

嵌入开发环境搭建与系统移植

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）

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

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

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；

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）

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

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

主机交叉开发环境的配置

配置控制台程序

配置TFTP服务

Linux下TFTP服务配置

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	*:*	

NFS文件系统

Network FileSystem的简称，也即网络文件系统。

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。