

## 第3章 嵌入式 Linux 驱动程序 开发环境的建立

嵌入式 Linux 驱动程序的开发需要依赖相应的开发环境,包括交叉编译环境及文件共享等。本章首先讲述交叉编译环境的建立方法,包括宿主机 Linux 操作系统、目标机 Linux 操作系统以及交叉编译工具的安装,其次讲述交叉编译工具所需超级终端及文件共享的配置方法。

### 3.1 交叉编译环境的建立

使用高级语言编写的代码都需要通过编译的方式转换成计算机可以识别和执行的二进制代码。通常是在装有开发工具的 PC 上开发能在该计算机上运行的程序,这种编译方式因此称为 native compilation,中文意思为本机编译。然而,这种方式通常不适合于嵌入式系统的软件开发,在进行嵌入式系统开发时,运行程序的目标平台通常只具有有限的存储空间和运算能力,比如常见的 ARM 平台,其一般的静态存储空间大概是 16~32MB,而 CPU 的主频大概在 100~500MHz 之间。这种情况下,在 ARM 平台上进行本机编译就不太可能了,这是因为一般的编译工具链需要很大的存储空间,并需要很强的 CPU 运算能力。为了解决这个问题,嵌入式系统的软件开发通常采用一种交叉编译调试的方式,交叉编译调试环境建立在一台被称为宿主机的 PC 上,对应的开发板被称为目标板。

宿主机开发时使用宿主机上的交叉编译、汇编及链接工具形成可执行的二进制代码,这种可执行代码并不能在宿主机上执行,而只能下载到目标板上运行。进行嵌入式开发的首要工作是安装配置一台装有指定操作系统的 PC 作宿主机,对于嵌入式 Linux 驱动程序开发,宿主机上的操作系统一般要求也是 Linux,安装后宿主机的 Linux 系统后,再在宿主机上建立交叉编译调试的开发环境。

安装宿主机 Linux 操作系统有几个可选方案:①直接安装 Linux 操作系统;②在 Windows 操作系统下安装 CYGWIN 模拟 Linux 操作系统;③在 Windows 操作系统下安装虚拟机,然后再在虚拟机中安装 Linux 操作系统。鉴于大多数读者通常使用的是 Windows 操作系统,为了方便 Windows 应用程序的继续使用,本书推荐采用后两种方案,当然也可以采用直接安装 Linux 的办法来进行以后的实验。

开发嵌入式 Linux 设备驱动程序,目标机当然要烧写 Linux 内核,本书第 5~10 章的实例使用 Linux 2.4 内核,如需移植到 Linux 2.6 运行,则需做适当地修改。由于目前广泛使用的 ARM 9 实验开发平台都烧写了 Linux 内核,其烧写方法就不再讲述,确实需要该方法的读者可参考所用实验开发平台的实验指导书。

针对 ARM 目标平台的嵌入式 Linux 系统开发需要建立一个 ARM-Linux 交叉编译环境,该交叉编译环境由 gcc 编译器、binutils 工具链和 libc 库等组成,其中 gcc 和 binutils 又包含多个工具,其作用如表 3-1 所示。

表 3-1 ARM-Linux 交叉编译环境常用工具

名 称	类 别	作 用
arm-linux-gcc	gcc	编译.c 或.S 开头的 C 程序或汇编程序
armlinuxg++	gcc	编译 C++ 程序
armlinuxas	binutils	ARM 汇编程序
armlinuxld	binutils	连接器(Linker)，把多个.o 或库文件连接成一个可执行文件

交叉编译工具的准备可以有两种方法,一是从有关网站下载通用的软件包,然后经过一系列的编译过程产生所需的交叉编译工具;二是直接从 ARM 官方网站下载已经移植好的交叉编译工具。前者编译过程较复杂,本书采用第二种较为简单的方法。

(1) 下载软件包。

ftp://ftp.arm.linux.org.uk/pub/armlinux/toolchain 提供了 cross3.0.tar.bz2 等交叉编译工具的下载,在此就选择下载 cross3.0.tar.bz2,其中包含了表 3-1 所介绍的交叉编译工具和 libc 库。

(2) 假设下载后保存在/root 中,按以下步骤解压软件包。

```
# cd /root
# tar -jxvf cross3.0.tar.bz2
```

(3) 在/usr/local 建立 arm 目录。

```
# cd /usr/local
# mkdir arm
```

(4) 将解压后产生的 3.0 目录复制到/usr/local/arm 目录下。

```
# cd arm
# cp -dr /root/3.0 .
```

(5) 设置环境变量。

① 修改/etc/profile,在最后加上:

```
export PATH= $PATH:/usr/local/arm/3.0/bin
```

② 重新登录或运行下面的命令让新设置的环境变量起作用。

```
# source /etc/profile
```

在任意路径下输入“arm”再按 Tab 键,如果能列出表 3-1 所示的交叉编译工具,则说明 arm 交叉编译工具已安装好了。

### 3.2 超级终端的配置

Linux 实验开发平台中的 Linux 操作系统一般是通过宿主机 Windows 下的超级终端或 Linux 下的 minicom 来操作的。如果宿主机的 Linux 操作系统是在虚拟机中安装的,则可以使用超级终端;如果宿主机的 Linux 操作系统是直接安装的,则需要使用 minicom。

超级终端与 minicom 的配置方法大体相同,在此只讲述超级中断的配置方法。

### 1. 新建超级终端

按“开始”→“程序”→“附件”→“通信”的顺序运行 Windows 系统下的超级终端(HyperTerminal)应用程序,新建一个超级终端。如果要求输入区号、电话号码等信息请随意输入,出现如图 3-1 所示的“连接描述”对话框时,为所建超级终端取一个名字,然后单击“确定”按钮。



图 3-1 连接描述对话框

### 2. 选择串行口

在图 3-2 所示的串行口选择对话框中选择嵌入式 Linux 实验开发平台实际连接的 PC 串行口(如 COM1),然后单击“确定”按钮。

### 3. 设置串行口

在图 3-3 所示的串行口属性对话框中,设置通信的格式。波特率选“115200”,数据位选“8”,奇偶校验选“无”,停止位选“1”,数据流控制选“无”。单击“确定”按钮完成设置。

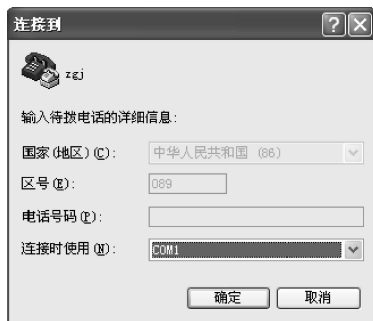


图 3-2 串行口选择对话框



图 3-3 串行口属性对话框

完成新建超级终端的设置以后,选择超级终端文件菜单中的“保存”子菜单,将当前设置保存,以备后用。

## 3.3 文件共享的配置

开发时用到宿主机 Windows 系统、宿主机 Linux 系统及目标机 Linux 系统,经常需要在不同系统之间共享文件。宿主机 Windows 系统与宿主机 Linux 系统之间的文件共享需要通过配置 samba 来实现,宿主机 Linux 系统与目标机 Linux 系统之间的文件共享需要通过配置 NFS(网络文件系统)来实现。

### 3.3.1 配置防火墙

利用 samba 使宿主机中的 Windows 系统与 Linux 系统共享需要配置 Windows 防火墙,利用 NFS 使宿主机 Linux 系统及目标机 Linux 系统之间的文件共享需要配置 Linux 防火墙。

#### 1. Windows 防火墙配置

首先配置 Windows 防火墙。按“开始”→“控制面板”→“安全中心”顺序打开 Window 防火墙配置窗口,如图 3-4 所示。

如果防火墙已关闭,单击“取消”按钮退出配置,如果已启用,单击“例外”选项卡弹出图 3-5 所示 Windows 防火墙“例外”窗口。

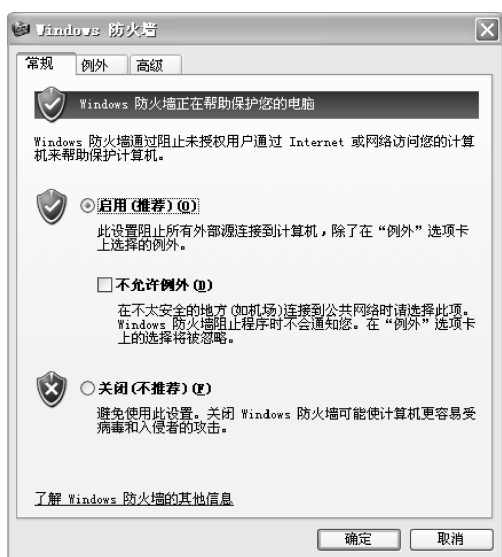


图 3-4 Windows 防火墙“常规”窗口

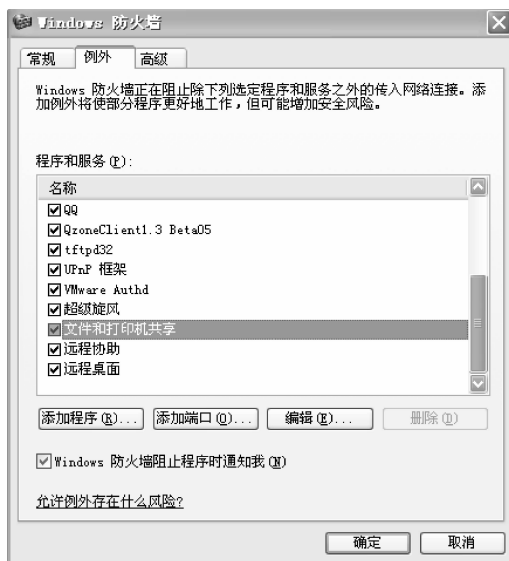


图 3-5 Windows 防火墙“例外”窗口

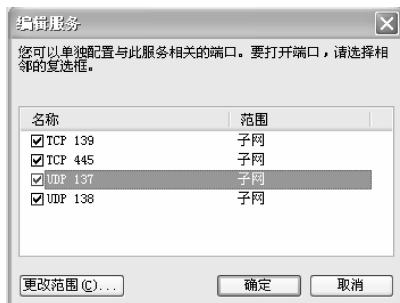


图 3-6 Windows 防火墙“编辑服务”窗口

在 Windows 防火墙“例外”窗口中选择“文件和打印机共享”,再单击“编辑”按钮选择防火墙允许的端口,如图 3-6 所示。

由于 samba 服务器使用 TCP 139 和 TCP 445 端口,因此选择这两个端口,选后单击“确定”按钮返回,在图 3-5 和图 3-4 中也单击“确定”按钮返回,至此已具有 samba 服务器所需要的防火墙配置。

#### 2. 配置 Linux 防火墙

对于 REDHAT 9.0,它默认的是打开了防火墙,因此对于外来的 IP 访问它全部拒绝,这样其他网络设备根本无法访问它,即无法用 NFS mount 它,许多网络功能都将无法使用。因此网络安装完毕后,应立即关闭防火墙。

### 3.3.2 配置 samba 使 Windows 与 Linux 共享

samba 缩写为 SMB,是一组软件包,用于 Linux 之间或 Linux 与 Windows 之间的文件共享。

(1) 安装 samba 服务器软件。

samba 服务器软件包由 system-config-samba、samba-common、samba-client 和 samba 几部分组成,如果在安装 Linux 时已经安装了 samba,就不需要再次安装了。但如果不确定是否已经安装了 samba,可使用下面的命令来确认。

```
# rpm -qa | grep samba
```

如果确认没有安装 samba,可将 Red Hat Linux 9 的第一张安装光盘放入光盘驱动器中,然后通过下面的命令进行安装。

```
# mount /mnt/cdrom
```

将终端的当前工作目录切换到 Red Hat Linux 9 的安装光盘的 Red Hat/RPMS 目录,使用如下命令。

```
# cd /mnt/cdrom/Red Hat/RPMS
```

安装所需要 RPM 包,使用如下命令。

```
# rpm -ivh samba-common-2.2.7a-7.9.0.i386.rpm
# rpm -ivh samba-2.2.7a-7.9.0.i386.rpm
# rpm -ivh samba-client-2.2.7a-7.9.0.i386.rpm
```

安装 samba 的图形配置工具,使用如下命令。

```
# rpm -ivh redhat-config-samba-1.0.4-1.noarch.rpm
```

(2) 在 Linux 系统中查看 Windows 系统中的共享文件。

在 Linux 系统中可使用 samba 提供的 samba 客户程序 smbclient 访问 Windows 提供的共享文件夹,命令格式如下。

```
smbclient //IP 地址/共享文件夹
```

共享文件夹是在 Windows 中设置为共享的文件夹。

举例:

```
[root@localhost ~]# smbclient //202.202.159.139/2410-SV4.1
Password: 直接按 Enter 键
Domain=[ZGJ] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \>
```

登录到 samba 服务器后,可以用 smbclient 的一些指令,像用 FTP 指令一样上传和下载文件,smbclient 指令见表 3-2。

(3) 在 Windows 系统中访问 Linux 系统中的 samba 服务器共享文件夹。

按以下步骤从 Windows 系统中访问 Linux 系统中的 samba 服务器共享文件夹。

表 3-2 smbclient 命令

命 令	功 能
? 或 help [command]	提供帮助或某个命令的帮助
! [shell command]	执行 SHELL 命令
cd [目录]	切换到服务器端的指定目录
lcd [目录]	切换到客户端指定的目录
dir 或 ls	列出当前目录下的文件
exit 或 quit	退出 smbclient
get file1 [file2]	从服务器下载 file1,以文件名 file2 存在本地机上
mget file1 file2 file3	从服务器上下载多个文件
md 或 mkdir 目录	在服务器上创建目录
Rd 或 rmdir 目录	删除服务器上的目录
put file1 [file2]	向服务器上传 file1, 以文件名 file2 存在服务器上
mput file1 file2 filen	向服务器上传多个文件

第一步,通过 smbpasswd 命令添加 samba 用户并设置密码。

用法: smbpasswd -a username

username 是 Linux 系统的用户名,现在被配置为 samba 用户,但其 Linux 系统用户密码和 samba 用户的密码可以不相同。

举例: 添加 samba 用户 zgj。

```
[root@localhost samba] # smbpasswd -a zgj
New SMB password: 输入 samba 用户 zgj 的密码
Retype new SMB password: 再输入一次 samba 用户 zgj 的密码
```

第二步,配置 linux 系统中的 samba 服务器共享文件夹。

修改/etc/samba/smb.conf 文件,将 global 部分修改如下。

```
[global]
    workgroup=mygroup
    server string=samba Server
```

其中,mygroup 为 Windows 系统的工作组名称,根据实际情况改写。

在该文件的 Share Definitions 部分增加 samba 服务器共享文件,格式如下。

```
[sharename]
    path=sharepath
    writeable=yes
    browseable=yes
    guest ok=yes
```

其中,sharename 是共享文件名称,sharepath 共享文件路径,writeable 表示是否可写,browseable 表示是否可以查看。

举例: 配置共享文件夹 root。

```
[root]
    path=/root
    writeable=yes
```

```
browseable=yes  
guest ok=yes
```

第三步,关闭防火墙。

如果 Linux 系统中有防火墙,一定要把 TCP 139 和 TCP 445 这两个端口打开,如果不知道怎么打开,就用如下命令把防火墙规则清掉。

```
[root@localhost ~]# iptables -F
```

第四步,启动 smbd 和 nmbd 服务器。

```
[root@localhost ~]# service smb start
```

启动 SMB 服务: [ 确定 ]

启动 NMB 服务: [ 确定 ]

第五步,在 Windows 系统中访问 Linux 系统的 samba 服务器共享文件夹。

首先配置 Windows 系统的 IP 地址与 Linux 系统的 IP 地址在同一网段,然后使用以下方法访问 Linux 系统的 samba 服务器共享文件夹。

方法: 打开“我的电脑”,在地址栏输入“\\IP 地址”

该 IP 地址是要访问的 samba 服务器所在 Linux 系统的 IP 地址。

举例: 在 Windows 系统中访问 IP 地址为 202.202.159.134 的 Linux 系统中 samba 服务器共享文件夹。

在浏览器地址栏输入“\\202.202.159.134”,然后按 Enter 键。

### 3.3.3 配置 NFS 使宿主机 Linux 与目标机 Linux 共享

NFS 即网络文件系统,是 FreeBSD 支持的文件系统中的一种,它允许一个系统在网络上与他人共享目录和文件。通过使用 NFS,用户和程序可以像访问本地文件一样访问远端系统上的文件。

#### 1. NFS 服务器的配置

etc/exports 文件指定了哪个文件系统应该输出,该文件每行指定一个输出的文件系统,哪些机器可以访问该文件系统以及访问权限。

格式: [共享目录] [主机名 1 或 IP1(参数 1,参数 2)] [主机名 2 或 IP2(参数 3,参数 4)]

共享目录是 NFS 客户端可以访问的目录,主机名或 IP 地址是要访问共享目录的主机名或 IP 地址,当主机名或 IP 地址为空时,则代表共享给任意客户机提供服务。参数是可选的,当不指定参数时,NFS 将使用默认选项。默认的共享选项是 sync, ro, root\_squash, no\_delay。

下面是一些 NFS 共享的常用参数。

ro: 只读访问。

rw: 读写访问。

sync: 所有数据在请求时写入共享。

async: NFS 在写入数据前可以响应请求。

secure: NFS 通过 1024 以下的安全 TCP/IP 端口发送。

insecure: NFS 通过 1024 以上的端口发送。

wdelay: 如果多个用户要写入 NFS 目录,则归组写入(默认)。

no\_wdelay: 如果多个用户要写入 NFS 目录,则立即写入,当使用 async 时,无需此设置。

hide: 在 NFS 共享目录中不共享其子目录。

no\_hide: 共享 NFS 目录的子目录。

subtree\_check: 如果共享/usr/bin 之类的子目录时,强制 NFS 检查父目录的权限(默认)。

no\_subtree\_check: 和上面相对,不检查父目录权限。

all\_squash: 共享文件的 UID 和 GID 映射匿名用户 anonymous,适合公用目录。

no\_all\_squash: 保留共享文件的 UID 和 GID(默认)。

root\_squash: root 用户的所有请求映射成如 anonymous 用户一样的权限(默认)。

no\_root\_squas: root 用户具有根目录的完全管理访问权限。

anonuid=xxx: 指定 NFS 服务器/etc/passwd 文件中匿名用户的 UID。

anongid=xxx: 指定 NFS 服务器/etc/passwd 文件中匿名用户的 GID。

举例: 先创建共享目录/home/share,在该目录中创建文件 zgjtest,赋予 202.202.159.\* 主机读写权限,其他机器仅有只读权限。

```
[root@localhost ~]# mkdir /home/share
```

```
[root@localhost ~]# emacs /etc/exports
```

在 exports 中输入以下内容并保存。

```
/home/share 202.202.159.* (rw, sync) *(ro)
```

**注意:** 配置后需重新启动 NFS。

## 2. NFS 服务的启动和停止

要把自己的文件系统共享出去,首先保证自己的机器上启动了 NFS 服务。启动 NFS 服务就是需要启动一组程序,包括装配服务器和 NFS 协议服务器。

可使用以下命令来查看 NFS 守护进程的当前状态。

```
# service nfs status
rpc.mountd 已停
nfsd 已停
rpc.rquotad 已停
```

可以使用以下命令来启动 NFS 守护进程。

```
# service nfs start
```

启动 NFS 服务:	[确定]
关掉 NFS 配额:	[确定]
启动 NFS 守护进程:	[确定]
启动 NFS mountd:	[确定]

可使用以下命令来停止 NFS 守护进程。

```
# service nfs stop
```

如果修改了 NFS 的配置,可使用以下命令来重新启动 NFS 的守护进程。



```
# service nfs restart
```

### 3. NFS 服务器的状态查询

showmount 命令用来显示远程主机上 NFS 服务器的状态。

用法：# showmount [参数][NFS 服务器主机地址]

缺省参数时显示正连接到指定主机上 NFS 服务器的客户端，主要参数选项如下。

-e：显示指定主机上 NFS 服务器的所有输出目录。

-d：显示指定主机上 NFS 服务器已被客户端连接的所有输出目录。

-a：显示指定 NFS 服务器的所有客户端主机及其所连接的目录。

举例：显示本机上的 NFS 服务器信息。

```
[root@localhost ~]# showmount
Hosts on localhost.localdomain:
202.202.159.139
[root@localhost ~]# showmount -e
Export list for localhost.localdomain:
/home/share (everyone)
[root@localhost ~]#
```

### 4. 在 NFS 的客户端访问 NFS 服务器共享目录

通过查询 NFS 服务器状态确认 NFS 服务器设置正确后，在客户端主机使用 mount 命令来挂载 NFS 服务器的共享目录到本地目录。

用法：mount NFS 服务器地址:共享目录 本地挂载点目录

在 NFS 服务器地址与共享目录之间用“:”隔开，挂载点为本地空目录。

举例：挂载 202.202.159.139 计算机的/home/share 目录到本机的/mnt/nfs 目录，并进入该目录。

```
[root@localhost ~]# mount 202.202.159.139:/home/share/ /mnt/nfs
[root@localhost ~]# cd /mnt/nfs
[root@localhost share]# ls
zgjtest
```

当用户不需要再使用某个 NFS 服务器的共享目录时，可使用 unmount 命令来卸载目录的共享。

举例：卸载上面挂载的/mnt/nfs 目录。

```
[root@localhost ~]# umount /mnt/nfs
```

**注意：**如果以上目录没有，则先要建立。

## 习 题 3

1. 为什么要安装交叉编译工具？
2. 配置 NFS 的目的是什么？
3. 按书中所述方法建立嵌入式 Linux 驱动程序开发环境。

## 第 4 章 Linux 设备驱动内核基础

本章对开发 Linux 设备驱动程序所需要的内核基础知识进行简单介绍,包括 Linux 设备表示、I/O 端口访问机制、中断管理、并发控制等。

### 4.1 Linux 设备的表示

Linux 内核包括进程管理、存储管理、文件管理、设备驱动、网络协议栈、系统调用接口等功能模块,因此设备驱动程序是 Linux 内核的一部分。

在 Linux 的设备管理中,除了设备类型(字符设备、块设备和网络设备)以外,内核还需要一对称作主、次设备号的参数,才能唯一标识设备。主设备号(major number)是驱动程序的标识,不同的设备可以使用相同的驱动程序,但它们这时具有相同的主设备号。次设备号(minor number)用来区分使用相同驱动程序的不同设备,如使用相同驱动程序的不同硬盘。内核在与设备驱动程序通信时,使用设备类型、主设备号和次设备号来标识一个具体的设备。

从用户角度而言,由于用户通常不会记住每台设备的主、次设备号,而且,在使用不同设备时,用户并不希望使用不同的操作方法,而是希望能用同样的方法来访问设备和普通文件。为此,Linux 采用了设备文件的概念。简单地说,系统试图使它对所有各类设备的输入、输出看起来就好像对普通文件的输入、输出一样。Linux 将设备当做文件进行处理,这一类特殊文件就是设备文件,通常在/dev 下面存在一个对应的逻辑设备节点。设备文件可以使用对一般文件进行操作的相关函数进行操作,这样就大大方便了对设备的处理。由于引入设备文件这一概念,Linux 为文件和设备提供了一致的用户接口。对用户来说,设备文件与普通文件并无区别,用户可以打开和关闭设备文件,可以读数据,也可以写数据等。

在引入设备文件系统(devfs)之前,Linux 习惯上将设备文件放在目录/dev 或其子目录之下。设备文件名通常由两部分组成,第一部分通常较短,可能只有 2 个或 3 个字母组成,用来表示设备大类。例如,普通硬盘如 IDE 接口的为“hd”,SCSI 硬盘为“sd”,软盘为“fd”。第二部分通常为数字或字母用来区别设备实例。例如,/dev/hda、/dev/hdb、/dev/hdc 表示第 1~3 块硬盘;而/dev/hda1、/dev/hda2、/dev/hda3 则表示第一硬盘的第 1~3 分区。然而自从 Linux 内核中正式引入了设备文件系统 devfs 之后,所有的设备文件作为一个可以挂接的文件系统被纳入了文件系统的管理范围之内,设备文件可以挂接到任何需要的地方,同时命名规则也有了更改,目录管理更加具有条理性。

### 4.2 设备文件系统(devfs)

从 Linux 内核的测试版本 2.3.46 开始,devfs 被正式地引入到官方的内核版本,从此设备的管理有了新的模式,不必再受主、次设备号的数目的限制。devfs 仅仅是一个虚拟的文