第3章

嵌入式 Linux 操作系统

本章的知识只需在 PC 上就可完成。学习本章内容后读者将掌握如下知识。

- Linux 基本概念。
- Linux 的目录结构。
- Linux 的常用命令。
- Linux 的文本编辑器。
- Linux 系统的启动过程。

视频讲解

3.1 Linux 基本概念

Linux 的出现，最早开始于一位名叫 Linus Torvalds 的计算机业余爱好者，当时他是芬兰赫尔辛基大学的学生。他的目的是想设计一个代替 Minix（由一位名叫 Andrew Tannebaum 的计算机教授编写的一个操作系统示教程序）的操作系统，这个操作系统可用于 386、486 或奔腾处理器的个人计算机上，并且具有 Unix 操作系统的全部功能，因而开始了 Linux 雏形的设计。

嵌入式系统其发展已有 20 多年的历史，国际上也出现了一些著名的嵌入式操作系统，如 VxWorks，Palm OS，Windows CE 等，但这些操作系统均属于商品化产品，价格昂贵且由于源代码不公开导致了诸如对设备的支持、应用软件的移植等一系列的问题。而 Linux 作为一种优秀的自由软件，近几年在嵌入式领域异军突起，成为有潜力的嵌入式操作系统。

从应用上讲，Linux 有 4 个主要部分：内核、Shell、文件系统和实用工具。

1. Linux 内核

Linux 内核是整个 Linux 系统的灵魂，Linux 系统的能力完全受内核能力的制约。Linux 内核负责整个系统的内存管理、进程调度和文件管理。Linux 内核的容量并不大，一般一个功能比较全面的内核也不会超过 1MB，而且大小可以裁减，这个特性对于嵌入式是非常有好处的。合理地配置 Linux 内核是嵌入式开发中很重要的环节，对内核的充分了解是嵌入式 Linux 开发的基本功。

下面简单介绍 Linux 内核功能，Linux 内核的功能大致有如下几个部分。

1）进程管理

进程管理功能负责创建和撤销进程，以及处理它们和外部世界的连接。不同进程之间的通信是整个系统的基本功能，因此也由内核处理。除此之外，控制进程如何共享 CPU 资源的调度程序也是进程管理的一部分。概括地说，内核的进程管理活动就是在单个或多个

CPU 上实现多进程的抽象。

2）内存管理

内存是计算机的主要资源之一，用来管理内存的策略是决定系统性能的一个关键因素。内核在有限的可用资源上为每个进程都创建了一个虚拟寻址空间。内核的不同部分在和内存管理子系统交互时使用一套相同的系统调用。

3）文件管理

Linux 在很大程度上依赖于文件系统的概念，Linux 中的每个对象都可以被视为文件。

4）设备控制

几乎每个系统操作最终都会映射到物理设备上。除了处理器、内存以及其他有限的几个实体外，所有的设备控制操作都由与被控制设备相关的代码来完成，这段代码叫作设备驱动程序。内核必须为系统中的每个外设嵌入相应的驱动程序。

5）网络功能

网络功能也必须由操作系统来管理，因为大部分网络操作都和具体的进程无关。在每个进程处理这些数据之前，数据报必须已经被收集、标识和分发。系统负责在应用程序和网络之间传递数据。另外，所有的路由和地址解析问题都由内核进行处理。

2. Linux Shell

Shell 是 Linux 系统下的命令解释器，它提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行，类似于 Microsoft Windows 的 Command 命令。

Linux 内核与界面是分离的，它可以脱离图形界面单独运行，同样也可以在内核的基础上运行图形化的桌面。在 Linux 的图形用户界面(GUI)下，终端窗口就是 Shell。

每个 Linux 系统的用户可以拥有自己的用户界面或 Shell，用以满足自身专门的 Shell 需要。

3. Linux 文件系统

Linux 的文件系统和 Microsoft Windows 的文件系统有很大的不同。Linux 只有一个文件树，整个文件系统是以一个树根/为起点的，所有的文件和外部设备都以文件的形式挂接在这个文件树上，包括硬盘、软盘、光驱、调制解调器等，这和以“驱动器盘符”为基础的 Microsoft Windows 系统有很大区别。

Linux 的文件系统如图 3.1 所示。

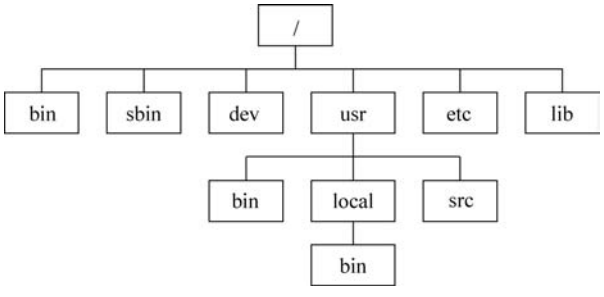

图 3.1　Linux 文件系统的目录结构

下面对各主要目录做一个简要的介绍。

1) /bin 和/sbin

这两个目录通常存放 Linux 基本操作命令的执行文件,其中的内容是一样的,二者的主要区别是:/sbin 中的程序只能由 root(系统管理员)来执行。

2) /dev

这是一个非常重要的目录,它存放着各种外部设备的镜像文件。在 Linux 中,所有的设备都当作文件进行操作。例如,第一个硬盘的名字是 hda,硬盘中的第一个分区是 hda1,第二个分区是 hda2,第一个光盘驱动器的名字是 hdc。用户可以非常方便地像访问文件一样对外部设备进行访问。

3) /lib

该目录用来存放系统动态链接共享库。Linux 系统内核内置的已经编译好的驱动程序存放在/lib/modules/kernel 目录下。几乎所有的应用程序都会用到这个目录下的共享库。因此,不要轻易对这个目录进行操作。

4) /usr

该目录用来存放用户应用程序和文件,类似于 Windows 下的 ProgramFiles 目录。Linux 系统内核的源码存放在 usr/src/kernels 目录下。

5) /etc

该目录用来存放系统的各种配置文件,系统在启动过程中需要读取其参数进行相应的配置。其中:

- /etc/rc.d 目录存放启动或改变运行级时运行的脚本文件及目录。
- /etc/passwd 文件为用户数据库,其中的字段给出了用户名、真实姓名、起始目录、加密的口令和用户的其他信息。
- /etc/profile 文件为系统环境变量的配置文件。

内核、Shell 及文件系统一起形成了基本的操作系统结构。它们使得用户可以运行程序、管理文件以及使用系统。此外,Linux 操作系统还有许多被称为实用工具的程序,能辅助用户完成一些特定的任务。

3.2 Linux 常用操作命令

通常,在完成 Linux 安装后,就可以进入与 Windows 类似的图形化窗口界面。这个界面是 Linux 图形化界面 X 窗口系统的一部分。要注意的是,X 窗口系统仅仅是 Linux 的一个应用软件(或称为服务),它不是 Linux 自身的一部分。为了让 Linux 系统能高效、稳定地工作,建议读者尽可能地使用 Linux 的命令行界面,也就是在 Shell 环境下工作。

Linux 中运行 Shell 的环境是图形用户界面下的“终端”窗口,读者可以单击“终端”启动 Shell 环境。

Linux 中的命令非常多,本节只讲解最常用的一些操作命令。要了解更多的命令使用方法,请查阅相关资料和书籍。

3.2.1　文件目录相关命令


视频讲解

Linux 中有关文件目录的操作是最常用的，与文件目录相关的命令如表 3.1 所示。

表 3.1　文件目录相关命令

命　令	命 令 含 义	程序所在目录
ls	显示文件名(相当于 DOS 的 dir 命令)	/bin
cd	切换目录(相当于 DOS 的 cd 命令)	Shell 内部提供
cp	复制文件(相当于 DOS 的 copy 命令)	/bin
mkdir	创建新目录(相当于 DOS 的 md 命令)	/bin
rm	删除文件(相当于 DOS 的 del 命令)	/bin
rmdir	删除空目录(相当于 DOS 的 rd 命令)	/bin
mv	移动文件，另兼有更换文件名的作用	/bin
pwd	显示目前所在目录	/bin
cat	显示文本文件内容	/bin
env	查看环境设置	/usr/bin
find	查找文件	/usr/bin
grep	寻找某字串内容	/bin
more	分屏显示文本文件内容或输出结果	/bin
mtools	与 MS-DOS 兼容的操作命令集	/usr/bin
su	用于切换用户	/bin
df	查看磁盘使用情况	/bin
uname	查看当前 Linux 版本信息，使用时要带参数-r	/bin

1. ls 命令

1) 作用

ls 的功能为列出目录的内容。该命令类似于 DOS 下的 dir 命令。

2) 命令格式

```
ls [-选项] [目录或文件名]
```

3) 命令选项

- -a：显示指定目录下所有子目录与文件名，包括隐藏文件。
- -l：以长格式来显示文件的详细信息。

4) 示例

查看当前根目录下的文件。

```
[root@localhost  /]# ls
bin    dev    home    lib         misc    opt    root    tftpboot    usr
boot etc   initrd   lost+found   mnt     proc    sbin    tmp       var
```

查看当前 root 目录下的所有文件，包括隐藏文件。

```
[root@localhost  root]#ls  -a
..esd_auth            .gtkrc                  .tcshrc      ebook
...fonts.cache-1      .gtkrc-1.2-gnome2       .viminfo     tmp
```

查看当前 root 目录下的文件属性。

```
[root@localhost  root]#ls  -l
total  72
-rw-r--r--     1  root  root      965    Jan  3  2007    anaconda-ks.cfg
drwx------     4  root  root      4096   Jan  3  2007    evolution
-rw-r--r--     1  root  root      49492  Jan  3  2007    install.log
```

每行显示的信息依次是：文件类型与权限、链接数、文件属主、文件属组、文件大小、建立或最近修改的时间、文件名。

显示的信息中，开头是由 10 个字符构成的字符串，其中第一个字符表示文件类型，它可以是下述类型之一。

- -：普通文件
- d：目录
- l：符号链接
- b：块设备文件
- c：字符设备文件

后面的 9 个字符表示文件的访问权限，分为 3 组，每组 3 位。

第 1 组表示文件属主的权限，第 2 组表示同组用户的权限，第 3 组表示其他用户的权限。每一组的 3 个字符分别表示对文件的读、写和执行权限。

2. 文件权限的表示

用户对文件的读、写和执行权限(简称文件权限)如下所示。

- r：读权限。
- w：写权限。
- x：执行权限，对于目录，表示可进入权限。

文件权限也可用数字表示，其约定如下。

- 数字 0：无权限。
- 数字 1：可执行。
- 数字 2：写权限。
- 数字 4：读权限。

可用数字求和来表示多权限的组合。

例如，用户对某一文件拥有可读、可写、可执行的权限，则可表示为 7(1+2+4=7)，对另一文件拥有可读、可执行的权限，则可表示为 5(1+4=5)。

若对文件拥有可读、可写的权限，则可表示为 6(4+2=6)，若对文件拥有可写、可执行的权限，则可表示为 3(1+2=3)。

总结文件权限的对应关系，如表 3.2 所示。

表 3.2　权限对应关系

字符表示	数字表示	对应权限	字符表示	数字表示	对应权限
-	0	无权限	wx	3	写和执行
x	1	只能执行	rx	5	读和执行
w	2	只写	rw	6	读和写
r	4	只读	rwx	7	读、写和执行

有时，用 3 位数字来表示文件权限，其中每位数字分别表示文件拥有者、同组用户、不同组用户的权限。

例如：

600：表示文件拥有者具有读写权限，其他用户均无任何操作权限。

777：表示文件拥有者、同组用户、不同组用户均具有读、写和执行的权限。

3. cd 命令

1）作用

改变工作目录，该命令与 DOS 下的 cd 命令作用是相同的。

2）命令格式

```
cd  [目录路径/]目录名
```

3）示例

将目录/usr/test 设为当前目录。

```
[root@localhost  root]#cd  /usr/test
[root@localhost  test]#pwd
/usr/test
```

命令 pwd 显示当前目录的绝对路径(从根目录/开始)。

4. mkdir 命令

1）作用

创建一个目录，该命令类似于 DOS 下的 md 命令。

2）命令格式

```
mkdir  [目录路径/新目录名]
```

5. cp 命令

1）作用

复制文件，可以使用通配符，该命令类似于 DOS 下的 copy 命令。

2）命令格式

```
cp  [选项]  [源文件路径]源文件名  目标路径[目标文件名]
```

3）示例

在/tmp 目录下，新建一个子目录 mysub，并将/usr/test 目录下的所有文件复制到 mysub 目录下：

```
[root@localhost  root]#mkdir  /tmp/mysub
[root@localhost  root]#cp  /usr/test/*.*  /tmp/mysub
```

6. rm 命令和 rmdir 命令

1）作用

- rm 为删除指定文件，可以使用通配符，该命令类似于 DOS 下的 del 命令。
- rmdir 为删除指定的目录，该目录必须为空目录。

2）命令格式

- rm [选项] 文件名
- rmdir 目录路径/目录名

3）命令选项

rm 的命令选项如下。

- -i：询问是否删除(y 表示是,n 表示否)。
- -f：不询问是否删除。
- -r：递归删除整个目录,同 rmdir。

4）示例

删除前面示例中在/tmp 目录下建立的子目录 mysub。由于 rmdir 只能删除空目录,因此,要先将该目录下的所有文件删除。

```
[root@localhost root]#rm -f /tmp/mysub/*.*
[root@localhost root]#rmdir /tmp/mysub
```

7. cat 命令

1）作用

cat 为在屏幕上显示文本文件内容的命令。

2）命令格式

```
cat 文件名
```

3）示例

设有一个文本文件 a.txt,应用 cat 显示其内容：

```
[root@localhost abc]#cat a.txt
Hello,I'm writing to this file
```

cat 命令也常用于查看当前 Linux 系统的版本,例如：

```
[root@localhost root]# cat /proc/version
Linux version 2.6.35-22-generic (buildd@rothera) (gcc version 4.4.5 (Ubuntu/
Linaro 4.4.4-14ubuntu4) ) #33-Ubuntu SMP Sun Sep 19 20:34:50 UTC 2010
```

8. pwd 命令

1）作用

pwd 命令用来查看当前工作目录的完整路径。

2）命令格式

```
pwd
```

3）示例

显示当前所在的目录路径。

```
[root@localhost abc]# pwd
/mnt/abc
```

3.2.2　磁盘及系统操作


视频讲解

在 Linux 中与磁盘操作及系统操作相关的命令如表 3.3 所示。

表 3.3　与磁盘及系统操作相关的命令

命　　令	命 令 含 义	程序所在目录
fdisk	硬盘分区及显示分区状态的工具程序	/sbin
df	检查硬盘所剩(所用)空间	/bin
free	查看当前系统内存的使用情况	/usr/bin
mount	挂载某一设备成为某个目录名称	/bin
umount	取消挂载的设备	/bin
du	检查目录所用的空间	/usr/bin
mkbootdisk	制作启动盘	/sbin
shutdown	整个系统关机	/sbin
reboot	重启系统	/sbin
login	用户登录	/bin
logout	用户注销	Shell 内部提供

1. fdisk 命令

1) 作用

fdisk 命令可以用来给磁盘进行分区,查看磁盘情况等,往往使用参数-l 来显示系统的分区情况。

2) 命令格式

```
fdisk  [选项]
```

3) 命令选项

```
-l  显示系统的分区情况
```

4) 示例

```
[root@localhost  root]#fdisk  -l
Disk  /dev/sda: 8589MB,  8589934592  bytes
255 heads,63 sectors/track, 1044  cylinders
Units = cylinders  of 16065 * 512  =  8225280 bytes
  Device  Boot         Start       End      Blocks         Id      System
/dev/sda1   *             1         13      104391         83      Linux
/dev/sda2                14       1004     7960207+      86      Linux
/dev/sda3              1005       1044     321300         82      Linux swap
```

2. df 命令

1) 作用

检查硬盘所剩(所用)空间。

2) 命令格式

```
df  [选项]
```

3）命令选项

- -h：以1024KB=1MB的方式显示磁盘的使用情况。
- -H：以1000Bytes为换算单位的方式显示磁盘的使用情况。

4）示例

```
[root@localhost  root]#df  -h
Filesystem              Size        Used    Avail   Use%     Mounted on
/dev/sda2              7.5G       4.8G     2.4G     67%     /
/dev/sda1              99M        9.3M     85M      10%     /boot
None                   78M         0       78M      0%      /dev/shm
[root@localhost  root]#df  -H
Filesystem               Size      Used    Avail    Use%     Mounted on
/dev/sda2                8.1G      5.1G    2.6G     67%      /
/dev/sda1                104M      9.7M    89M      10%      /boot
None                     82M        0      82M      0%       /dev/shm
```

3. free命令

1）作用

free命令的功能是查看当前系统内存的使用情况，它显示系统中剩余及已用的物理内存和交换内存，以及共享内存和被核心使用的缓冲区。

2）命令格式

```
free  [选项]
```

3）命令选项

- -b：以B(字节)为单位显示。
- -k：以KB为单位显示。
- -m：以MB为单位显示。

4）示例

```
[root@localhost  root]#free  -b
             total  used        free      shared    buffers    cached
Mem       162107392  83656784  78450688    0       9613312    33099776
-/+ buffers/cache:    40943616    121163776
Swap:  329003008       0      329003008
[root@localhost  root]#free  -m
             total       used        free    shared    buffers    cached
Mem            154         79         74        0          9         31
-/+ buffers/cache:  39    115
Swap:      313       0        313
```

4. mount命令

1）作用

挂载某一设备使之成为某个目录名称。

2）命令格式

```
mount [选项] <-t 类型> [-o 挂载选项] <设备> <挂载点>
```

3）命令选项

- -t：该参数配合选项用于指定一个文件系统分区的类型。
- -o：该参数配合选项用于指定一个或多个挂载选项。

其具体可供选择的内容见表 3.4 所示。

表 3.4　mount 命令的参数选项

命 令 参 数	对 应 选 项	选 项 说 明
-t	vfat	挂载 Windows 95/98 的 FAT32 文件系统
	ntfs	挂载 Windows NT/2000 的文件系统
	hpfs	挂载 OS/2 用的文件系统
	ext2、ext3、nfs	挂载 Linux 用的文件系统
	iso9660	挂载 CD-ROM 光盘
-o	ro，rw	挂载区为只读(ro)或读写(rw)
	async，sync	挂载区为同步写入(sync)或异步写入(async)
	auto，noauto	允许此挂载区被 mount -a 自动挂载(auto)
	dev，nodev	是否允许在此挂载区上建立档案，dev 为可以
	exec，noexec	是否允许此挂载区上拥有可执行的二进制文件
	user，nouser	是否允许此挂载区让普通用户拥有 mount 的权限
	defaults	默认值为 rw，suid，dev，exec，auto，nouser，async
	remount	重新挂载

4）命令使用说明

挂载设备之前，首先要确定设备的类型和设备名称，确定设备名称可通过使用命令 fdisk-l 查看。

要卸载已经挂载的设备，使用 umount 命令。

umount 命令是 mount 命令的逆操作，umount 命令的作用是卸载一个文件系统。例如，将光驱装载到 /mnt/cdrom 目录后，若要取出光盘，必须先使用 umount 命令进行卸载，否则无法取下。它的参数使用方法和 mount 命令是一样的，命令格式如下。

```
umount  <挂载点|设备>
```

5）示例

【例 3-1】 挂载一个 Linux 分区，将其挂载到/mnt 目录下(/mnt 称为挂载点)。

```
[root@localhost  root]# mount -t ext3 /dev/hdb1 /mnt
```

【例 3-2】 挂载硬盘的 Windows 分区，将其挂载到/mnt/wind 目录下。

(1) 用 fdisk-l 查看硬盘的 Windows 分区在 linux 下的设备名称。

```
[root@localhost /]# fdisk -l
Disk /dev/hda: 500.1 GB, 500105249280 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

  Device Boot      Start         End      Blocks   Id  System
/dev/hda1               1        5100    40965718+  2d  Unknown
```

```
/dev/hda2          5101        60801    447418282+   f  W95 Ext'd (LBA)
/dev/hda5          5101        24223    153605466    b  W95 FAT32
/dev/hda6         24224        43346    153605466   2d  Unknown
/dev/hda7         43347        60801    140207256   2d  Unknown

Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

  Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        2610    20860402+  8e  Linux LVM
```

(2) 将设备名称为/dev/hda5 的 Windows 分区挂载到/mnt/wind。

```
[root@localhost  root]#mount -t vfat  /dev/hda5  /mnt/wind
```

/mnt/wind 称为挂载设备/dev/hda5 的挂载点。在挂载了硬盘的 Windows 分区之后，可直接访问 Windows 下的磁盘内容。

【例 3-3】 挂载 U 盘。

U 盘的挂载方法同硬盘的挂载方法是一样的，先用 fdisk -l 查看 U 盘的设备名称，U 盘一般是以 sdb 出现的。设 U 盘的名称为/dev/sdb1，则其挂载命令如下。

```
[root@localhost  /]#mount  -t  vfat  /dev/sdb1  /mnt/usb
```

卸载 U 盘的命令如下。

```
[root@localhost  /]#umount /mnt/usb
```


视频讲解

3.2.3 打包压缩相关命令

Linux 常用的压缩及解压缩命令如表 3.5 所示。

表 3.5 Linux 常用的压缩及解压缩命令说明

压缩工具	解压工具	压缩文件扩展名	压缩工具	解压工具	压缩文件扩展名
gzip	gunzip	.gz	compress	uncompress	.Z
zip	unzip	.zip	tar	tar	.tar

1. gzip 命令

1) 作用

对单个文件进行压缩或对压缩文件进行解压缩，压缩文件名后缀为.gz。

2) 命令格式

```
gzip  压缩或解压缩文件名
```

3) 命令选项

- -d：对压缩文件进行解压缩。
- -r：递归方式查找指定目录并压缩其中所有文件或解压缩。

- -v：对每个压缩文件显示文件名和压缩比。
- -num：用数值 num 指定压缩比，num 取值 1～9，其中 1 代表压缩比最低，9 代表压缩比最高，默认值为 6。

4）示例

```
[root@localhost  test]# gzip test.txt
[root@localhost  test]# ls
test.txt.gz
```

2. tar 命令

1）作用

对文件进行打包或解包，打包文件名后缀为.tar。利用 tar 命令，可以把多个文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。注意，打包与压缩是两个不同的概念，打包只是把多个文件组成一个总的文件，不一定被压缩。

2）命令格式

```
tar ［选项］ 目标文件名  源文件列表
```

3）命令选项

- -A 或--catenate：新增文件到已存在的备份文件。
- -c 或--create：建立新的备份文件。
- -f <备份文件>或--file=<备份文件>：指定备份文件。
- -r 或--append：新增文件到已存在的备份文件的结尾部分。
- -t 或--list：列出备份文件的内容。
- -u 或--update：仅置换较备份文件内的文件更新的文件。
- -v 或--verbose：显示指令执行过程。
- -w 或--interactive：遭遇问题时先询问用户。
- -x 或--extract 或--get：从备份文件中还原文件。
- -z 或--gzip 或--ungzip：通过 gzip 指令处理备份文件。

4）示例

将文件包 filetest.tar.gz 解包的命令如下。

```
#tar - zxvffiletest.tar.gz
```

3.2.4　网络相关命令


视频讲解

Linux 有许多与网络相关的命令，下面介绍几个常用的网络操作命令。

1. ifconfig 命令

1）作用

用于查看和配置网络接口的地址和参数，包括 IP 地址、网络掩码和广播地址。它的使用权限是超级用户。

2）命令格式

- 查看网卡配置信息：ifconfig

· 设置网卡：ifconfig eth0 ［主机 IP 地址］

eth0 代表第 1 块网卡，eth1 代表第 2 块网卡，若主机上仅安装了一块网卡，则为 eth0。

3）示例

```
[root@localhost /]# ifconfig
eth0  Link encap: Ethernet HWaddr 00: 11: 11: 11: 23: 5A
      inet addr: 192.168.1.15 Bcast: 192.168.1.255 Mask: 255.255.255.0
      inet6 addr: fe80::208:2ff:fee0:c18a/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets: 26931 errors: 0 dropped: 0 overruns: 0 frame: 0
      TX packets: 3209 errors: 0 dropped: 0 overruns: 0 carrier: 0
      collisions: 0 txqueuelen: 1000
      RX bytes: 6669382 (6.3 MiB) TX bytes: 321302 (313.7 KiB)
      Interrupt: 11

lo    Link encap: Local Loopback
      inet addr: 127.0.0.1 Mask: 255.0.0.0
      inet6 addr: ::1/128 Scope: Host
      UP LOOPBACK RUNNING MTU: 16436 Metric: 1
      RX packets: 1381 errors: 0 dropped: 0 overruns: 0 frame: 0
      TX packets: 1381 errors: 0 dropped: 0 overruns: 0 carrier: 0
      collisions: 0 txqueuelen: 0
      RX bytes: 1354690 (1.2 MiB) TX bytes: 1354690 (1.2 MiB)
```

重新设置网卡的 IP 地址，改设为 192.168.1.100，则使用以下命令。

```
[root@localhost /]# ifconfig eth0 192.168.1.100
```

注意：用 ifconfig 命令配置的网络参数不需重启就可生效，但机器重新启动以后其设置将会失效。

2. ping 命令

1）作用

ping 命令用于检测网络连接情况，从而判断主机联网是否连接正常。

2）命令格式

```
ping [IP 地址]
```

3）示例

```
[root@localhost /]# ping 192.168.1.1
PING 192.168.1.1  (192.1168.1.1)  56(84) bytes of data.
64 bytes from 192.168.1.1 icmp_seq = 0  ttl = 64  time = 2.439 ms
64 bytes from 192.168.1.1 icmp_seq = 1  ttl = 64  time = 1.149 ms
64 bytes from 192.168.1.1 icmp_seq = 2  ttl = 64  time = 1.043 ms
64 bytes from 192.168.1.1 icmp_seq = 3  ttl = 64  time = 1.038 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received. 0 % packet loss, time 7004ms
rtt min/avg/max/mdev = 1.038/1.924/2.439/1.529 ms, pipe 2
```

3.3　Linux 的文本编辑器

视频讲解

3.3.1　Vi 文本编辑器

Vi 是 Linux 系统的第一个全屏幕交互式编辑程序，它从诞生至今一直得到广大用户的青睐，历经数十年仍然是人们主要使用的文本编辑工具，足以见其生命力之强，而强大的生命力是其强大的功能带来的。由于大多数读者在此之前都已经用惯了 Windows 的 Word 等编辑器，因此，在刚刚接触 Vi 时总会或多或少不适应，但只要习惯之后，就能感受到它的方便与快捷。

1. Vi 的模式

Vi 有 3 种模式，分别为命令行模式、插入模式和底行模式。各模式的功能具体进行介绍如下。

1）命令行模式

用户在用 Vi 编辑文件时，最初进入的为一般模式。在该模式中可以通过上下移动光标进行“删除字符”或“整行删除”等操作，也可以进行“复制”“粘贴”等操作，但无法编辑文字。

2）插入模式

只有在该模式下，用户才能进行文字编辑输入，用户可按 Esc 键回到命令行模式。

3）底行模式

在该模式下，光标位于屏幕的底行。用户可以进行文件保存或退出操作，也可以设置编辑环境，如寻找字符串、列出行号等。

2. Vi 的基本流程

（1）进入 Vi，即在命令行下输入 Vi hello（文件名）。此时进入的是命令行模式，光标位于屏幕的上方，如图 3.2 所示。

图 3.2　进入 Vi 命令行模式

（2）在命令行模式下输入 i 进入插入模式，如图 3.3 所示。可以看出，在屏幕底部显示有“插入”字样表示插入模式，在该模式下可以输入文字信息。

（3）在插入模式中，按 Esc 键，则当前模式转入命令行模式，此时在底行行中输入：wq（存盘退出）进入底行模式，如图 3.4 所示。

图 3.3 进入 Vi 插入模式

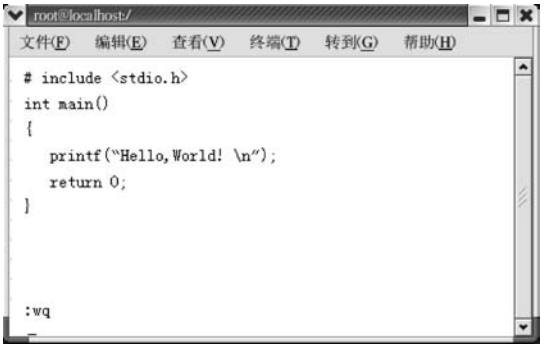

图 3.4 进入 Vi 底行模式

这样,就完成了一个简单的 Vi 操作流程:命令行模式→插入模式→底行模式。由于 Vi 在不同的模式下有不同的操作功能,因此,读者一定要时刻注意屏幕最下方的提示,分清所在的模式。

3. Vi 的各模式功能键

(1) 命令行模式常见功能键如表 3.6 所示。

表 3.6 Vi 命令行模式功能键

功 能 键	功 能 说 明
I	切换到插入模式,此时光标位于开始输入文件处
A	切换到插入模式,并从目前光标所在位置的下一个位置开始输入文字
O	切换到插入模式,且从行首开始插入新的一行
Ctrl+B	屏幕往后翻动一页
Ctrl+F	屏幕往前翻动一页
Ctrl+U	屏幕往后翻动半页
Ctrl+D	屏幕往前翻动半页
0	光标移到本行的开头
G	光标移动到文章的最后
nG	光标移动到第 n 行
$	移动到光标所在行的行尾
n	光标向下移动 n 行

续表

功　能　键	功 能 说 明
/name	在光标之后查找一个名为 name 的字符串
? name	在光标之前查找一个名为 name 的字符串
X	删除光标所在位置的一个字符
dd	删除光标所在行
ndd	从光标所在行开始向下删除 n 行
yy	复制光标所在行
nyy	复制光标所在行开始的向下 n 行
p	将缓冲区内的字符粘贴到光标所在位置(与 yy 搭配)
U	恢复前一个动作

(2) 插入模式的功能键只有一个,也就是 Esc 退出到命令行模式。

(3) 底行模式常见功能键如表 3.7 所示。

表 3.7　Vi 底行模式功能键

功　能　键	功 能 说　明
: w	将编辑的文件保存到磁盘中
: q	退出 Vi(系统对做过修改的文件会给出提示)
: q!	强制退出 Vi(对修改过的文件不作保存)
: wq	存盘后退出
: w [filename]	另存一个名为 filename 的文件
: set nu	显示行号,设定之后,会在每一行的前面显示对应行号
: set nonu	取消行号显示

3.3.2　gedit 文本编辑器


视频讲解

除了 vi 之外,Linux 下还有一个功能同样强大的编辑器 gedit。gedit 是一个 GNOME 桌面环境下兼容 UTF-8 的文本编辑器。它简单易用,有良好的语法高亮,对中文支持很好,支持包括 GB2312、GBK 在内的多种字符编码,是一款自由软件。

gedit 是一个功能强大的文本编辑器,类似于 Windows 系统下面的记事本,它的功能比 Windows 系统的记事本更强大,还具有行号显示、括号匹配、文本自动换行、自动文件备份等功能,适合编写程序代码。

1. gedit 的启动

gedit 的启动方式有多种,可以从菜单启动,也可以从终端命令行启动。从菜单启动时,选择桌面顶部的“应用程序”|“附件”|“文本编辑器”命令即可打开;从终端启动,只需要输入代码 $ gedit 再按 Enter 键即可。

gedit 启动之后的主界面如图 3.5 所示。

2. 窗口说明

读者可以看到 gedit 启动的界面和 Windows 中的“写字板”程序相似。窗口上有菜单栏、工具栏、编辑栏、状态栏等。

图 3.5 gedit 主界面

3. 常用的技巧

1) 打开多个文件

要从命令行打开多个文件,请输入"gedit file1.txt file2.txt file3.txt"命令,然后按下 Enter 键。

2) 将命令的输出输送到文件中

例如,要将 ls 命令的输出输送到一个文本文件中,请输入"ls | gedit",然后按下 Enter 键。ls 命令的输出就会显示在 gedit 窗口的一个新文件中。

3) 更改"突出显示模式"以适用各种文件

例如,更改以适应 html 文件的步骤为,依次选择菜单中的"查看"|"突出显示模式"|"标记语言"|HTML,即可以彩色模式查看 html 文件。

4) 插件

gedit 中有多种插件可以选用,这些插件极大地方便了用户处理代码,常用的包括以下几种。

- 文档统计信息:选择菜单栏中的"工具"|"统计文档"命令,出现"文档统计信息"对话框,里面显示了当前文件中的行数、单词数、字符数及字节数。
- 高亮显示:选择"视图"|"高亮",然后选择需要高亮显示的文本。
- 插入日期/时间:选择"编辑"|"插入时间和日期"命令,则在文件中插入当前时间和日期。
- 跳到指定行:选择"查找"|"进入行"命令,之后输入需要定位的行数,即可跳到指定的行。

5) 常用的快捷键

gedit 常用的快捷键如表 3.8 所示。

表 3.8 gedit 常用的快捷键

快捷键	功能说明	快捷键	功能说明
Ctrl+Z	撤销	Ctrl+Q	退出
Ctrl+C	复制	Ctrl+S	保存
Ctrl+V	粘贴	Ctrl+R	替换
Ctrl+T	缩进		

3.4　Linux 启动过程

了解 Linux 的常见命令之后，下面介绍 Linux 的启动过程。Linux 的启动过程包含了 Linux 工作原理的精髓，在嵌入式系统的开发过程中非常需要这方面的知识积累。

3.4.1　Linux 系统的引导过程

许多人对 Linux 的启动过程感到很神秘，因为所有的启动信息都在屏幕上一闪而过。其实，Linux 的启动过程并不像启动信息所显示的那样复杂，它主要分成以下两个阶段。

(1) 启动内核。在这个阶段，内核装入内存并在初始化每个设备驱动器时打印信息。

(2) 执行程序 init。装入内核并初始化设备后，运行 init 程序。init 程序处理所有程序的启动，包括重要系统精灵程序和其他指定在启动时装入的软件。

首先，当用户打开 PC 的电源后，CPU 将自动进入实模式，这时 BIOS 进行开机自检，并按 BIOS 中设置的启动设备(通常是硬盘)进行启动引导。BIOS 通常是转向硬盘的第一个扇区，寻找用于装载操作系统的指令。装载操作系统的这个程序就是 BootLoader。针对不同的硬件平台，需要有专门的 Bootloader 程序。Bootloader 程序依赖于特定的硬件。

Linux 里面的 BootLoader 通常是 lilo 或者 grub，从 Red Hat Linux 7.2 起，GRUB(GRand Unified Bootloader)取代 lilo 成为默认的启动装载程序。

对于嵌入式 Linux 系统，经常使用另一款功能强大的 BootLoader：Blob。Blob 是 Boot Loader Object 的缩写，它遵循 GPL，源代码完全开放。Blob 既可以用来简单地调试，也可以启动 Linux 内核。

下面以 Red Hat 为例，简单介绍 Linux 在 PC 上运行时的启动过程。

当用户打开 PC 的电源，BIOS 开机自检，按 BIOS 中设置的启动设备(通常是硬盘)启动，接着启动设备上安装的引导程序 lilo 或 grub 开始引导 Linux。Linux 首先进行内核的引导，接下来执行 init 程序。init 程序调用 rc.sysinit 和 rc 等程序，rc.sysinit 和 rc 完成系统初始化和运行服务的任务后，返回 init；init 启动 mingetty 后，打开终端供用户登录系统，用户登录成功后进入 Shell，这样就完成了从开机到登录的整个启动过程。启动流程如图 3.6 所示。

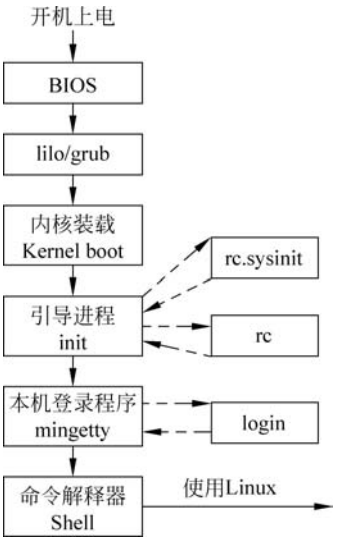

图 3.6　Linux 的启动过程

1. 启动内核

计算机启动时，BIOS 装载 MBR，然后从当前活动分区启动，LILO 获得引导过程的控制权后，会显示 LILO 提示符。此时如果用户不进行任何操作，LILO 将在等待指定时间后自动引导默认的操作系统，而如果在此期间按下 Tab 键，则可以看到一个可引导的操作系统列表，选择相应的操作系统名称就能进入相应的操作系统。

当用户选择启动 Linux 操作系统时，LILO 就会根据事先设置好的信息从 ROOT 文件系统所在的分区读取 Linux 映象，然后装入内核映象并将控制权交给 Linux 内核。Linux

内核获得控制权后,以如下步骤继续引导系统。

(1) Linux 内核一般是压缩保存的,因此,它首先要进行自身的解压缩。内核映象前面的一些代码完成解压缩。

(2) 如果系统中安装有可支持特殊文本模式的、且 Linux 可识别的 SVGA 卡,Linux 会提示用户选择适当的文本显示模式。但如果在内核的编译过程中预先设置了文本模式,则不会提示选择显示模式。该显示模式可通过 LILO 或 RDEV 工具程序设置。

(3) 内核接下来检测其他的硬件设备,例如硬盘、软驱和网卡等,并对相应的设备驱动程序进行配置。这时,显示器上出现内核运行输出的一些硬件信息。

(4) 接下来,内核装载 ROOT 文件系统。ROOT 文件系统的位置可在编译内核时指定,也可通过 LILO 或 RDEV 指定。文件系统的类型可自动检测。如果由于某些原因装载失败,则内核启动失败,最终会终止系统。

2. 执行 init 程序

利用 init 程序可以方便地定制启动期间装入哪些程序。init 的任务是启动新进程和退出时重新启动其他进程。例如,在大多数 Linux 系统中,启动时最初装入 6 个虚拟的控制台进程,退出控制台窗口时,进程死亡,然后 init 启动新的虚拟登录控制台,因而总是提供 6 个虚拟登录控制台。控制 init 程序操作的规则存放在文件/etc/inittab 中。Red Hat Linux 默认的 inittab 文件如下。

```
# inittab This file describes how the INIT process should set up the system in a certain
# run-level.
# Default runlevel. The runlevels used by RHS are:
# 0 - halt(Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS(the same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot(Do NOT set initdefault to this)
id: 3: initdefault:

# system initialization
si::sysinit: /etc/rc.d/rc.sysinit
10:0:wait: /etc/rc.d/rc 0
11:1:wait: /etc/rc.d/rc 1
12:2:wait: /etc/rc.d/rc 2
13:3:wait: /etc/rc.d/rc 3
14:4:wait: /etc/rc.d/rc 4
15:5:wait: /etc/rc.d/rc 5
16:6:wait: /etc/rc.d/rc 6
# Things to run in every runlevel
ud:once: /sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel: /sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes of
```

```
power left. Schedule a
# shutdown for 2 minutes from now.
# This does, of course, assume you have powered installed and your UPS
connected and working
# correctly.
pf::powerfail: /sbin/shutdown -f -h 2 "Power Restored;Shutdown Cancelled"

# Run gettys in standard runlevels
1: 2345: respawn: /sbin/minggetty tty1
2: 2345: respawn: /sbin/minggetty tty2
3: 2345: respawn: /sbin/minggetty tty3
4: 2345: respawn: /sbin/minggetty tty4
5: 2345: respawn: /sbin/minggetty tty5
6: 2345: respawn: /sbin/minggetty tty6
# Run xdm in runlevel 5

x: 5: respawn: /usr/bin/X11/xdm -nodaemon
```

Linux 有个运行级系统，运行级是表示系统当前状态和 init 应运行哪个进程并保持在这种系统状态中运行的数字。在 inittab 文件中，第一个项目指定启动时装入的默认运行级。

上例中是个多用户控制台方式，运行级为 3。然后，inittab 文件中每个项目指定第 2 个字段的项目用哪种运行级（每个字段用冒号分开）。因此，对运行级 3，下列行是相关的。

```
13: 3: wait: /etc/rc.d/rc 3
1: 2345: respawn: /sbin/minggetty tty1
2: 2345: respawn: /sbin/minggetty tty2
3: 2345: respawn: /sbin/minggetty tty3
4: 2345: respawn: /sbin/minggetty tty4
5: 2345: respawn: /sbin/minggetty tty5
6: 2345: respawn: /sbin/minggetty tty6
```

最后 6 行建立 Linux 提供的 6 个虚拟控制台。第一行运行启动脚本/etc/rc.d/rc 3，将运行目录/etc/rc.d/rc3.d 中包含的所有脚本，这些脚本表示系统初始化时要启动的程序。一般来说，这些脚本不需要编辑或改变，是系统默认的。

3.4.2　ARM Linux 操作系统

ARM Linux 是一种常见的嵌入式操作系统，主要运行在以 ARM 为核心的处理器上。根据运行的层次，可以划分为三大部分：启动引导（Bootloader）、操作系统内核（Linux Kernel）和文件系统（File System）。

启动引导程序 Bootloader 非常像 PC 中的 BIOS 程序，主要负责初始化系统的最基本设备，通常主要包括 CPU、网络、串行接口。当基本部分初始化成功后，会把操作系统的镜像文件装载到内存中，最后把 CPU 的控制权交给内核程序。

内核接管系统后，会重新检查外部器件的运行状态，初始化所有外部硬件设备，加载驱动程序，检查系统参数表，装载文件系统，运行 SHELL 程序，等待用户输入命令，或直接运行设定好的应用程序。内核在运行的过程中，会把基本的初始化信息打印到终端（通常是串口 0

或LCD),并且通过终端接收用户命令,它负责控制应用程序的运行状态,实现对整个系统的控制。Linux内核是Linux的最核心部分,内核的优劣决定了整个系统是否稳定与高效。

文件系统是一种数据结构,使操作系统明确存储介质(Flash或硬盘等)上的文件,即在存储介质上组织文件的方法。文件系统通常占用大部分的存储空间,主要负责保存应用程序和数据,由Linux内核管理。

Bootloader、KERNEL、FS(FILE SYSTEM)都存储在Flash中,运行时,根据需要被加载到内存里。图3.7给出了板上内存的地址空间分布:MEMORY MAP。

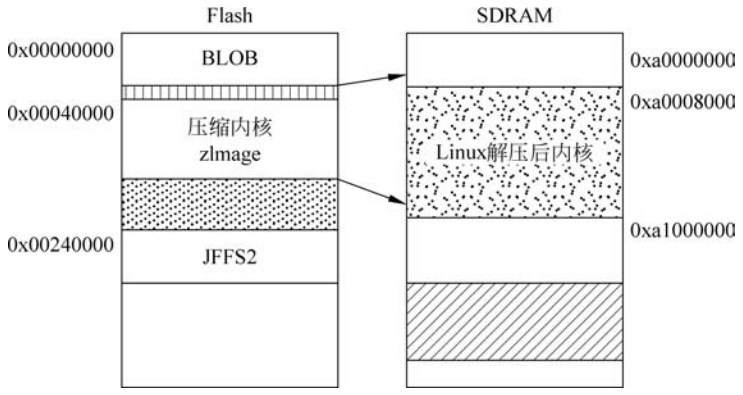

图3.7 内存的地址空间分布:MEMORY MAP

3.5 数据共享与数据传输


视频讲解

3.5.1 应用串口通信协议传输数据

1. 串口通信协议

串口通信协议由Xmodem、Ymodem、Zmodem等协议组成。

Xmodem协议是一种应用于串口通信的文件传输协议。这种协议以包为传输信息的单位来传输数据,并且每个包都使用一个校验和过程来进行错误检测。1个包=128字节,传输速度较慢。

Ymodem协议由Xmodem协议演变而来,传输效率及可靠性均较高,它的1个包=1024字节。Ymodem一次传输可发送或接收多个文件。

Zmodem协议也是由Xmodem协议演变而来,以连续的数据流发送数据,传输效率更高。

2. Windows系统主机传输文件到Linux系统开发板

当需要把Windows系统主机的文件传输到Linux系统开发板时,可以使用本方法来实现。首先,用串口通信数据线连接Windows系统主机和Linux系统开发板,如图3.8所示。

1) 在Windows系统主机端设置发送文件

在Windows系统主机的桌面"开始"菜单中,选择"程序"|"附件"|"通信"|"超级终端"项,打开"连接描述"对话框,填写超级终端连接的名称,单击"确定"按钮,如图3.9所示。Windows 7以后版本没有自带"超级终端",可以从网络上搜索下载"超级终端"软件。

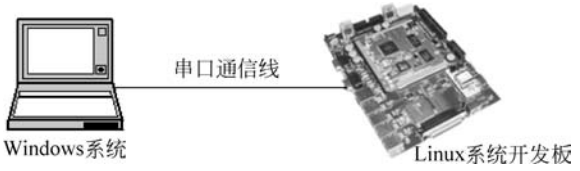

图 3.8　用串口通信数据线连接 Windows 系统主机和 Linux 系统开发板

在打开的“连接到”对话框中选择串口通信的连接端口为 COM1 端口，单击“确定”按钮，如图 3.10 所示。

图 3.9　设置超级终端的连接名称

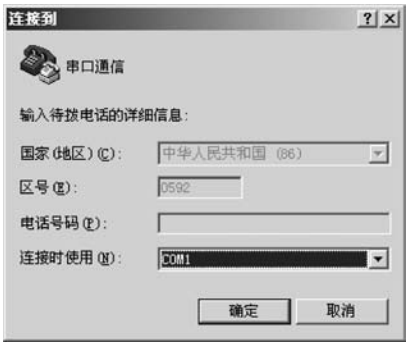

图 3.10　选择 COM1 为连接端口

在打开的“COM1 属性”对话框中，设置端口的各个重要参数值：每秒位数（波特率）为 115200，数据位为 8 位，奇偶校验为“无”，停止位为 1，数据流控制为“无”，如图 3.11 所示。设置好端口的参数值后，单击“确定”按钮。

图 3.11　设置端口参数

2）在 Linux 系统开发板端设置接收文件

在开发板端设置接收文件的操作很简单，只需通过 minicom 窗口，进入准备接收数据文件的目录等待发送来的文件即可。

3）发送数据

在 Windows 系统主机端，继续前面的“超级终端”窗口操作。

在超级终端的串口通信窗口的“发送”菜单中，选择“发送文件”项，如图 3.12 所示。

在弹出的“发送文件”对话框中，单击“浏览”按钮，选择需要传送的数据文件；然后在“协议”下拉列表框中，选择 Xmodem 协议，如图 3.13 所示。

这时，在“为串口通信发送 Xmodem 文件”窗口可以看到数据传送的过程，如图 3.14 所示。

文件传输完毕后，在开发板的接收数据文件的目录中可以看到传送的文件。

图 3.12 在超级终端选择“发送文件”菜单项

图 3.13 选择发送的文件和 Xmodem 协议

图 3.14 传输数据

3. Linux 系统主机传输数据到 Linux 系统开发板

经常需要把在 Linux 系统主机上经过交叉编译后的文件传输到 Linux 系统开发板运行,可以使用本方法来实现传送文件。

首先,用串口通信数据线连接 Linux 系统主机和 Linux 系统开发板,如图 3.15 所示。

1) 在开发板端设置接收文件

通过 minicom 窗口操作开发板端文件系统,进入准备接收数据文件的目录,等待发送来的文件。

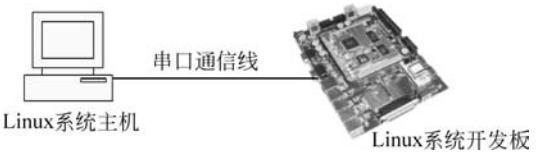

图 3.15　用串口通信数据线连接 Linux 系统主机和 Linux 系统开发板

2）从 Linux 系统主机端发送文件

在 minicom 窗口中，按下 Ctrl＋A＋S 快捷键，弹出选择传输数据协议的对话框，如图 3.16 所示。

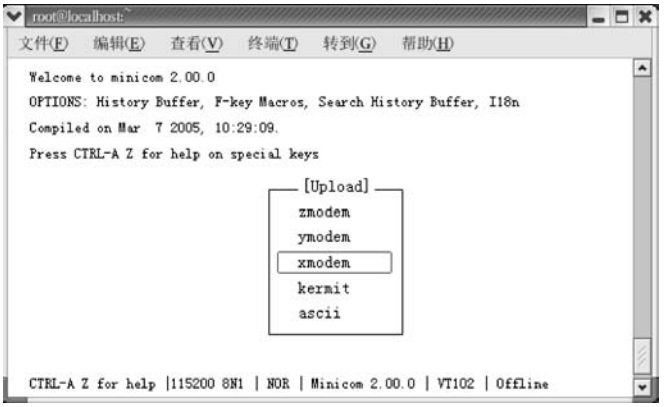

图 3.16　在 minicom 窗口中弹出选择传输数据协议的对话框

用键盘上的方向键移动光标至 xmodem 项后，按 Enter 键确定。进入 Linux 系统主机端的文件系统，按“上”“下”方向键移动文件目录前面的小方块，按两次空格键进入选定的目录，若要返回到上一级目录，则选中[..]后按两次空格键，如图 3.17 所示。

图 3.17　按“上”“下”方向键移动小方块，选择需要的文件目录

进入需要的文件目录后，按“上”“下”方向键移动小方块，选择需要传送的文件。找到需要的文件按空格键选中该文件；若要取消选中，则再次按空格键，如图 3.18 所示。

选中需要传输的文件，按 Enter 键，则开始发送文件。当显示 Transfer incomplete 时，表示文件传输完毕，如图 3.19 所示。

图 3.18 按空格键选中需要的文件

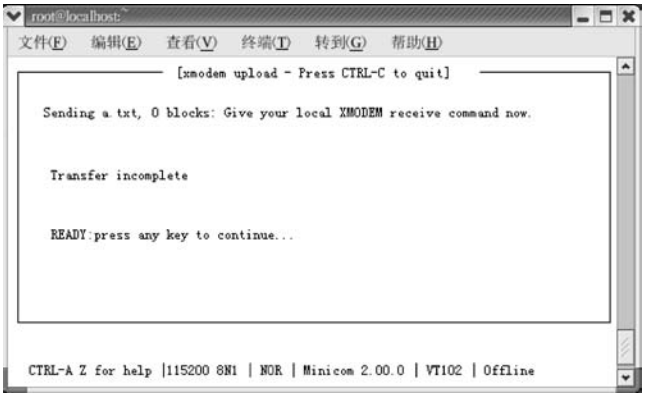

图 3.19 发送文件完毕

视频讲解

3.5.2 在 VMware 虚拟机中设置 Windows 与 Linux 系统的数据共享

在 VMware 虚拟机中可以设置 Windows 与 Linux 系统的共享。设 Windows 操作系统的 VMware 中安装有 Linux 操作系统,通过 VMware 虚拟机可以设置 Windows 与 Linux 系统的共享,如图 3.20 所示。

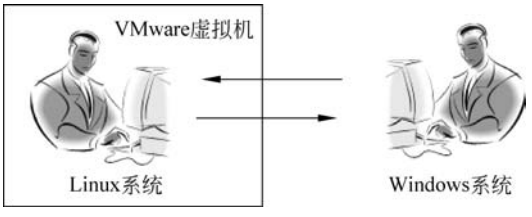

图 3.20 在 VMware 虚拟机中设置 Windows 与 Linux 系统的数据共享

1. 安装 VMware Tools

在 VMware 虚拟机中选择“虚拟机(VM)”菜单,在弹出的下拉菜单中选择 Install VMware Tools 项,Linux 系统桌面上会出现一个名为 VMware Tools 的光盘图标。

双击 VMware Tools 光盘图标，打开光盘，复制 VMware Tools. tar. gz 文件到/home 目录下，将其解压至/home/vmware-tools-distrib 目录下。进入安装目录/home/vmware-tools-distrib 中，在终端运行如下命令。

```
./vmware - install.pl
```

安装过程中会有一些文件安装路径的提示问题，按 Enter 键即可，如图 3.21 所示。

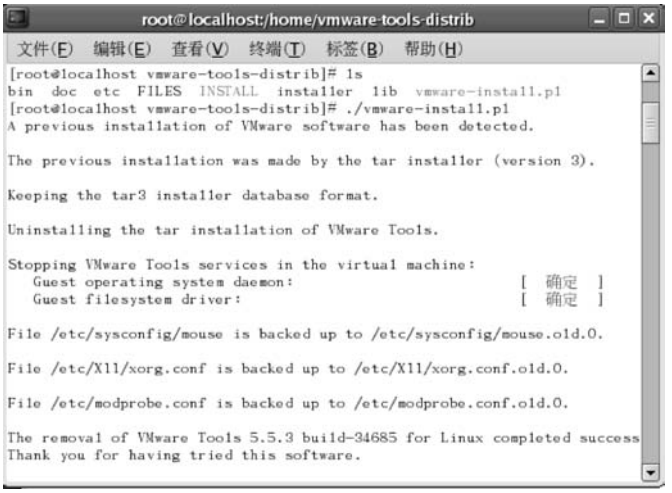

图 3.21　在终端运行 vmware-install. pl

2. 设置共享文件夹

选择 VMware 虚拟机“虚拟机(VM)”菜单中的“设置(Settings)”项，弹出“虚拟机设置”对话框。选择“选项”选项卡，在左侧选择“共享文件夹”项，然后单击右侧的“添加”按钮，添加 Windows 系统中的共享文件夹，如图 3.22 所示。

图 3.22　设置 Windows 系统的共享文件夹

3. 在 Linux 系统中操作 Windows 系统的共享文件夹

在 Linux 系统中,打开/mnt 目录,可以看到其中存在一个 hgfs 目录。打开/mnt/hgfs 目录,可以看到 Windows 系统的共享文件夹,如图 3.23 所示。因而在 Linux 系统中,可以很方便地对这些共享文件夹的文件进行复制、删除、修改等操作。

图 3.23 在 Linux 系统中操作 Windows 系统的共享文件夹

本章小结

本章介绍了 Linux 的基本概念、Linux 文件系统的概念、嵌入式 Linux 系统中常用的命令、Linux 系统的文本编辑器的使用、Linux 系统的启动过程等。这些都是 Linux 中最基础、最常见的概念,掌握和理解这些知识对进一步学习和使用 Linux 系统有很大帮助,因此,必须多上机练习,熟练掌握它们。

习 题

1. 查看 Linux 目录结构,说出下列目录放置的数据类型。

```
/etc/:
/etc/rc.d/init.d/:
/usr/bin:
/bin:
/sbin:
/dev:
```

2. Bootloader 有什么作用? 为什么不作为操作系统的一部分加以实现?

3. 叙述在主机端配置 NFS 服务的过程。

4. 应用串口协议传输,把主机端的一个文件传输到开发板上,并记录下操作过程。

5. 在 VMware 虚拟机中建立 Windows 操作系统与 Linux 操作系统的数据共享文件目录。