



嵌入开发环境搭建 与系统移植

Version 1.0

西安电子科技大学

回顾

1. 交叉编译 (宿主机/目标机)
2. 远程登录/远程控制 (串口、ssh)
3. 交叉调试 (远程调试)
4. 嵌入式系统启动过程 (bootloader -> 操作系统)

一个实际问题

老板给了你一块开发板，说明了CPU类型(例如ARM)和硬件相关配置（内存、Flash存储器、JTAG口、USB口、串口、网口、触摸屏、专用外设...），

要求你：

- (1) 在开发板上移植一个Linux内核，
- (2) 编写满足项目需求的Linux应用程序，
- (3) 该应用程序要求有GUI界面，
- (4) 该应用程序能够控制专用外设。

你该先做什么？再做什么？

如何从零开始构建嵌入式Linux系统

1. 在宿主机上建立交叉编译环境(arm-linux-gcc)
2. 构建Bootloader，生成Bootloader镜像
3. 裁剪并编译Linux内核，生成内核镜像(zImage)
4. 构建根文件系统，生成根文件系统镜像(rootfs.img)
5. 将3类镜像依次下载到开发板（怎么下？）
6. 在宿主机和开发板之间建立好远程登录环境
7. 在宿主机和开发板之间建立好文件下载环境
8. 在宿主机和开发板之间建立好远程调试环境
9. 在开发板上移植第三方开发库（如QT GUI开发库、SQLite库等）
10. 编写Linux应用程序或驱动程序，交叉编译后下载到开发板观察运行效果，或进行远程调试

如何构造交叉编译环境

1. 从gcc.gnu.org上下载gcc源码，用宿主机上的gcc (host-gcc)程序将其编译，生成针对目标机的target-gcc;
2. 用target-gcc生成target-glibc库;

嵌入式交叉编译环境搭建

- 宿主机上toolchain包括binutils、gcc、glibc、glibc-linuxthreads等
 - gcc是用来生成交叉编译器的，主要生成arm-linux-gcc交叉编译工具
 - binutils主要用于生成一些辅助工具，如readelf、objcopy、objdump、ar、ldd等;
 - Ldd列出一个可执行文件在运行时所需要的共享库
 - Objdump显示一个目标文件中所有信息（反汇编text节中的指令）
 - Realelf显示一个目标文件的完整结构
 - Size列出目标文件中节的名字和大小
 - Nm列出一个目标文件的符号表中定义的符号
 - Strip从目标文件中删除符号表信息
 - Ar创建静态库
 - glibc主要是提供用户程序所使用的一些基本的函数库
 - Ulibc, libc
 - glibc-linuxthreads是线程相关函数库。

如何构造并下载bootloader

1. 下载开源bootloader(uboot)源码，进行定制或移植；
2. 在宿主主机上用target-gcc将uboot编译成能够在目标机上运行的bootloader映像文件；
3. 系统加电或复位后，所有CPU都会从某个地址开始执行，这是由处理器设计决定的。ARM处理器在复位时从地址0x00000000取第一条指令。嵌入式系统的开发板都要把板上ROM或Flash映射到这个地址。因此，必须用JTAG把Bootloader程序存储在相应的Flash位置。系统加电后，CPU将首先执行它。

“Bootloader”和“Monitor”

1. 严格来说，“Bootloader”只是引导设备并且执行主程序的固件；而“Monitor”还提供了更多的命令行接口，可以进行调试、读写内存、烧写Flash、配置环境变量等。
2. “Monitor”在嵌入式系统开发过程中可以提供很好的调试功能，开发完成以后，就完全设置成了一个“Bootloader”。所以，习惯上大家把它们统称为Bootloader。

U-Boot

1. U-Boot最早是DENX软件工程中心的Wolfgang Denk基于8xxrom的源码创建了PPCBOOT工程，并且不断添加处理器的支持。后来，Sysgo Gmbh把PPCBOOT移植到ARM平台上，创建了ARMBOOT工程。然后以PPCBOOT工程和ARMBOOT工程为基础，创建了U-Boot工程。
2. U-Boot是ARM平台事实上的标准bootloader
3. 很多厂商对U-BOOT进行定制
4. 常用命令 `bootm bootp go iminfo loadb loads nfs printenv rarpboot run setenv`

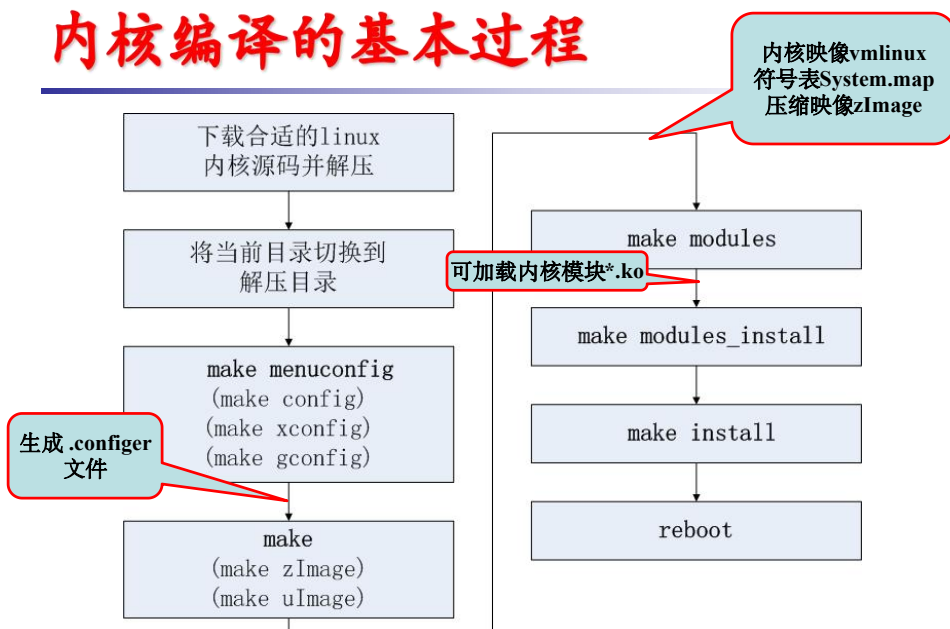
如何编译Linux内核

1. 下载Linux内核源码，进行定制或修改源码移植；
2. 在宿主主机上用target-gcc将Linux内核源码编译成能够在目标机上运行的内核映像文件；
 - 修改makefile
`ARCH ? = $(SUBARCH)`
修改为: `ARCH ? = arm`
`CROSS_COMPILE ?=`
修改为: `CROSS_COMPILE ?= arm-linux-`
 - `make menuconfig`
 - `make uImage`
 - `make modules`

需要重新编译内核的场景

1. 正在使用的内核版本过旧、或是发行商提供的非标准内核；
2. 正在使用的内核过于臃肿，需要进行裁剪；
3. 需要在一个新的嵌入式目标机上移植Linux（需要交叉编译）；
4. 开发嵌入式Linux驱动时，必须首先在开发用宿主机上交叉编译出目标机上的Linux内核。（编译驱动时会用到编译目标内核时生成的makefile文件、头文件和符号表文件）

内核编译的基本过程



make menuconfig的工作原理

1. 内核源码目录树的很多目录下都有一个Kconfig文件和一个Makefile文件;
2. 从内核源码树的根目录开始递归地进入到各级子目录, 依次搜索各个目录下的Kconig文件;
3. 根据这些Kconfig文件生成相应的菜单选项, 供用户选择; 配置界面以树状的菜单形式组织, 主菜单下有若干个子菜单, 子菜单下又有子菜单或配置选项。每个子菜单或选项可以有依赖关系, 只有被依赖项的父项已经被选中, 子项才会显示。
4. 对于各个选项, 用户一般可以选择“Y” (包含该选项), “N” (不包含该选项), “M” (将该选项编译为可加载内核模块)
5. 将用户的选择保存在config文件中

西安电子科技大学

make menuconfig



make menuconfig的配置选项

1. CPU类型和特性配置
2. 电源管理配置
3. 总线配置
4. 网络配置
5. 设备驱动配置
6. 文件系统配置
7. 块设备层配置
8. 可加载内核模块配置
9.

15

西安电子科技大学

用make工具编译内核的过程

1. Make工具从内核源码树的根目录开始，根据.config的设置，依次进入被用户选中的各个目录，并根据该目录下的Makefile文件的指示进行编译。
2. 编译完成之后，一般会在根目录下找到未压缩的内核映像vmlinux和内核符号表文件System.map，在arch/arm/boot下找到压缩的内核映像zImage。

16

西安电子科技大学

如何在内核中增加自己的程序

1. 将编写的源码考入Linux内核源代码的相应目录。
2. 在该目录的Kconfig中增加关于新源码对应项目的配置选项。
3. 在该目录的Makefile文件中增加对新源代码的编译条目。

构造根文件系统

1. 利用工具箱BusyBox, 它集成压缩了Linux的许多工具和命令。(实验中利用了mkfs.jffs2工具)
2. 建立必须的子目录 dev etc home lib mnt opt proc sys tmp var
3. 加入glibc动态库
4. 加入需要动态加载的驱动(*.ko文件)
5. 配置/etc下的启动脚本和配置文件
6. 加入应用程序映像
7. 将以上内容打包制成jffs/yaffs/cramfs映像

实验中如何下载Linux内核与根文件系统

1. 实验中生成的bootloader就像一个小型的操作系统，支持很多命令，部分实现了TCP/IP协议
2. 在宿主机上搭建tftp Server，然后利用bootloader中的tftp客户端下载Linux内核镜像与根文件系统镜像。

宿主机到开发板的远程登录方式

1. 串口（minicom、超级终端）
2. ssh（linux ssh客户端、putty），被登录一端必须运行sshd软件。

宿主机到开发板的文件下载方式

1. 利用U盘、SD卡等移动存储设备
2. 利用minicom软件和串口
3. 用tftp和网口
4. 用scp/WinScp工具和网口（底层先要跑通ssh协议）
5. 用网络文件系统NFS方式，将开发板的文件系统映射到宿主机文件系统的某个子目录下。
6.

可以更简单

- ◆ 开发板的厂家会提供针对开发板经过测试的SDK
 - Toolchain
 - 移植的内核（内核源代码）
 - 根文件系统（如何生成）
 - Demo
- ◆ 有些事情必须做

主机交叉开发环境的配置

►配置控制台程序

Windows操作系统中有超级终端 (HyperTerminal) 工具 /SecureCRT; Linux/Unix操作系统有minicom (使用 “minicom”命令启动该软件) 等工具

超级终端配置



主机交叉开发环境的配置

►配置TFTP服务

tftp是一个传输文件的简单协议，它基于UDP协议而实现。此协议设计的时候是进行小文件传输的。

Bootloader作为tftp客户端，从宿主机上下载内核，根文件系统

主机交叉开发环境的配置

► Linux下TFTP服务配置

tftp是一个传输文件的简单协议，它基于UDP协议而实现。此协议设计的时候是进行小文件传输的

vim /etc/xinetd.d/tftp

```
service tftp
{
    socket_type      = dgram
    protocol         = udp
    wait             = yes
    user             = root
    server           = /usr/sbin/in.tftpd
    server_args      = -s /tftpboot
    disable          = no
    per_source       = 11
    cps              = 100 2
    flags            = IPv4
}
```

25

西安电子科技大学

主机交叉开发环境的配置

► 启动TFTP服务

\$ /etc/init.d/xinetd start

► 关闭TFTP服务

\$ /etc/init.d/xinetd stop

► 重启TFTP服务

\$ /etc/init.d/xinetd restart

► 查看TFTP状态

\$ netstat -au | grep tftp

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
udp	0	0	*:tftp		*:*

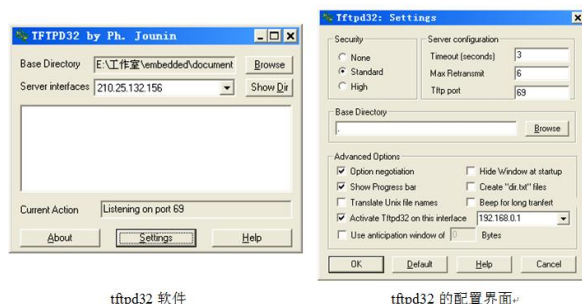
26

西安电子科技大学

主机交叉开发环境的配置

• Windows下TFTP服务配置

在Windows下配置tftp服务需要安装使用tftp服务器软件，常见的可使用tftpd32，网上有很多下载该软件的地方，读者可以自行下载。要注意的是，该软件是tftp的服务器端，而目标板上则是tftp的客户端



tftpd32 软件

tftpd32 的配置界面

27

西安电子科技大学

主机交叉开发环境的配置

► NFS文件系统

NFS为Network FileSystem的简称，最早是由Sun公司提出发展起来的，其目的就是让不同的机器、不同的操作系统之间可以彼此共享文件。

- NFS的使用分为服务器端和客户端，其中服务器端提供要共享的文件，而客户端则通过挂载“mount”这一动作来实现对共享文件的访问操作。在嵌入式开发中，通常NFS服务端在宿主机上运行，而客户端在目标板上运行。可以直接访问到远程目录中的文件。
- NFS服务器端是通过读入它的配置文件“/etc/exports”来决定所共享的文件目录的。

28

西安电子科技大学

主机交叉开发环境的配置

► NFS配置

配置文件: /etc/exports

配置文件每一行格式:

[共享的目录][客户端主机名称或IP]([参数1, 参数2...])

NFS配置文件常用参数:

选 项	参 数 含 义
Rw	可读写的权限
Ro	只读的权限
no_root_squash	NFS 客户端分享目录使用者的权限, 即如果客户端使用的是 root 用户, 那么对于这个共享的目录而言, 该客户端就具有 root 的权限
Sync	资料同步写入到内存与硬盘当中
Async	资料会先暂存于内存当中, 而非直接写入硬盘

NFS配置文件举例:

cat /etc/exports

/home/david/project *(rw, sync, no_root_squash)

主机交叉开发环境的配置

► NFS服务启动

设置NFS服务在每次系统引导时自动开启:

/sbin/chkconfig nfs on