步骤

- Ø • 预处理(Pre-Processing)
- Ø • 编译(Compiling)
- Ø • 汇编(Assembling)
- Ø • 链接(Linking)



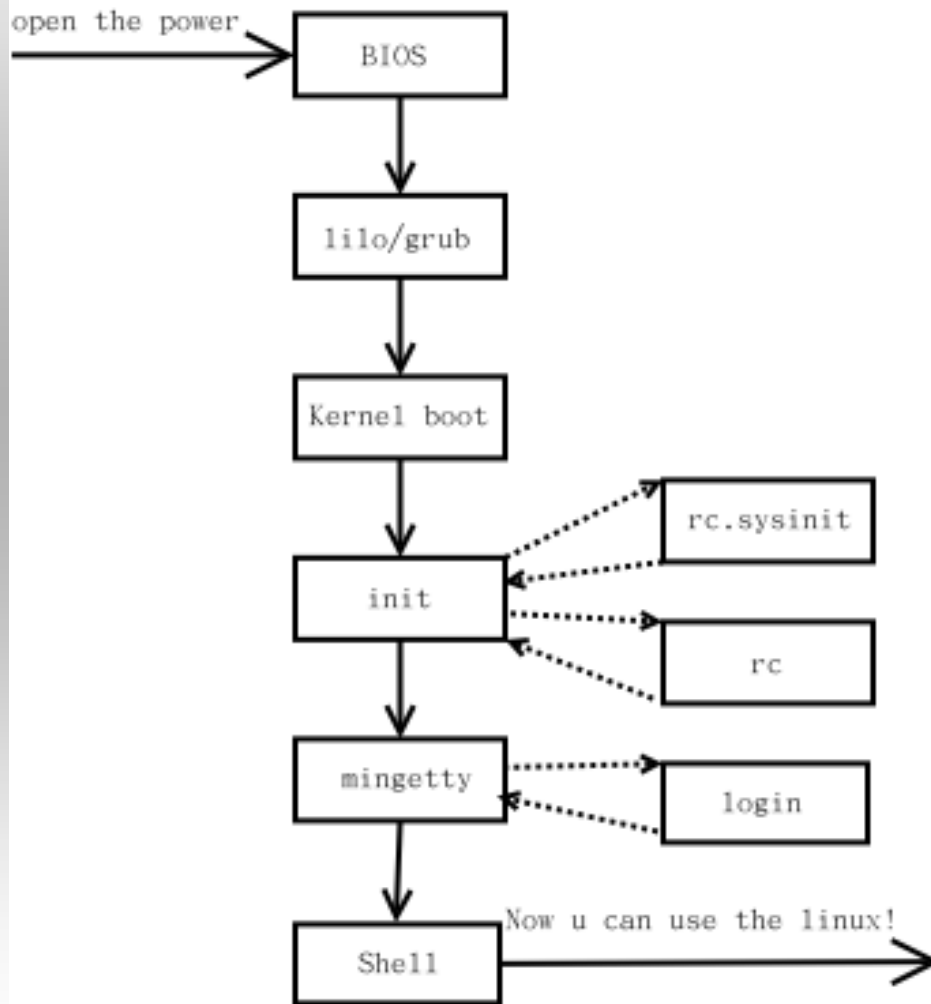
GCC交叉编译器的生成

- ✓ 第1步，取得Binutils、GCC、Glibc 源码。
- ✓ 第2步，配置并编译Binutils 取得我们所需要的汇编和连接程序。
- ✓ 第3步，配置并编译GCC 源码生成GCC 编译器。一般是C 编译器首先生成，然后以这个为基础在结合下一步生成的Glibc 的C 函数库，再编译生成其它编译器。
- ✓ 第4步，配置Glibc 并编译生成Glibc 的C 函数库。
- ✓ 第5步，再次配置和编译GCC 源码，生成其它语言的编译器，如C++编译器、生成其它语言的编译器，如C++编译器、生成其它语言的编译器，如C++编译器等。

Make 工程管理器

- √ 工程管理器，顾名思义，是指管理较多的文件
- √ Make 工程管理器也就是个“自动编译管理器”，这里的“自动”是指它能根据文件时间戳自动发现更新过的文件而减少编译的工作量，同时，它通过读入 Makefile 文件文件的内容来执行大量的编译工作

Linux Boot Step



初始化硬件设备
载入内核
执行内核

自解压
初始化驱动程序
挂载根文件系统
执行第一个用户空间程序 (linuxrc)

配置用户空间
启动用户服务

Bootloader的概念

- √ BootLoader就是在操作系统内核或用户应用程序运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核或用户应用程序准备好正确的环境。

常见的Bootloader

- ❖ 下载源代码
 - ❖ 阅读Readme文件
 - ❖ 了解u-boot结构
 - ❖ 添加修改
 - ❖ 编译
 - ❖ 调试
 - ❖ 发布
- ❖ `include/configs/smdk2410.h`
 - ❖ `board/smdk2410/smdk2410.c`
 - ❖ `int board_int(void)`
 - ❖ `int dram_init(void)`
 - ❖ `board/smdk2410/flash.c`
 - ❖ `cpu/arm920t/serial.c`
 - ❖ `drivers/rtl8019.c`

移植u-boot的步骤

✓ 下载源代码

✓ include/configs/smdk2410.h

✓ 阅读Readme文件

✓ board/smdk2410/smdk2410.c

✓ 了解u-boot结构

✓ int board_init(void)

✓ 添加修改

✓ int dram_init(void)

✓ 编译

✓ board/smdk2410/flash.c

✓ 调试

✓ cpu/arm920t/serial.c

✓ 发布

✓ drivers/rtl8019.c

√ 内核子系统

- ∅ 进程调度
- ∅ 内存管理
- ∅ 内存管理
- ∅ 虚拟文件系统
- ∅ 网络
- ∅ 进程间通讯

√ 内核目录结构： /usr/src/Linux-***

∅ /arch (重点)

∅ /drivers

∅ /fs

∅ /include

∅ /init

∅ /ipc

∅ /kernel

∅ /mm

∅ /net

√ 为什么要编译内核？

√ 编译模式

∅ 直接编译进内核

∅ 编译成模块

√ 目的是创建嵌入式Linux

√ 裁剪的对象包括：

∅ Linux内核

∅ 库

∅ 守护进程

∅ GUI

文件系统

- √ 文件系统 (FileSystem) 是一组程序，它们告诉操作系统如何访问及解释存储在磁盘或磁带驱动器或者其它存储媒介上的内容
- √ 常见的文件系统包括：FAT 和 FAT-32 (DOS/Windows)、HPFS (OS/2)、NFS、NTFS (Windows NT/2000) 以及其它文件系统

√ 嵌入式文件系统技术

- Ø Flash简介
- Ø ramdisk技术
- Ø cramfs、romfs
- Ø jffs2、yaffs
- Ø MTD技术

嵌入式驱动开发

- √ 驱动程序简介
- √ Linux设备驱动分类
- √ Linux设备驱动模型
- √ 编写字符设备驱动程序

Linux设备驱动的概念

- ✓ 驱动程序为硬件提供一个定义良好的内部接口
- ✓ 驱动程序封装了硬件细节
- ✓ 驱动程序为应用程序提供了访问设备的机制
- ✓ 驱动程序是内核的一部分

Linux设备和模块的分类

√ 设备和模块的分类：

∅ 字符设备

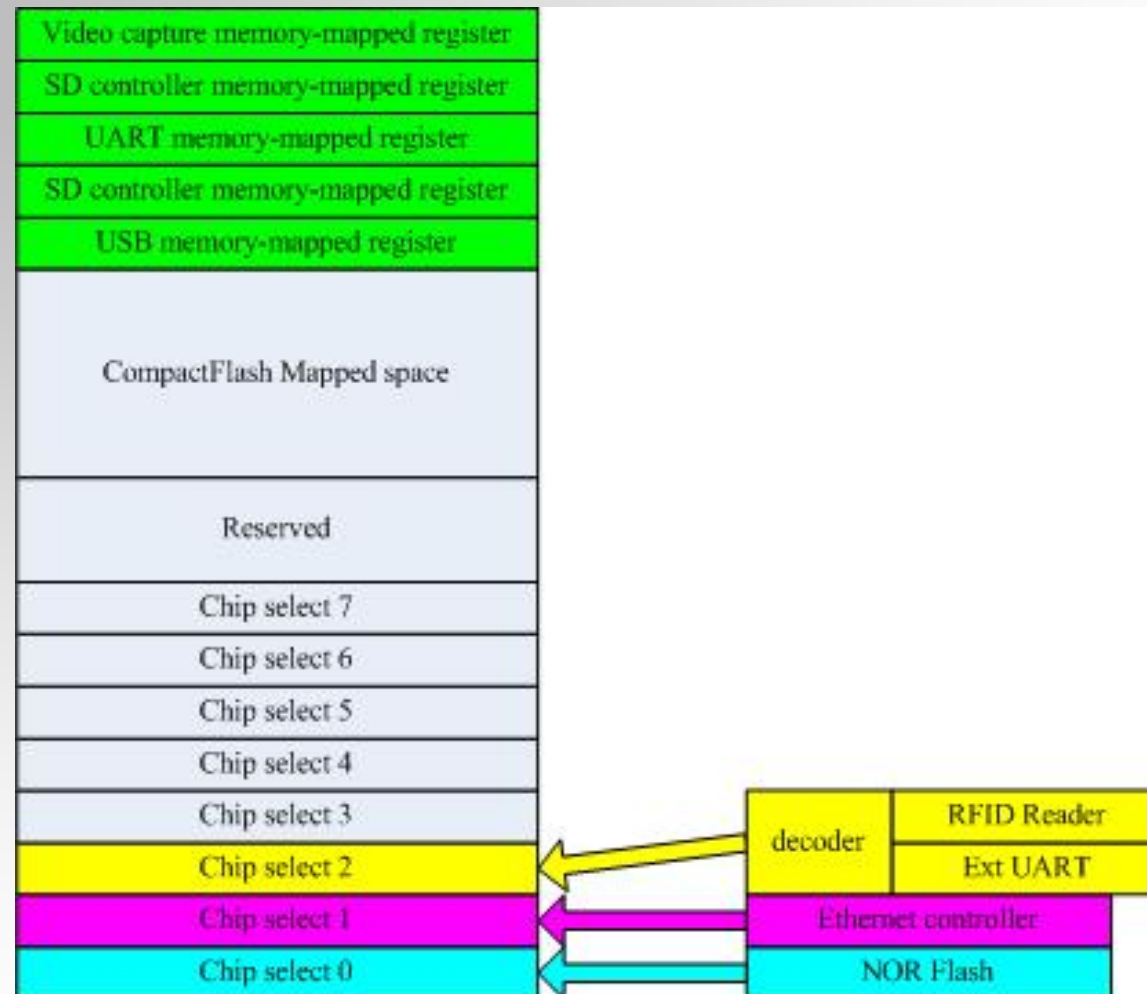
∅ 块设备

∅ 网络接口

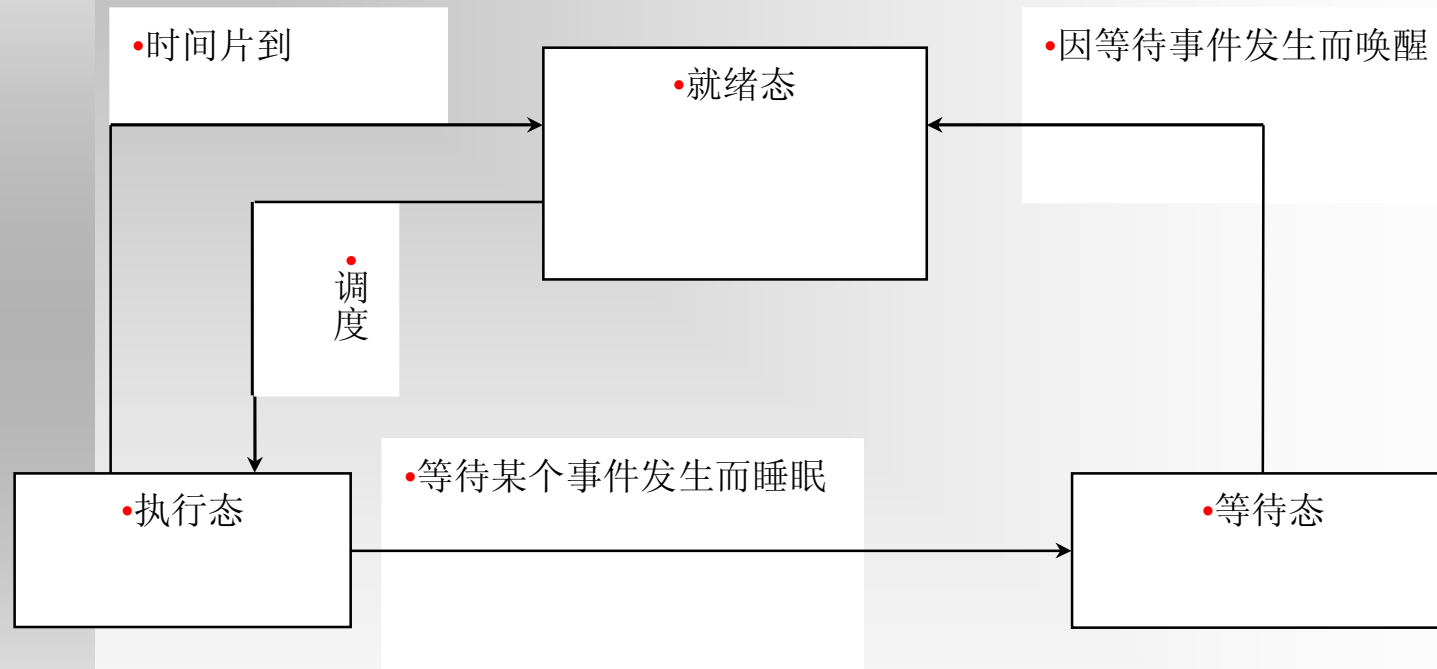
∅ 提供公共服务的特定类型设备

√ 其中字符设备和块设备是最主要的设备文件类型

典型的嵌入式设备存储器映射

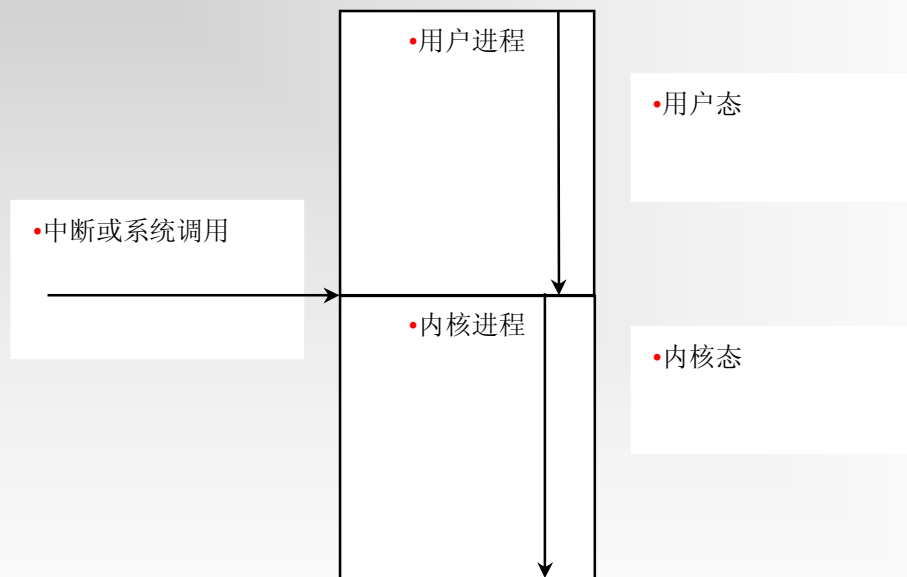


进程运行的状态



Linux 下的进程的模式和类型

✓ 进程的执行模式划分为用户模式和内核模式



Linux 下进程间通信概述

✓ Unix 平台上进程通信手段继承而来

✓ 两大主力

∅ AT&T 的贝尔实验室，对 Unix 早期的进程间通信手段进行了系统的改进和扩充，形成了 "system V IPC"，其通信进程主要局限在单个计算机内

∅ BSD（加州大学伯克利分校的伯克利软件发布中心），跳过了该限制，形成了基于套接□（socket）的进程间通信机制

✓ Linux 则把两者的优势都继承了下来

√ 网络编程与普通程序的区别

- ∅ 客户端与服务器模式

√ Linux网络编程

- ∅ 提供套接字(socket)

- ∅ 通讯的文件描述符

- ∅ 设备无关性

√ 网络编程与普通程序的区别

- ∅ 客户端与服务器模式

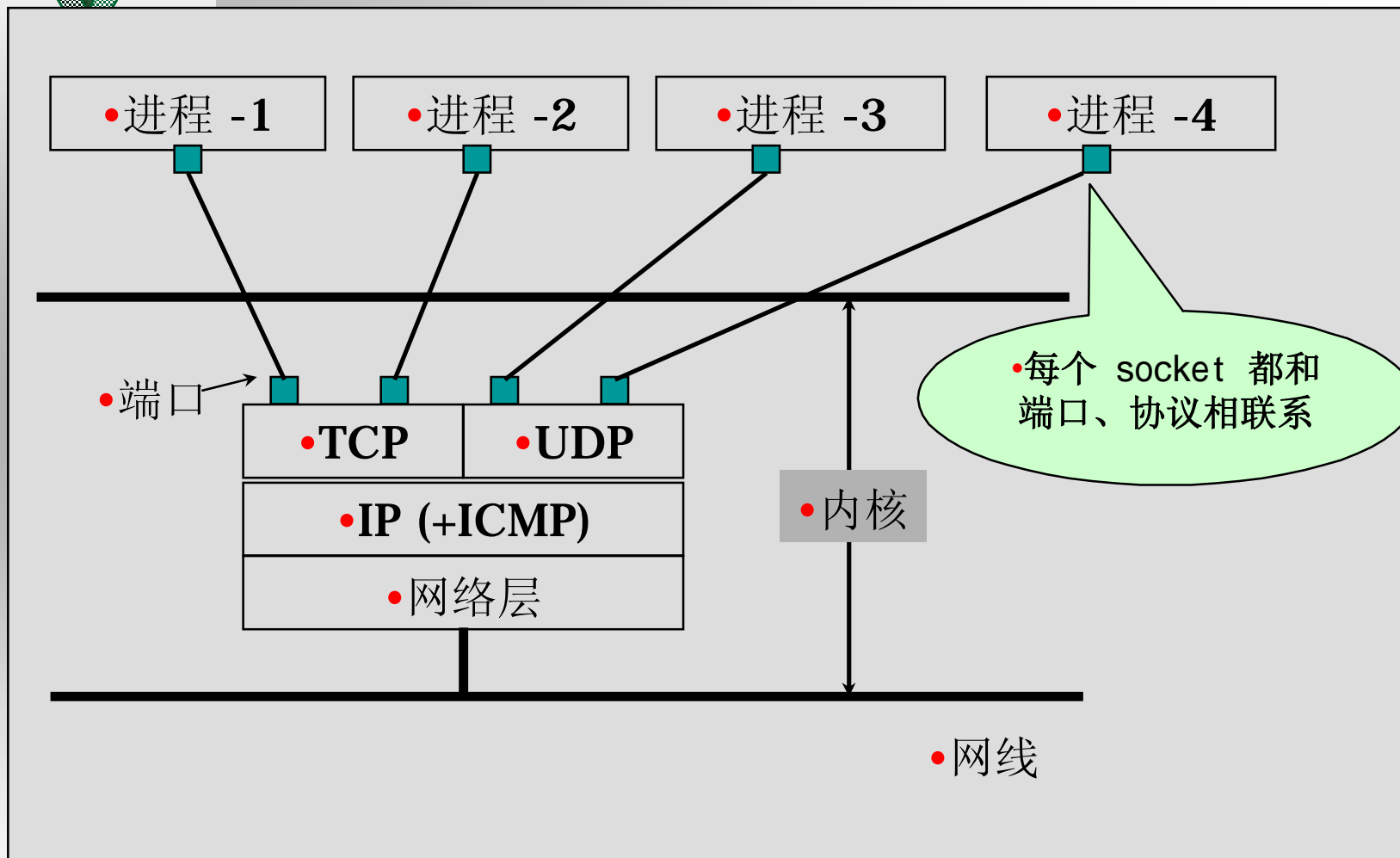
√ Linux网络编程

- ∅ 提供套接字(socket)

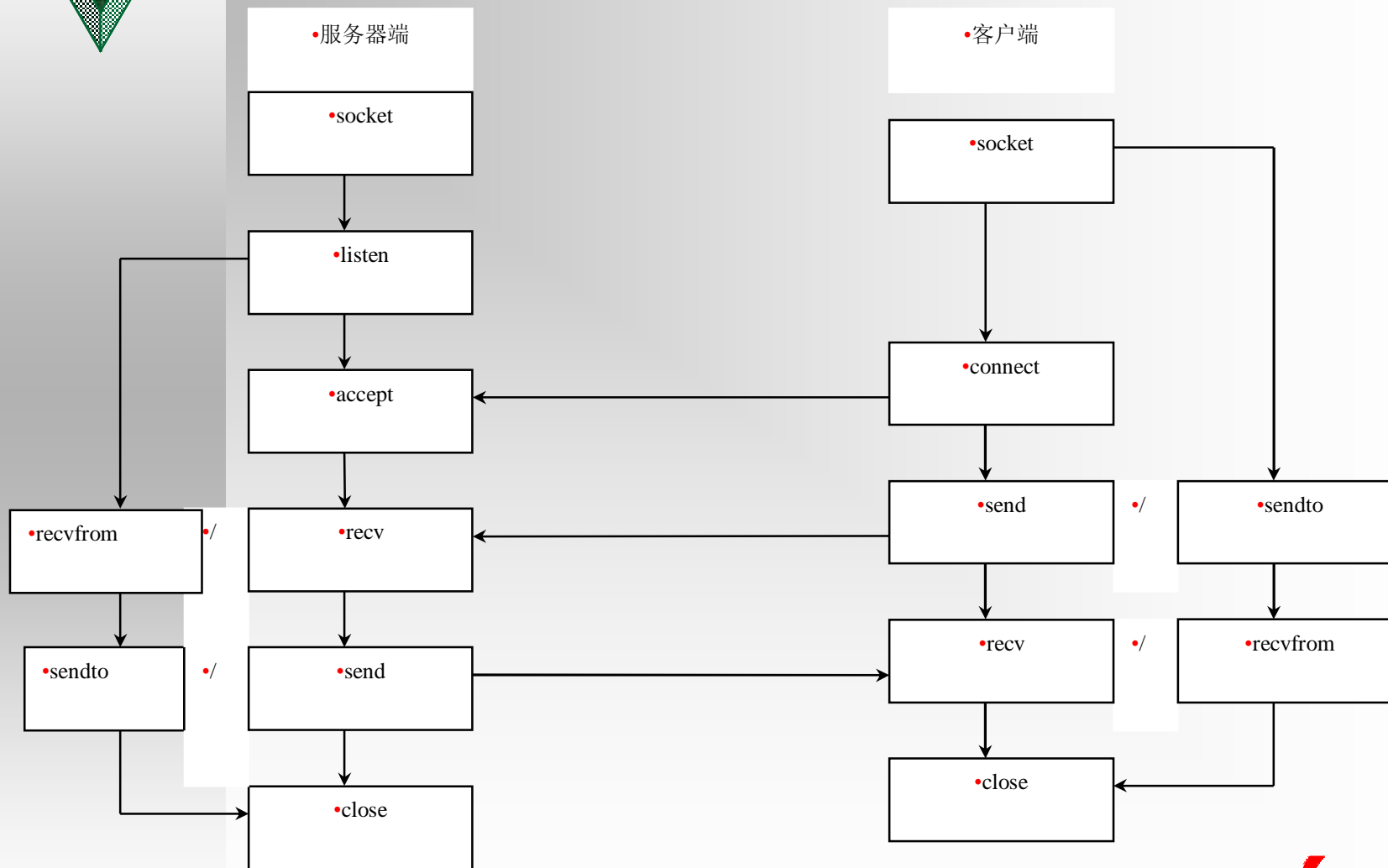
- ∅ 通讯的文件描述符

- ∅ 设备无关性

套接字和端口



TCP/UDP协议socket编程流程



✓ Linux有一套简便易学的图形用户接口(GUI)，用户使用鼠标就可以完成大多数工作。在Linux中，GUI由窗口系统，窗口管理器，工具包和风格等几个部分组成。窗口系统用于组织显示屏上的图形输出，窗口管理器用于对窗口的操作如最小化等，工具包是用于编程界面的库，风格是应用程序的用户界面。

	Applications	Games	Settings	Documents		
GUI	Qt/Embedded					
	Qtopia					
Kernel	Arm linux kernel					
			JFFS2			
	Driver		MTD	fb		
Board	CPU	SDRAM	Bootloader	FLASH	LCD	...

✓ Linux有一套简便易学的图形用户接口(GUI)，用户使用鼠标就可以完成大多数工作。在Linux中，GUI由窗口系统，窗口管理器，工具包和风格等几个部分组成。窗口系统用于组织显示屏上的图形输出，窗口管理器用于对窗口的操作如最小化等，工具包是用于编程界面的库，风格是应用程序的用户界面。

	Applications	Games	Settings	Documents		
GUI	Qt/Embedded					
	Qtopia					
Kernel	Arm linux kernel					
			JFFS2			
	Driver		MTD	fb		
Board	CPU	SDRAM	Bootloader	FLASH	LCD	...

嵌入式系统中常见的GUI

- ✓ Tiny-X，是标准X-windows在嵌入式系统的小巧实现，作为一个图形环境，X-window是成功的，但由于在体系接口上的原因，限制了它对游戏、多媒体的支持能力。
- ✓ MicroWindows，其主要特色在于提供了C/S体系结构，同时也提供了相对完善的图形功能。但却无任何硬件加速能力，图形引擎中也存在着许多未经优化的低效算法。
- ✓ MiniGUI，是建立在比较成熟的图形引擎之上，其特点是小巧精致。它尽量保持与Win32的兼容，这样在Win CE应用的场合，也可以使用MiniGUI。



华清远见

嵌入式Linux课程简介

- ✓ 嵌入式Linux应用开发班
- ✓ 嵌入式Linux系统开发班
- ✓ 嵌入式Linux驱动开发班

2007-11-24

The logo for FAR SIGHT is a green inverted triangle with a textured, wood-grain-like interior. The word "FAR SIGHT" is written in white, bold, sans-serif capital letters across the center of the triangle. A red, curved line separates the word "FAR" from "SIGHT".

FAR SIGHT

The success's road

A horizontal rectangular banner with a dark blue background and a light-colored border. It contains the Chinese characters "卓识源于远见" written in white, elegant calligraphy.

卓识源于远见

谢谢!