

2.2.4 常用 Shell 命令

我们做嵌入式开发用的最多就是 Shell 命令，Shell 命令是所有的 Linux 系统发行版所通用的，并不是说我在 Ubuntu 下学会了 Shell 命令，换另外一个 Linux 发行版操作系统以后就没用了(不同的发行版 Linux 系统可能会自定义一些命令)。本节我们先来介绍一些 Shell 下常用的命令：

1、目录信息查看命令 ls

文件浏览是最基本的操作了，Shell 下文件浏览命令为 ls，格式如下：

```
ls [选项] [路径]
```

ls 命令主要用于显示指定目录下的内容，列出指定目录下包含的所有的文件以及子目录，它的主要参数有：

- a 显示所有的文件以及子目录，包括以“.”开头的隐藏文件。
- l 显示文件的详细信息，比如文件的形态、权限、所有者、大小等信息。
- t 将文件按照创建时间排序列出。
- A 和-a 一样，但是不列出“.”(当前目录)和“..”(父目录)。
- R 递归列出所有文件，包括子目录中的文件。

Shell 命令里面的参数是可以组合在一起用的，比如组合“-al”就是显示所有文件的详细信息，包括以“.”开头的隐藏文件，ls 命令使用如图 2.2.4.1 所示：

```
zuozhongkai@ubuntu:~/tmp$ ls
a b c
zuozhongkai@ubuntu:~/tmp$ ls -a
. .. a b c
zuozhongkai@ubuntu:~/tmp$ ls -al
总用量 8
drwxrwxr-x  2 zuozhongkai zuozhongkai 4096 12月 19 21:41 .
drwxr-xr-x 20 zuozhongkai zuozhongkai 4096 12月 19 20:31 ..
-rw-rw-r--  1 zuozhongkai zuozhongkai   0 12月 19 21:41 a
-rw-rw-r--  1 zuozhongkai zuozhongkai   0 12月 19 21:41 b
-rw-rw-r--  1 zuozhongkai zuozhongkai   0 12月 19 21:41 c
```

图 2.2.4.1 ls 命令演示

注意图 2.2.4.1 中 tmp 文件夹是我为了演示方便，自己创建的，里面的文件 a, b 和 c 也是我创建的，关于文件夹和文件的创建后面会详细的讲解。

2、目录切换命令 cd

要想在 Shell 中切换到其它的目录，使用的命令是 cd，命令格式如下：

```
cd [路径]
```

路径就是我们要进入的目录路径，比如下面所示操作：

```
cd /           //进入到根目录“/”下，Linux 系统的根目录为“/”，
cd /usr        //进入到目录“/usr”里面。
cd ..          //进入到上一级目录。
cd ~           //切换到当前用户主目录
```

比如我们要进入到目录“/usr”下去，并且查看“/usr”下有什么文件，操作如图 2.2.4.2 所示：

```
zuozhongkai@ubuntu:~$ cd /usr
zuozhongkai@ubuntu:/usr$ ls
bin  games  include  lib  local  locale  sbin  share  src
```

图 2.2.4.2 cd 命令演示

在图 2.2.4.2 中，我们先使用命令“cd /usr”进入到“/usr”目录下，然后使用“ls”命令显示“/usr”目录下的所有文件。仔细观察图 2.2.4.2 可以看到，当我们切换到其它目录以后在符号“\$”前面就会以蓝色的字体显示出当前目录名字，如图 2.2.4.3 所示：

```
zuozhongkai@ubuntu:/usr$ ls
bin  games  include  lib  local  locale  sbin  share  src
```

图 2.2.4.3 目录路径显示

3、当前路径显示命令 pwd

pwd 命令用来显示当前工作目录的绝对路径，不需要任何的参数，使用如图 2.2.4.4 所示：

```
zuozhongkai@ubuntu:~$ pwd
/home/zuozhongkai
```

图 2.2.4.4 pwd 命令

4、系统信息查看命令 uname

要查看当前系统信息，可以使用命令 `uname`，命令格式如下：

```
uname [选项]
```

可选的选项参数如下：

- r 列出当前系统的具体内核版本号。
- s 列出系统内核名称。
- o 列出系统信息。

使用如图 2.2.4.5 所示：

```
zuozhongkai@ubuntu:~$ uname
Linux
zuozhongkai@ubuntu:~$ uname -r
4.15.0-29-generic
zuozhongkai@ubuntu:~$ uname -s
Linux
zuozhongkai@ubuntu:~$ uname -o
GNU/Linux
```

图 2.2.4.5 uanme 命令操作

5、清屏命令 clear

clear 命令用于清除终端上的所有内容，只留下一行提示符。

6、切换用户执行身份命令 sudo

Ubuntu(Linux)是一个允许多用户的操作系统，其中权限最大的就是超级用户 root，有时候我们执行一些操作的时候是需要用 root 用户身份才能执行，比如安装软件。通过 `sudo` 命令可以使我们暂时将身份切换到 root 用户。当使用 `sudo` 命令的时候是需要输入密码的，这里要注意输入密码的时候是没有任何提示的！命令格式如下：

```
sudo [选项] [命令]
```

选项主要参数如下：

- h 显示帮助信息。

- l 列出当前用户可执行与不可执行的命令
- p 改变询问密码的提示符。

假如我们现在要创建一个新的用户 `test`，创建新用户的命令为“`adduser`”，创建新用户的权限只有 `root` 用户才有，我们在装系统的时候创建的那个用户是没有这个权限的，比如我的“`zuozhongkai`”用户。所以创建新用户的话需要使用“`sudo`”命令以 `root` 用户执行“`adduser`”这个命令，如图 2.2.4.6 所示：

```
zuozhongkai@ubuntu:~$ adduser test
adduser: 只有 root 才能将用户或组添加到系统。
zuozhongkai@ubuntu:~$ sudo adduser test
正在添加用户"test"...
正在添加新组"test" (1001)...
正在添加新用户"test" (1001) 到组"test"...
创建主目录"/home/test"...
正在从"/etc/skel"复制文件...
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
正在改变 test 的用户信息
请输入新值，或直接敲回车键以使用默认值
    全名 []:
    房间号码 []:
    工作电话 []:
    家庭电话 []:
    其它 []:
这些信息是否正确? [Y/n] y
```

图 2.2.4.6 `sudo` 命令演示

在图 2.2.4.6 中，我们一开始直接使用“`adduser test`”命令添加用户的时候提示我们“`adduser: 只有 root 才能将用户或组添加到系统。`”所以我们要在前面加上“`sudo`”命令，表示以 `root` 用户执行 `adduser` 操作。

7、添加用户命令 `adduser`

在讲解 `sudo` 命令的时候我们已经用过命令“`adduser`”，此命令需要 `root` 身份去运行。命令格式如下：

```
adduser [参数] [用户名]
```

常用的参数如下：

- system 添加一个系统用户
- home DIR DIR 表示用户的主目录路径
- uid ID ID 表示用户的 `uid`。
- ingroup GRP 表示用户所属的组名。

`adduser` 的使用我们前面已经演示过了，大家可以试着再添加一个用户。

8、删除用户命令 `deluser`

前面讲了添加用户的命令，那肯定也有删除用户的命令，删除用户使用命令“`deluser`”，命令参数如下：

```
deluser [参数] [用户名]
```

主要参数有：

- system 当用户是一个系统用户的时候才能删除。
- remove-home 删除用户的主目录
- remove-all-files 删除与用户有关的所有文件。

-backup 备份用户信息

同样的，命令“deluser”也要使用“sudo”来以 root 用户运行，以删除我们前面创建的用户 test 为例，deluser 使用如图 2.2.4.7 所示：

```
zuozhongkai@ubuntu:~$ sudo deluser --remove-all-files test
正在寻找要备份或删除的文件...
/usr/sbin/deluser: 无法处理特殊文件 /sys/kernel/security/apparmor/.null
/usr/sbin/deluser: 无法处理特殊文件 /run/udev/static_node-tags/uaccess/snd\x2fseq
/usr/sbin/deluser: 无法处理特殊文件 /run/udev/static_node-tags/uaccess/snd\x2ftimer
/usr/sbin/deluser: 无法处理特殊文件 /dev/vcsa7
/usr/sbin/deluser: 无法处理特殊文件 /dev/vcs7
.....
/usr/sbin/deluser: 无法处理特殊文件 /lib/systemd/system/single.service
/usr/sbin/deluser: 无法处理特殊文件 /lib/systemd/system/cryptdisks.service
/usr/sbin/deluser: 无法处理特殊文件 /lib/systemd/system/cryptdisks-early.service
正在删除文件...
正在删除用户 'test'...
警告：组 "test"没有其它成员了。
完成。
```

图 2.2.4.7 命令 deluser 演示

9、切换用户命令 su

前面在讲解命令“sudo”的时候说过，“sudo”是以 root 用户身份执行一个命令，并没有更改当前的用户身份，所有需要 root 身份执行的命令都必须在前面加上“sudo”。命令“su”可以直接将当前用户切换为 root 用户，切换到 root 用户以后就可以尽情的进行任何操作了！因为你已经获得了系统最高权限，在 root 用户下，所有的命令都可以无障碍执行，不需要在前面加上“sudo”，“su”命令格式如下：

```
su [选项] [用户名]
```

常用选项参数如下：

- c -command** 执行指定的命令，执行完毕以后恢复原用户身份。
- login** 改变用户身份，同时改变工作目录和 PATH 环境变量。
- m** 改变用户身份的时候不改变环境变量
- h** 显示帮助信息

以切换到 root 用户为例，使用如图 2.2.4.8 所示：

```
zuozhongkai@ubuntu:~$ sudo su
[sudo] zuozhongkai 的密码：
root@ubuntu:/home/zuozhongkai#
```

图 2.2.4.8 su 命令演示

在图 2.2.4.8 中，先使用命令“sudo su”切换到 root 用户，su 命令不写明用户名的话默认切换到 root 用户。然后输入密码，密码正确的话就会切换到 root 用户，可以看到切换到 root 用户以后提示符的“@”符号前面的用户名变成了“root”，表示当前的用户是 root 用户。并且以“#”结束。

注意！！ 由于 root 用户权限太大，稍微不注意就可能删除掉系统文件，导致系统奔溃，因此强烈建议大家，不要以 root 用户运行 Ubuntu。当要用到 root 身份执行某些命令的时候使用“sudo”命令即可。

要切换回原来的用户，使用命令“sudo su 用户名”即可，比如我要从 root 切换回 zuozhongkai 这个用户，操作如图 2.2.4.9 所示：

```
root@ubuntu:/home/zuozhongkai# sudo su zuozhongkai
zuozhongkai@ubuntu:~$
```

图 2.2.4.9 切换回原来用户

10、显示文件内容命令 cat

查看文件内容是最常见的操作了，在 windows 下可以直接使用记事本查看一个文本文件内容，linux 下也有类似记事本的软件，叫做 gedit，找到一个文本文件，双击打开，默认使用的就是 gedit，如图 2.2.4.10 所示：



图 2.2.4.10 gedit 打开文档

我们现在讲解的是 Shell 命令，那么 Shell 下有没有办法读取文件的内容呢？肯定有的，那就是命令“cat”，命令格式如下：

```
cat [选项] [文件]
```

选项主要参数如下：

- n 由 1 开始对所有输出的行进行编号。
- b 和-n 类似，但是不对空白行编号。
- s 当遇到连续两个行以上空白行的话就合并为一个行空白行。

比如我们以查看文件“/etc/environment”的内容为例，结果如图 2.2.4.11 所示：

```
zuozhongkai@ubuntu:~$ cat /etc/environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"
zuozhongkai@ubuntu:~$ cat /etc/environment -n
1 PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"
```

图 2.2.4.11 命令 cat 演示

11、显示和配置网络属性命令 ifconfig

ifconfig 是一个跟网络属性配置和显示密切相关的命令，通过此命令我们可以查看当前网络属性，也可以通过此命令配置网络属性，比如设置网络 IP 地址等等，此命令格式如下：

```
ifconfig interface options | address
```

主要参数如下：

- interface** 网络接口名称，比如 eth0 等。
- up** 开启网络设备。
- down** 关闭网络设备。
- add** IP 地址，设置网络 IP 地址。

`netmask add` 子网掩码。

命令 `ifconfig` 的使用如图 2.2.4.12 所示：

```
zuozhongkai@ubuntu:~$ ifconfig
ens33    Link encap:以太网  硬件地址 00:0c:29:96:89:d6
         inet 地址:192.168.31.235  广播:192.168.31.255  掩码:255.255.255.0
         inet6 地址: fe80::e92a:d882:e1b:7402/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
         接收数据包:13404  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:967  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:1000
         接收字节:810508 (810.5 KB)  发送字节:75728 (75.7 KB)

lo       Link encap:本地环回
         inet 地址:127.0.0.1  掩码:255.0.0.0
         inet6 地址: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  跃点数:1
         接收数据包:248  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:248  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:1000
         接收字节:19004 (19.0 KB)  发送字节:19004 (19.0 KB)

zuozhongkai@ubuntu:~$ ifconfig ens33
ens33    Link encap:以太网  硬件地址 00:0c:29:96:89:d6
         inet 地址:192.168.31.235  广播:192.168.31.255  掩码:255.255.255.0
         inet6 地址: fe80::e92a:d882:e1b:7402/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
         接收数据包:13406  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:971  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:1000
         接收字节:810658 (810.6 KB)  发送字节:76218 (76.2 KB)
```

图 2.2.4.12 `ifconfig` 命令演示

在图 2.2.4.12 中有两个网卡：`ens33` 和 `lo`，`ens33` 是我的电脑实际使用的网卡，`lo` 是回测网卡。可以看出网卡 `ens33` 的 IP 地址为 192.168.31.235，我们使用命令“`ifconfig`”将网卡 `ens33` 的 IP 地址改为 192.168.31.20，操作如图 2.2.4.13 所示：

```
zuozhongkai@ubuntu:~$ sudo ifconfig ens33 192.168.31.20
zuozhongkai@ubuntu:~$ ifconfig ens33
ens33    Link encap:以太网  硬件地址 00:0c:29:96:89:d6
         inet 地址:192.168.31.20  广播:192.168.31.255  掩码:255.255.255.0
         inet6 地址: fe80::e92a:d882:e1b:7402/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
         接收数据包:15188  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:1040  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:1000
         接收字节:917930 (917.9 KB)  发送字节:84590 (84.5 KB)
```

图 2.2.4.13 修改网卡 IP 地址

从图 2.2.4.13 可以看出，我在使用命令“`ifconfig`”修改网卡 `ens33` 的 IP 地址的时候使用了“`sudo`”，说明在 Ubuntu 下修改网卡 IP 地址是需要 root 用户权限的。当修改完以后使用命令“`ifconfig ens33`”再次查看网卡 `ens33`，发现网卡 `ens33` 的 IP 地址变成了 192.168.31.20

12、系统帮助命令 `man`

Ubuntu 系统中有很多命令，这些命令都有不同的格式，不同的格式对应不同的功能，要完全记住这些命令和格式几乎是不可能的，必须有一个帮助手册，当我们需要了解一个命令的详细信息的时候查阅这个帮助手册就行了。Ubuntu 提供了一个命令来帮助用户完成这个功能，那就是“`man`”命令，通过“`man`”命令可以查看其它命令的语法格式、主要功能、主要参数说明

等，“man”命令格式如下：

```
man [命令名]
```

比如我们要查看命令“ifconfig”的说明，输入“man ifconfig”即可，如图 2.2.4.14 所示：

```
zuozhongkai@ubuntu:~$ man ifconfig
```

图 2.2.4.14 man 命令演示

在终端中输入图 2.2.4.14 所示的命令，然后点击回车键就会打开“ifconfig”这个命令的详细说明，如图 2.2.4.15 所示：

```
IFCONFIG(8)                Linux Programmer's Manual                IFCONFIG(8)
NAME
  ifconfig - configure a network interface
SYNOPSIS
  ifconfig [-v] [-a] [-s] [interface]
  ifconfig [-v] interface [atype] options | address ...
DESCRIPTION
  Ifconfig is used to configure the kernel-resident network interfaces.
  It is used at boot time to set up interfaces as necessary. After that,
  it is usually only needed when debugging or when system tuning is
  needed.

  If no arguments are given, ifconfig displays the status of the cur-
  rently active interfaces. If a single interface argument is given, it
  displays the status of the given interface only; if a single -a argu-
  ment is given, it displays the status of all interfaces, even those
  that are down. Otherwise, it configures an interface.

Address Families
  If the first argument after the interface name is recognized as the
Manual page ifconfig(8) line 1 (press h for help or q to quit)
```

图 2.2.4.15 命令“ifconfig”详细介绍信息

图 2.2.4.15 就是命令“ifconfig”的详细介绍信息，按“q”键退出到终端。

13、系统重启命令 reboot

通过点击 Ubuntu 主界面右上角的齿轮按钮来选择关机或者重启系统，同样的我们也可以使用 Shell 命令“reboot”来重启系统，直接输入命令“reboot”然后点击回车键即可，如图 2.2.4.16 所示：

```
zuozhongkai@ubuntu:~$ reboot
```

图 2.2.4.16 reboot 命令演示

14、系统关闭命令 poweroff

使用命令“reboot”可以重启系统，使用命令“poweroff”就可以关闭系统，在终端中输入命令“poweroff”然后按下回车键即可关闭 Ubuntu 系统，如图 2.2.4.17 所示：

```
zuozhongkai@ubuntu:~$ poweroff
```

图 2.2.4.17 poweroff 命令演示

15、软件安装命令 install

截至目前，我们都没有讲过 Ubuntu 下如何安装软件，因为 Ubuntu 安装软件不像 Windows 下那样，直接双击 exe 文件就开始安装了。Ubuntu 下很多软件是需要先自行下载源码，下载源

码以后自行编译，编译完成以后使用命令“`install`”来安装。当然 Ubuntu 下也有其它的软件安装方法，但是用的最多的就是自行编译源代码然后安装，尤其是嵌入式 Linux 开发。命令“`install`”格式如下：

```
install [选项]... [-T] 源文件      目标文件
或：install [选项]...      源文件...  目录
或：install [选项]... -t  目录      源文件...
或：install [选项]... -d  目录...
```

“`install`”命令是将文件(通常是编译后的文件)复制到目的位置，在前三种形式中，将源文件复制到目标文件或将多个源文件复制到一个已存在的目录中同时设置其所有权和权限模式。在第四种形式会创建指定的目录。命令“`install`”通常和命令“`apt-get`”组合在一起使用的，关于“`apt-get`”命令我们稍后会讲解。

以上就是 Shell 最基本一些命令，还有一些其它的命令我们在后面在讲解，循序渐进嘛。

2.4 APT 下载工具

对于长时间使用 Windows 的我们，下载安装软件非常容易，Windows 下有很多的下载软件，Ubuntu 同样有不少的下载软件，本节我们讲解 Ubuntu 下我们用的最多的下载工具：APT 下载工具，APT 下载工具可以实现软件自动下载、配置、安装二进制或者源码的功能。APT 下载工具和我们前面讲解的“`install`”命令结合在一起构成了 Ubuntu 下最常用的下载和安装软件方法。它解决了 Linux 平台下安装软件的一个缺陷，即软件之间相互依赖。

APT 采用的 C/S 模式，也就是客户端/服务器模式，我们的 PC 机作为客户端，当需要下载软件的时候就向服务器请求，因此我们需要知道服务器的地址，也叫做安装源或者更新源。打开系统设置，打开“软件和更新”设置，打开以后如图 2.4.1.1 所示：



图 2.4.1.1 软件和更新设置

在图 2.4.1.1 中的“Ubuntu 软件”选项卡下面的“下载自”就是 APT 工具的安装源，因为

我们是在中国，所以需要选择中国的服务器，否则的话可能会导致下载失败！这个也就是网上说的 Ubuntu 安装成功以后要更新源。

在我们使用 APT 工具下载安装或者更新软件的时候，首先会在下载列表中与本机软件对比，看一下需要下载哪些软件，或者升级哪些软件，默认情况下 APT 会下载最新的软件包，被安装的软件包所依赖的其它软件也会被下载安装。说了这么多，APT 下载工具究竟怎么用呢？APT 工具常用的命令如下：

1、更新本地数据库

如果想查看本地哪些软件可以更新的话可以使用如下命令：

```
sudo apt-get update
```

这个命令会访问源地址，并且获取软件列表并保存在本电脑上，过程如图 2.4.1.2 所示：

```
zuozhongkai@ubuntu:~$ sudo apt-get update
[sudo] zuozhongkai 的密码：
命中:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
命中:2 http://cn.archive.ubuntu.com/ubuntu xenial InRelease
获取:3 http://cn.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
获取:4 http://cn.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
已下载 216 kB，耗时 13秒 (16.5 kB/s)
正在读取软件包列表... 完成
```

图 2.4.1.2 更新本地数据库

2、检查依赖关系

有时候本地某些软件可能存在依赖关系，所谓依赖关系就是 A 软件依赖于 B 软件。通过如下命令可以查看依赖关系，如果存在依赖关系的话 APT 会提出解决方案：

```
sudo apt-get check
```

上述命令的执行结果如图 2.4.1.3 所示：

```
zuozhongkai@ubuntu:~$ sudo apt-get check
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
```

图 2.4.1.3 检查依赖关系

3、软件安装

这个是重点了，安装软件，使用如下命令：

```
sudo apt-get install package-name
```

可以看出上述命令是由“apt-get”和“install”组合在一起的，“package-name”就是要安装的软件名字，“apt-get”负责下载软件，“install”负责安装软件。比如我们要安装软件 Ubuntu 下的串口工具“minicom”，我们就可以使用如下命令：

```
sudo apt-get install minicom
```

执行上述命令以后就会自动下载和安装 minicom 软件，如图 2.4.1.3 所示：

```
zuozhongkai@ubuntu:~$ sudo apt-get install minicom
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
  lrzsz
下列【新】软件包将被安装：
  lrzsz minicom
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 92 个软件包未被升级。
需要下载 306 kB 的归档。
解压缩后会消耗 1,193 kB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取:1 http://cn.archive.ubuntu.com/ubuntu xenial/universe amd64 lrzsz amd64 0.12.21-8 [73.8 kB]
获取:2 http://cn.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 minicom amd64 2.7-1+deb8u1build0.16.04.1 [232 kB]
已下载 306 kB，耗时 3秒 (80.8 kB/s)
正在选中未选择的软件包 lrzsz。
(正在读取数据库 ... 系统当前共安装有 217399 个文件和目录。)
正准备解包 .../lrzsz_0.12.21-8_amd64.deb ...
正在解包 lrzsz (0.12.21-8) ...
正在选中未选择的软件包 minicom。
正准备解包 .../minicom_2.7-1+deb8u1build0.16.04.1_amd64.deb ...
正在解包 minicom (2.7-1+deb8u1build0.16.04.1) ...
正在处理用于 man-db (2.7.5-1) 的触发器 ...
正在设置 lrzsz (0.12.21-8) ...
正在设置 minicom (2.7-1+deb8u1build0.16.04.1) ...
```

图 2.4.1.3 安装 minicom 软件

图 2.4.1.3 就是安装 minicom 这个软件的过程，在图 2.4.1.3 中的安装过程中，会有如下所示询问：

```
您希望继续执行吗？ [Y/n]
```

如果希望继续执行的话就输入 y，如果不希望继续执行的话就输入 n。安装完成以后我们直接在终端输入如下命令打开 minicom 这个串口软件：

```
minicom -s
```

打开以后的 minicom 软件如图 2.4.1.4 所示：

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols     |
| Serial port setup           |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
```

图 2.4.1.4 minicom 软件

关于 minicom 的使用大家可以上网搜索一下，这里就不详细讲解了，要退出 minicom 可以直接按下 ESC 键

4、软件更新

有时候我们需要更新软件，更新软件的话使用命令：

```
sudo apt-get upgrade package-name
```

其中 package-name 为要升级的软件名字，比如我们升级刚刚安装的 minicom 这个软件，如图 2.4.1.5 所示：

```
zuozhongkai@ubuntu:~$ sudo apt-get upgrade minicom
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
minicom 已经是最新版 (2.7-1+deb8u1build0.16.04.1)。
正在计算更新... 完成
下列软件包的版本将保持不变：
  ubuntu-minimal
```

图 2.4.1.5 更新 minicom 软件

从图 2.4.1.5 可以看出，minicom 已经是最新的了，不用更新，不过有其它软件需要更新，因此会自动更新其它的软件。

5、卸载软件

如果要卸载某个软件的话使用如下命令：

```
sudo apt-get remove package-name
```

其中 package-name 是要卸载的软件，比如卸载前面安装的 minicom 这个软件，操作如图 2.4.1.6 所示：

```
zuozhongkai@ubuntu:~$ sudo apt-get remove minicom
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  lrzsz
使用 'sudo apt autoremove' 来卸载它(它们)。
下列软件包将被【卸载】：
  minicom
升级了 0 个软件包，新安装了 0 个软件包，要卸载 1 个软件包，有 1 个软件包未被升级。
解压缩后将会空出 928 kB 的空间。
您希望继续执行吗？ [Y/n] y
(正在读取数据库 ... 系统当前共安装有 217494 个文件和目录。)
正在卸载 minicom (2.7-1+deb8u1build0.16.04.1) ...
正在处理用于 man-db (2.7.5-1) 的触发器 ...
```

图 2.4.1.6 卸载软件

从图 2.4.1.6 中可以看出软件 minicom 被卸载掉了。关于 APT 下载工具就讲解到这里，我们用的最多的就是“sudo apt-get install package-name”来下载和安装软件。有关 Ubuntu 其它的安装软件的方法打开可以自行上网查阅学习，这里就不一一详解了。

2.5 Ubuntu 下文本编辑

2.5.1 Gedit 编辑器

进行文本编辑是最常用的操作，Windows 下我们会使用记事本来完成，或者其它一些优秀的文本编辑器，比如 notepad++，Ubuntu 下有一个自带的文本编辑器，那就是 Gedit。Gedit 是一个窗口式的编辑器，关于 Gedit 的使用前面我们已经讲解了。本节我们重点讲解的是另外一

个编辑器：VI/VIM 编辑器。

2.5.2 VI/VIM 编辑器

我们如果要在终端模式下进行文本编辑或者修改文件就可以使用 VI/VIM 编辑器，Ubuntu 自带了 VI 编辑器，但是 VI 编辑器对于习惯了 Windows 下进行开发的人来说不方便，比如竟然不能使用键盘上的上下左右键调整光标位置。因此我推荐大家使用 VIM 编辑器，VIM 编辑器是 VI 编辑器升级版本，VI/VIM 编辑器都是一种基于指令式的编辑器，不需要鼠标，也没有菜单，仅仅使用键盘来完成所有的编辑工作。

我们需要先安装 VIM 编辑器，命令如下：

```
sudo apt-get install vim
```

安装完成以后就可以使用 VIM 编辑器了，VIM 编辑器有 3 种工作模式：输入模式、指令模式和底行模式，通过切换不同的模式可以完成不同的功能，我们就以编辑一个文本文档为例讲解 VIM 编辑器的使用。打开终端，输入命令：`vi test.txt`，如图 2.5.2.1 所示：

```
zuozhongkai@ubuntu:~$ vim test.txt
```

图 2.5.2.1 新建 test.txt 文档

在终端中输入图 2.5.2.1 中所示的命令以后就会创建一个 test.txt 文档，并且用 VIM 打开了，如图 2.5.2.2 所示：

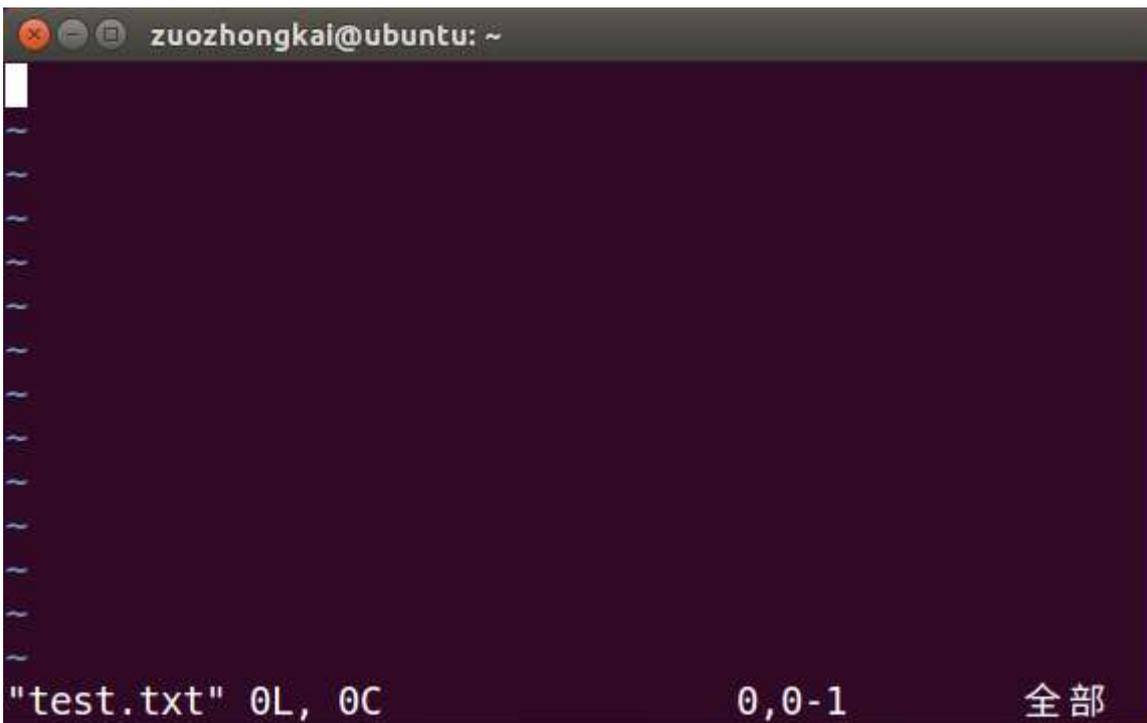


图 2.5.2.2 VIM 打开的 test.txt 文档

我们试着在图 2.5.2.2 中输入数字，发现根本没法输入，这不是因为你的键盘坏了。因为 VIM 默认是以只读模式打开的文档，因此我们要切换到输入模式，切换到输入模式的命令如下：

- i** 在当前光标所在字符的前面，转为输入模式。
- I** 在当前光标所在行的行首转换为输入模式。
- a** 在当前光标所在字符的后面，转为输入模式。
- A** 在光标所在行的行尾，转换为输入模式。
- o** 在当前光标所在行的下方，新建一行，并转为输入模式。
- O** 在当前光标所在行的上方，新建一行，并转为输入模式。
- s** 删除光标所在字符。
- r** 替换光标处字符。

最常用的就是“a”，我们在图 2.5.2.2 中按下键盘上的“a”键，这时候终端左下角会提示“插入”字样，表示我们进入到了输入模式，如图 2.5.2.3 所示：

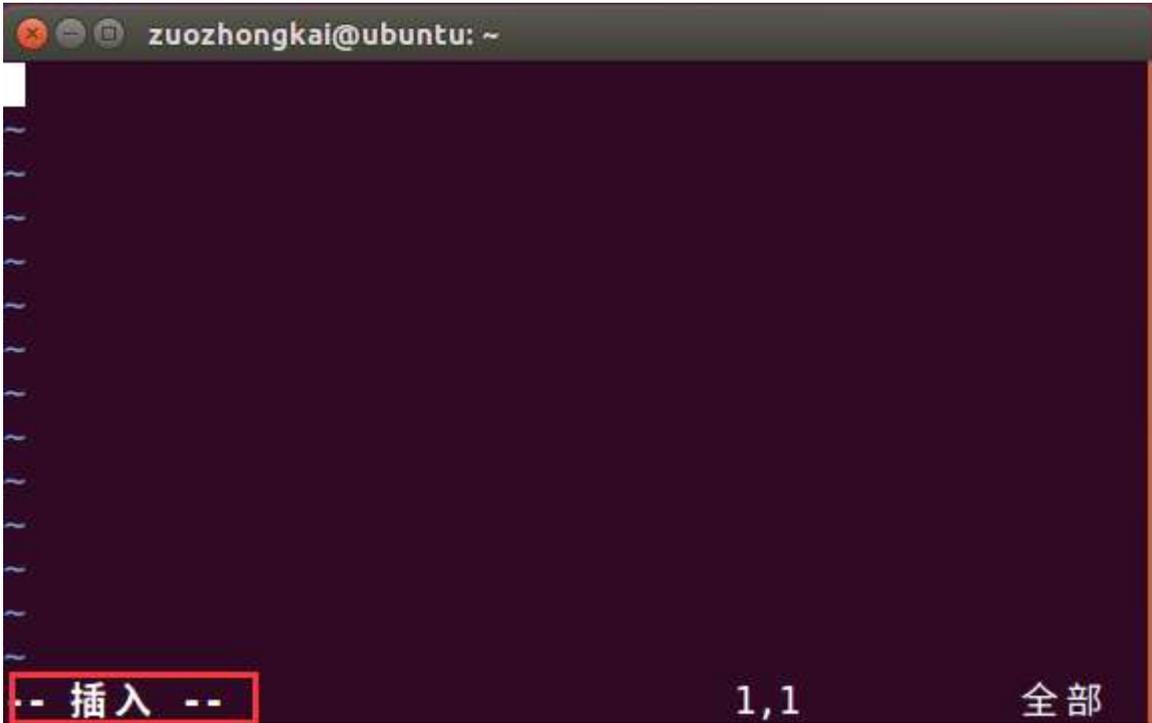


图 2.5.2.3 切换到插入模式

图 2.5.2.3 表明我们可以正常输入文本了，我们可以输入图 2.5.2.4 所示文本：

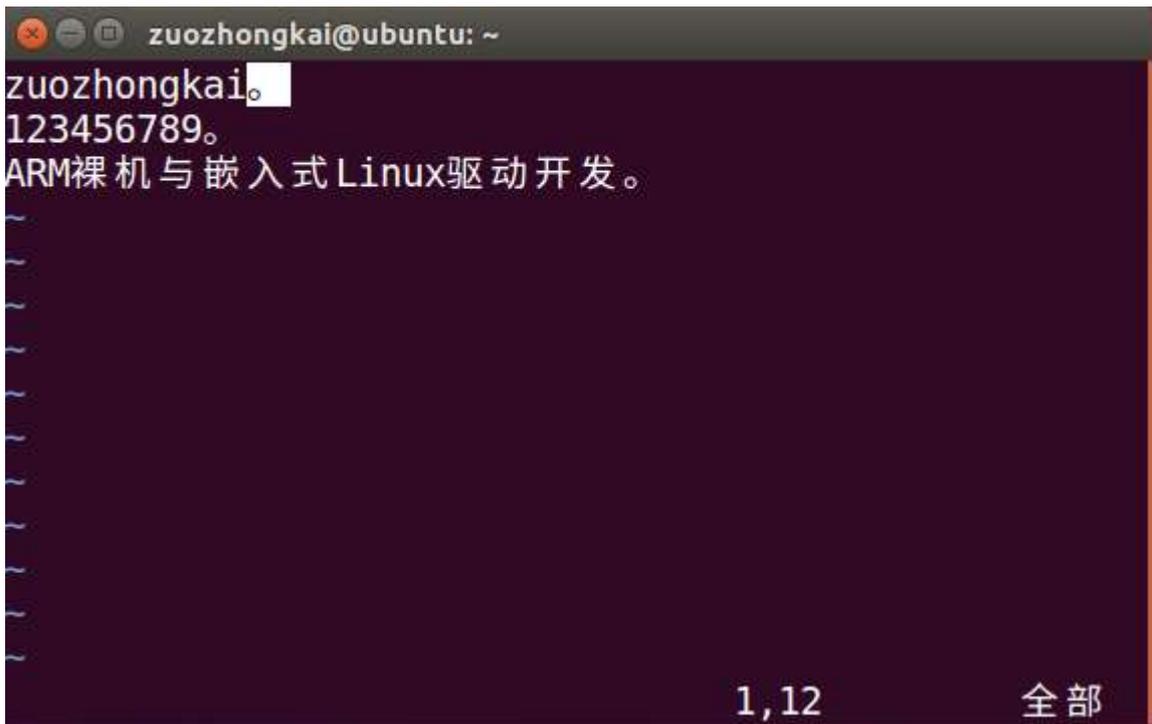


图 2.5.2.4 输入文本

在图 2.5.2.4 中我们在 test.txt 中输入了字母、数字和中文，我们输入完成以后需要保存文本啊，Windows 下的记事本可以使用快捷键 Ctrl+S 来保存，VIM 是否也可以使用 Ctrl+S 来保存呢？你会发现当你按下 Ctrl+S 键以后你的终端不能操作了!!! 这是因为在 Ubuntu 下 Ctrl+S 快捷键不是用来完成保存的功能的，而是暂停该终端！所以你一旦在使用终端的时候按下 Ctrl+S 快捷键，那么你的终端肯定不会再有任何反应，如果你按下 Ctrl+S 关闭了当前终端的话可以按下 Ctrl+Q 来重新打开终端。

既然 Ctrl+S 不能保存文本文档，那么有没有其它方法保存文本文档呢？肯定是有的，我们需要从 VIM 现在的输入模式切换到指令模式，方式就是按下键盘的 ESC 键，按下 ESC 键以后终端左下角的“插入”字样就会消失，此时你就不能在输入任何文本了，如果想再次输入文本的话就按下“a”键重新进入到输入模式。指令模式顾名思义就是输入指令的模式，这些指令是

控制文本的指令，我们将这些指令进行分类，如下所示：

1、移动光标指令：

h(或左方向键)	光标左移一个字符。
l(或右方向键)	光标右移一个字符。
j(或下方向键)	光标下移一行。
k(或上方向键)	光标上移一行。
nG	光标移动到第 n 行首。
n+	光标下移 n 行。
n-	光标上移 n 行。

2、屏幕翻滚指令

Ctrl+f	屏幕向下翻一页，相当于下一页。
Ctrl+b	屏幕向上翻一页，相当于上一页。

3、复制、删除和粘贴指令

cc	删除整行，并且修改整行内容。
dd	删除该行，不提供修改功能。
ndd	删除当前行向下 n 行。
x	删除光标所在的字符。
X	删除光标前面的一个字符。
nyy	复制当前行及其下面 n 行。
p	粘贴最近复制的内容。

上面就是 VI/VIM 的命令模式下最常用的一些命令，还有一些不常用的我没有列出来，感兴趣的可以自行上网查阅。从上面的命令可以看出，并没有保存文本的命令，那是因为保存文档的命令是在底行模式中，我们要先进入到指令模式，进入底行模式的方式是先进入指令模式下，然后在指令模式下输入“:”进入底行模式，如图 2.5.2.5 所示：

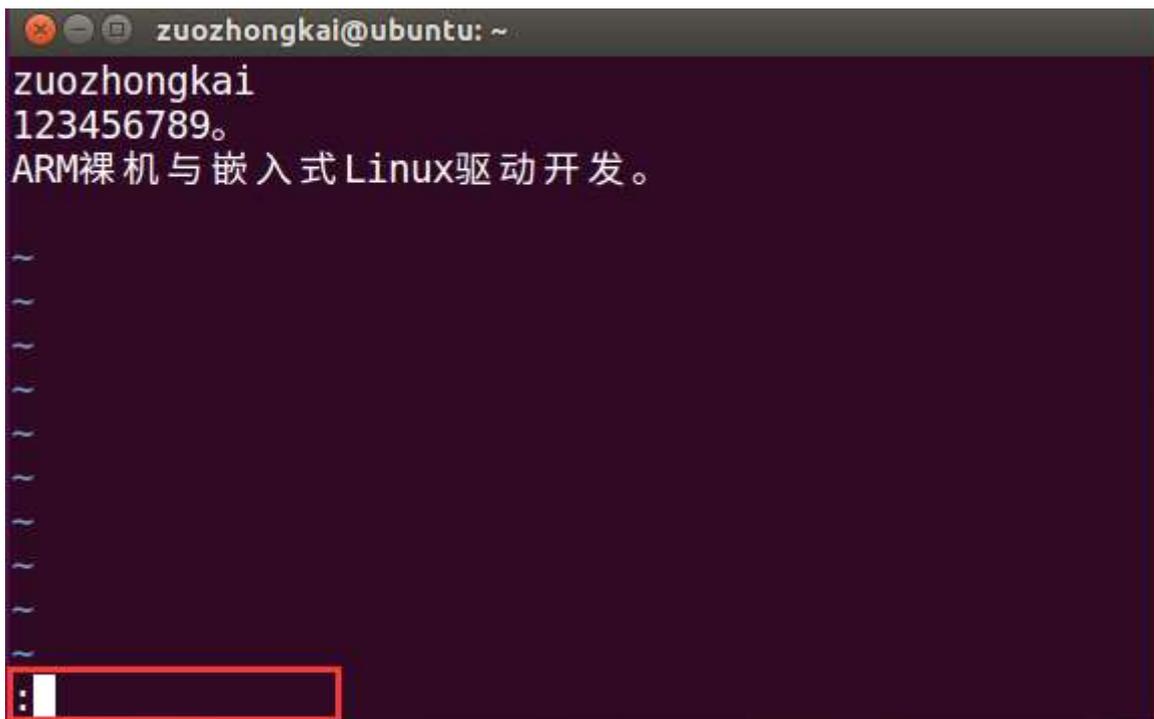


图 2.5.2.5 “:”底行模式

在图 2.5.2.5 中当进入底行模式以后会在终端的左下角就会出现符号“:”，我们可以在“:”后面输入命令，常用的命令如下：

x 保存当前文档并且退出。

q 退出。

w 保存文档。

q! 退出 VI/VIM，不保存文档。

如果我们要退出并保存文本的话需要在“:”底行模式下输入“wq”，如图 2.5.2.6 所示：

A terminal window titled 'zuozhongkai@ubuntu: ~' showing the VIM editor. The editor content is:
zuozhongkai
123456789。
ARM裸机与嵌入式Linux驱动开发。
The bottom line of the editor shows ':wq' in a red box, indicating the user is in bottom line mode and has entered the command to save and quit.

图 2.5.2.6 保存并退出 VIM

在“:”底行模式下输入“wq”以后按下回车键就保存 test.txt 并退出 VI/VIM 编辑器，退出以后我们可以使用命令“cat”来查看刚刚新建的 test.txt 文档的内容，如图 2.5.2.7 所示：

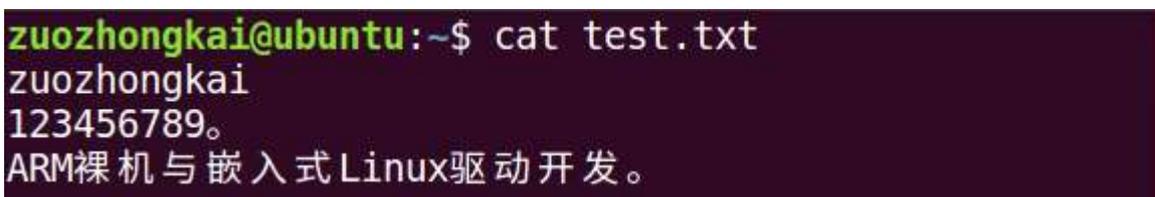
A terminal window showing the command 'cat test.txt' being executed. The output is:
zuozhongkai
123456789。
ARM裸机与嵌入式Linux驱动开发。

图 2.5.2.7 查看文档内容

从图 2.5.2.7 中可以看出，test.txt 中的内容就是我们用 VIM 输入的内容，至此我们就完整的进行了一遍 VI/VIM 创建文档、编辑文档和保存文档。

在上面讲解进入 VIM 的底行模式的时候之说了在指令模式下输入“:”的方法，还可以在指令模式下输入“/”进入底行模式，输入“/”以后如图 2.5.2.8 所示。



图 2.5.2.8 “/” 底行模式

在“/”底行模式下我们可以在文本中搜索指定的内容，比如搜索 test.txt 文件中“嵌入式”三个字，使用方法如图 2.5.2.9 所示：



图 2.5.2.9 搜索文本

在“/”后面输入要搜索的内容，然后按下回车键就会在 test.txt 中找到与字符串“嵌入式”匹配的部分，如图 2.5.2.10 所示：

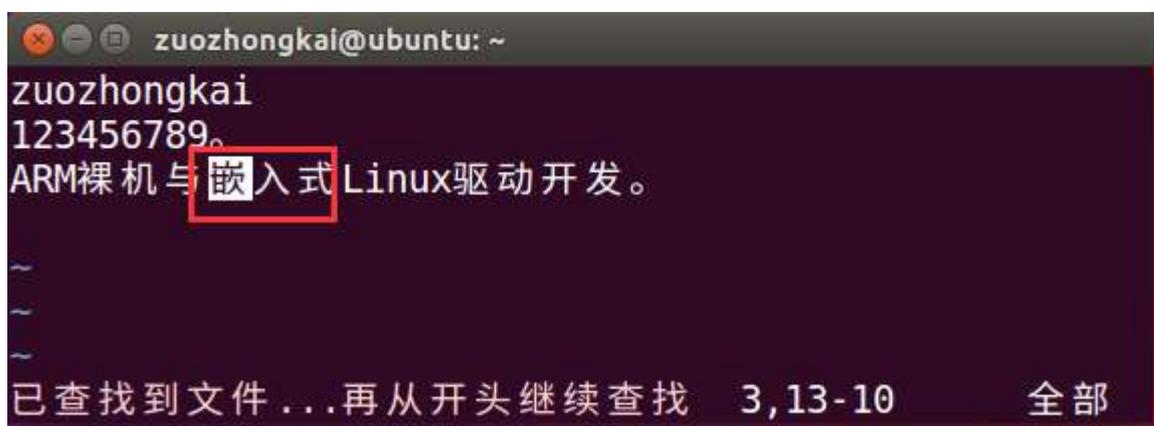


图 2.5.2.10 查找到指定内容

图 2.5.2.10 中可以看出，在 test.txt 中找到了“嵌入式”这个词，并且标记出来位置。我们以后要在一个文档中搜索是否存在某个字符串的时候就可以使用这种方法。有关 VI/VIM 编辑器的讲解就到这里，我们完整的练习了一遍如何使用 VIM 创建文档、编辑文档、保存文档和在文档中搜索字符串。有关更多更详细的 VIM 编辑器的操作大家自行上网查阅相关文档和博客。

2.6 Linux 文件系统

操作系统的基本功能之一就是文件管理，而文件的管理是由文件系统来完成的。Linux 支持多种文件系统，本节我们就来讲解 Linux 下的文件系统、文件系统类型、文件系统结构和文件系统相关 Shell 命令。

2.6.1 Linux 文件系统简介以及类型

1、Linux 文件系统简介

操作系统就是处理各种数据的，这些数据在硬盘上就是二进制，人类肯定不能直接看懂这些二进制数据，要有一个翻译器，将这些二进制的数数据还原为人类能看懂的文件形式，这个工作就是由文件系统来完成的，文件系统的目的就是实现数据的查询和存储，由于使用场合、使用环境的不同，Linux 有多种文件系统，不同的文件系统支持不同的体系。文件系统是管理数据的，而可以存储数据的物理设备有硬盘、U 盘、SD 卡、NAND FLASH、NOR FLASH、网络存储设备等。不同的存储设备其物理结构不同，不同的物理结构就需要不同的文件系统去管理，比如管理 NAND FLASH 的话使用 YAFFS 文件系统，管理硬盘、SD 卡的话就是 ext 文件系统等等。

我们在使用 Windows 的时候新买一个硬盘回来一般肯定是将这个硬盘分为好几个盘，比如 C 盘、D 盘等等。这个叫磁盘的分割，Linux 下也支持磁盘分割，Linux 下常用的磁盘分割工具为：fdisk，fdisk 这个工具我们后面会详细讲解怎么用，因为我们移植 Linux 的时候需要将 SD 卡分为三个分区来存储不同的东西。在 Windows 下我们创建一个新的盘符以后都要做格式化处理，格式化其实就是给这个盘符创建文件系统的过程，我们在 Windows 格式化某个盘的时候都会让你选择文件系统，如图 2.6.1.1 所示：



图 2.6.1.1 格式化磁盘

图 2.6.1.1 就是格式化磁盘的时候选择文件系统，Windows 下一般有 FAT、NTFS 和 exFAT 这些文件系统。同样的，在 Linux 下我们使用 fdisk 创建好分区以后也是要先在创建好的分区上面创建文件系统，也就是格式化。

在 Windows 下有磁盘分区概念，比如 C，D，E 盘等，在 Linux 下没有这个概念，因此 Linux 下你找不到像 C、D、E 盘这样的东西。前面我们说了 Linux 下可以给磁盘分割，但是没有

C、D、E 盘那怎么访问这些分区呢？在 Linux 下创建一个分区并且格式化好以后我们要将其“挂载”到一个目录下才能访问这个分区。Windows 的文件系统挂载过程是其内部完成的，用户是看不到的，Linux 下我们使用 mount 命令来挂载磁盘。挂载磁盘的时候是需要确定挂载点的，也就是你的这个磁盘要挂载到哪个目录下。

2、Linux 文件系统类型

前面我们说了，在 Windows 下有 FAT、NTFS 和 exFAT 这样的文件系统，在 Linux 下又有哪些文件系统呢，Linux 下的文件系统主要有 ext2、ext3、ext4 等文件系统。Linux 还支持其他的 UNIX 文件系统，比如 XFS、JFS、UFS 等，也支持 Windows 的 FAT 文件系统和网络文件系统 NFS 等。这里我们主要讲一下 Linux 自带的 ext2、ext3 和 ext4 文件系统。

ext2 文件系统：

ext2 是 Linux 早期的文件系统，但是随着技术的发展 ext2 文件系统已经不推荐使用了，ext2 是一个非日志文件系统,大多数的 Linux 发行版都不支持 ext2 文件系统了。

ext3 文件系统：

ext3 是在 ext2 的基础上发展起来的文件系统，完全兼容 ext2 文件系统，ext3 是一个日志文件系统，ext3 支持大文件，ext3 文件系统的特点有如下：

高可靠性：使用 ext3 文件系统的话，即使系统非正常关机、发生死机等情况，恢复 ext3 文件系统也只需要数十秒。

数据完整性：ext3 提高了文件系统的完整性，避免意外死机或者关机对文件系统的伤害。

文件系统速度：ext3 的日志功能对磁盘驱动器读写头进行了优化，文件系统速度相对与 ext2 来说没有降低。

数据转换：从 ext2 转换到 ext3 非常容易，只需要两条指令就可以完成转换。用户不需要花时间去备份、恢复、格式化分区等，用 ext3 文件系统提供的工具 tune2fs 即可轻松的将 ext2 文件系统转换为 ext3 日志文件系统。ext3 文件系统不需要经过任何修改，可以直接挂载成 ext2 文件系统。

ext4 文件系统：

ext4 文件系统是在 ext3 上发展起来的，ext4 相比与 ext3 提供了更佳的性能和可靠性，并且功能更丰富，ext4 向下兼容 ext3 和 ext2，因此可以将 ext2 和 ext3 挂载为 ext4。那么我们安装的 Ubuntu 使用的哪个版本的文件系统呢？在终端中输入如下命令来查询当前磁盘挂载的啥文件系统：

```
df -T -h
```

结果如图 2.6.1.2 所示：



```
zuozhongkai@ubuntu:/$ df -T -h
文件系统      类型      容量  已用  可用  已用%  挂载点
udev          devtmpfs  3.9G   0    3.9G   0%    /dev
tmpfs         tmpfs     796M   9.3M  787M   2%    /run
/dev/sda1     ext4      192G   5.2G  177G   3%    /
tmpfs         tmpfs     3.9G   252K  3.9G   1%    /dev/shm
tmpfs         tmpfs     5.0M   4.0K  5.0M   1%    /run/lock
tmpfs         tmpfs     3.9G   0     3.9G   0%    /sys/fs/cgroup
tmpfs         tmpfs     796M   56K   796M   1%    /run/user/1000
```

图 2.6.1.2 Ubuntu 使用的文件系统

在图 2.6.1.2 中，框起来的就是我们安装 Ubuntu 的这个磁盘，在 Linux 下一切皆为文件，“/dev/sda1”就是我们的磁盘分区，可以看出这个磁盘分区类型是 ext4，它的挂载点是“/”，也就

是根目录。

2.6.2 Linux 文件系统结构

在 Windows 下直接打开 C 盘，我们进入的就是 C 盘的根目录，打开 D 盘进入的就是 D 盘的根目录，比如 C 盘根目录如下：

> 本地磁盘 (C:)

名称	修改日期	类型	大小
\$WINDOWS.~BT	2017-01-18 10:25	文件夹	
\$Windows.~WS	2017-12-17 11:34	文件夹	
AppData	2018-08-27 14:47	文件夹	
DRMsoft	2017-09-07 11:24	文件夹	
ESD	2017-12-17 12:11	文件夹	
Intel	2016-09-07 15:13	文件夹	
M1530_MFP_Series_Basic_Solution	2016-12-29 23:30	文件夹	
PerfLogs	2016-07-16 19:47	文件夹	
Program Files	2017-12-13 15:08	文件夹	
Program Files (x86)	2018-12-18 0:04	文件夹	
ProgramData	2018-12-17 21:57	文件夹	
touchgfx-env	2017-12-05 11:37	文件夹	
TouchGFXProjects	2018-01-07 11:22	文件夹	
uacdump	2016-11-28 12:39	文件夹	
Users	2016-09-08 13:00	文件夹	
Windows	2018-10-31 23:49	文件夹	
InstallConfig.ini	2017-10-13 23:12	配置设置	1 KB

图 2.6.2.1 C 盘根目录

在 Linux 下因为没有 C、D 盘之说，因此 Linux 只有一个根目录，没有 C 盘根目录、D 盘根目录之类的。其实如果你的 Windows 只有一个 C 盘的话那么整个系统也就只有一个根目录。

Windows 下的 C 盘根目录就是“C:”，在 Linux 下的根目录就是“/”，你没有看错，Linux 根目录就是用“/”来表示的，打开 Ubuntu 的文件浏览器，文件浏览器在左侧的导航栏，图标如图 2.6.2.2 所示：



图 2.6.2.2 文件浏览器

打开以后的文件浏览器如图 2.6.2.3 所示：



图 2.6.2.3 文件浏览器

直接打开文件浏览器以后，我们默认不是处于根目录中的，不像 Windows，我们直接打开 C 盘就处于 C 盘根目录下。Ubuntu 是支持多用户的，Ubuntu 为每个用户创建了一个根目录，比如我电脑现在登陆的是“zuozhongkai”这个用户，因此默认进入的是“zuozhongkai”这个用户的根目录。我们点击图 2.6.2.3 中左侧的“计算机”，打开以后如图 2.6.2.4 所示：



图 2.6.2.4 根目录 “/”

图 2.6.2.4 就是 Ubuntu 的根目录 “/”，这时候肯定就有人有疑问，刚刚说 Ubuntu 会给每个

用户创建一个根目录，那这些用户的根目录在哪里？是不是和根目录“/”是一个地位的？其实所谓的给每个用户创建一个根目录只是方便说而已，这个所谓的用户根目录其实就是“/”下的一个文件夹，以我的“zuozhongkai”这个用户为例，其用户根目录就是：/home/zuozhongkai。只要你创建了一个用户，那么系统就会在/home 这个目录下创建一个以这个用户名命名的文件夹，这个文件夹就是这个用户的根目录。

用户可以对自已的用户根目录下的文件进行随意的读写操作，但是如果修改根目录“/”下的文件就会提示没有权限。打开终端以后默认进入的是当前用户根目录，比如我们打开终端以后输入“ls”命令查看当前目录下有什么文件，结果如图 2.6.2.5 所示：

```
zuozhongkai@ubuntu:~$ ls
examples.desktop  tmp      模板  图片  下载  桌面
test.txt          公共的  视频  文档  音乐
```

图 2.6.2.5 目录查看

可以看出图 2.6.2.5 中的文件和图 2.6.2.3 中的一模一样，都是“zuozhongkai”这个账户的根目录。我们来看一下根目录“/”下都有哪些文件，在终端中输入如下命令：

```
cd / //进入到根目录“/”
ls //查看根目录“/”下的文件以及文件夹
```

执行上述两行命令以后，终端如图 2.6.2.6 所示：

```
zuozhongkai@ubuntu:~$ cd /
zuozhongkai@ubuntu:/$ ls
bin    dev    initrd.img    lib64    mnt    root    snap    tmp    vmlinuz
boot   etc    initrd.img.old  lost+found  opt    run    srv    usr    vmlinuz.old
cdrom  home  lib           media    proc   sbin   sys    var
```

图 2.6.2.6 查看根目录“/”

图 2.6.2.6 中列举出了根目录“/”下面的所有文件夹，这里我们仔细观察一下，当我们进入到根目录“/”里面以后终端提示符“\$”前面的符号“~”变成了“/”，这是因为当我们在终端中切换了目录以后“\$”前面就会显示切换以后的目录路径。我们来看一下根目录“/”中的一些重要的文件夹：

- /bin** 存储一些二进制可执行命令文件，/usr/bin 也存放了一些基于用户的命令文件。
- /sbin** 存储了很多系统命令，/usr/sbin 也存储了许多系统命令。
- /root** 超级用户 root 的根目录文件。
- /home** 普通用户默认目录，在该目录下，每个用户都有一个以本用户名命名的文件夹。
- /boot** 存放 Ubuntu 系统内核和系统启动文件。
- /mnt** 通常包括系统引导后被挂载的文件系统的挂载点。
- /dev** 存放设备文件，我们后面学习 Linux 驱动主要是跟这个文件夹打交道的。
- /etc** 保存系统管理所需的配置文件和目录。
- /lib** 保存系统程序运行所需的库文件，/usr/lib 下存放了一些用于普通用户的库文件。
- /lost+found** 一般为空，当系统非正常关机以后，此文件夹会保存一些零散文件。
- /var** 存储一些不断变化的文件，比如日志文件
- /usr** 包括与系统用户直接有关的文件和目录，比如应用程序和所需的库文件。
- /media** 存放 Ubuntu 系统自动挂载的设备文件。
- /proc** 虚拟目录，不实际存储在磁盘上，通常用来保存系统信息和进程信息。
- /tmp** 存储系统和用户的临时文件，该文件夹对所有的用户都提供读写权限。
- /opt** 可选文件和程序的存放目录。
- /sys** 系统设备和文件层次结构，并向用户程序提供详细的内核数据信息。

2.6.2 文件操作命令

本节我们来学习一下在终端进行文件操作的一些常用命令：

1、创建新文件命令—touch

在前面学习 VIM 的时候我们知道可以用 vi 指令来创建一个文本文档，本节我们就学习一个功能更全面的文件创建命令—touch。touch 不仅仅可以用来创建文本文档，其它类型的文档也可以创建，命令格式如下：

```
touch [参数] [文件名]
```

使用 touch 创建文件的时候，如果[文件名]的文件不存在，那就直接创建一个以[文件名]命名的文件，如果[文件名]文件存在的话就仅仅修改一下此文件的最后修改日期，常用的命令参数如下：

- a 只更改存取时间。
- c 不建立任何文件。
- d<日期> 使用指定的日期，而并非现在日期。
- t<时间> 使用指定的时间，而并非现在时间。

进入到用户根目录下，直接使用命令“cd ~”即可快速进入用户根目录，进入用户根目录以后使用 touch 命令创建一个名为 test 的文件，创建过程如图 2.6.2.1 所示：

```
zuozhongkai@ubuntu:~$ cd ~ //进入用户根目录
zuozhongkai@ubuntu:~$ ls //查看当前目录下的文件
examples.desktop tmp 模板 图片 下载 桌面
test.txt 公共的 视频 文档 音乐
zuozhongkai@ubuntu:~$ touch test //创建test文件
zuozhongkai@ubuntu:~$ ls test //查看创建的test文件
test
zuozhongkai@ubuntu:~$ ls test -l //查看创建的test文件详细信息
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 22 01:24 test
```

图 2.6.2.1 touch 命令操作

2、文件夹创建命令—mkdir

既然可以创建文件，那么肯定也可以创建文件夹，创建文件夹使用命令“mkdir”，命令格式如下：

```
mkdir [参数] [文件夹名目录名]
```

主要参数如下：

- p 如所要创建的目录其上层目录目前还未创建，那么会一起创建上层目录。

我们在用户根目录下创建两个分别名为“testdir1”和“testdir2”的文件夹，操作如图 2.6.2.2 所示：

```
zuozhongkai@ubuntu:~$ ls //查看当前目录下的文件
examples.desktop test.txt 公共的 视频 文档 音乐
test tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ mkdir testdir1 //创建testdir1文件夹
zuozhongkai@ubuntu:~$ mkdir testdir2 //创建testdir2文件夹
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件，看看文件夹创建是否成功
examples.desktop testdir1 test.txt 公共的 视频 文档 音乐
test testdir2 tmp 模板 图片 下载 桌面
```

图 2.6.2.2 创建文件夹

在图 2.6.2.2 中，我们使用命令“mkdir”创建了“testdir1”和“testdir2”这两个文件夹。

3、文件及目录删除命令—rm

既然有创建文件的命令，那肯定有删除文件的命令，要删除一个文件或者文件夹可以使用命令“rm”，此命令可以完成删除一个文件或者多个文件及文件夹，它可以实现递归删除。对于链接文件，只删除链接，原文件保持不变，所谓的链接文件，其实就是 Windows 下的快捷方式文件，此命令格式如下：

```
rm [参数] [目的文件或文件夹目录名]
```

命令主要参数如下：

- d 直接把要删除的目录的硬连接数据删成 0，删除该目录。
- f 强制删除文件和文件夹(目录)。
- i 删除文件或者文件夹(目录)之前先询问用户。
- r 递归删除，指定文件夹(目录)下的所有文件和子文件夹全部删除掉。
- v 显示删除过程。

我们使用 rm 命令来删除前面使用命令“touch”创建的 test 文件，操作过程如图 2.6.2.3 所示：

```
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件
examples.desktop  testdir1  test.txt  公共的  视频  文档  音乐
test              testdir2  tmp       模板    图片  下载  桌面
zuozhongkai@ubuntu:~$ rm test //删除文件test
zuozhongkai@ubuntu:~$ ls //查看当前目录，看看test文件是否删除
examples.desktop  testdir2  tmp       模板  图片  下载  桌面
testdir1         test.txt  公共的  视频  文档  音乐
```

图 2.6.2.3 删除文件

命令“rm”也可以直接删除文件夹，我们可以试一下删除前面创建的 testdir1 文件夹，先直接使用命令“rm testdir1”测试一下是否可以删除，结果如图 2.6.2.4 所示：

```
zuozhongkai@ubuntu:~$ rm testdir1
rm: 无法删除 'testdir1': 是一个目录
```

图 2.6.2.4 删除文件夹

在图 2.6.2.4 中可以看出，直接使用命令“rm”是无法删除文件夹(目录)的，我们需要加上参数“-rf”，也就是强制递归删除文件夹(目录)，操作结果如图 2.6.2.5 所示：

```
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件
examples.desktop  testdir2  tmp       模板  图片  下载  桌面
testdir1         test.txt  公共的  视频  文档  音乐
zuozhongkai@ubuntu:~$ rm testdir1 -rf //使用参数“-rf”强制递归删除文件夹
zuozhongkai@ubuntu:~$ ls //查看删除文件夹以后的当前目录
examples.desktop  test.txt  公共的  视频  文档  音乐
testdir2         tmp      模板    图片  下载  桌面
```

图 2.6.2.5 带参数删除文件夹

从图 2.6.2.5 可以看出，当在命令“rm”中加入参数“-rf”以后就可以删除掉文件夹“testdir1”了。

4、文件夹(目录)删除命令—rmdir

上面我们讲解了如何使用命令“rm”删除文件夹，那就是要加上参数“-rf”，其实 Linux 提供了直接删除文件夹(目录)的命令—rmdir，它可以不加任何参数的删除掉指定的文件夹(目录)，命令格式如下：

```
rmdir [参数] [文件夹(目录)]
```

命令主要参数如下：

-p 删除指定的文件夹(目录)以后，若上层文件夹(目录)为空文件夹(目录)的话就将其一起删除。

我们使用命令“rmdir”删除掉前面创建的“testdir2”文件夹，操作过程如图 2.6.2.6 所示：

```
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件
examples.desktop  test.txt  公共的  视频  文档  音乐
testdir2          tmp      模板  图片  下载  桌面
zuozhongkai@ubuntu:~$ rmdir testdir2 //删除文件夹testdir2
zuozhongkai@ubuntu:~$ ls //查看删除文件夹以后的目录文件
examples.desktop  tmp      模板  图片  下载  桌面
test.txt          公共的  视频  文档  音乐
```

图 2.6.2.6 命令 rmdir 删除文件夹

5、文件复制命令—cp

在 Windows 下我们可以通过在文件上点击鼠标右键来进行文件的复制和粘贴，在 Ubuntu 下我们也可以点击文件右键进行文件的复制和粘贴。但是本节我们来讲解如何在终端下使用命令来进行文件的复制，Linux 下的复制命令为“cp”，命令描述如下：

```
cp [参数] [源地址] [目的地址]
```

主要参数描述如下：

- a** 此参数和同时指定“-dpR”参数相同
- d** 在复制有符号连接的文件时，保留原始的连接。
- f** 强行复制文件，不管要复制的文件是否已经存在于目标目录。
- I** 覆盖现有文件之前询问用户。
- p** 保留源文件或者目录的属性。
- r 或-R** 递归处理，将指定目录下的文件及子目录一并处理

我们在用户根目录下，使用前面讲解的命令“mkdir”创建两个文件夹：test1 和 test2，过程如图 2.6.2.7 所示。

```
zuozhongkai@ubuntu:~$ ls //查看当前目录下的文件
examples.desktop  tmp      模板  图片  下载  桌面
test.txt          公共的  视频  文档  音乐
zuozhongkai@ubuntu:~$ mkdir test1 test2 //创建test1和test2两个文件夹
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件
examples.desktop  test2    tmp      模板  图片  下载  桌面
test1             test.txt 公共的  视频  文档  音乐
```

图 2.6.2.7 创建 test1 和 test2 两个文件夹

进入上面创建的 test1 文件夹，然后在 test1 文件夹里面创建一个 a.c 文件，操作过程如图 2.6.2.8 所示：

```
zuozhongkai@ubuntu:~$ cd test1 //进入test1文件夹
zuozhongkai@ubuntu:~/test1$ touch a.c //创建a.c文件
zuozhongkai@ubuntu:~/test1$ ls //查看当前目录下所有文件
a.c
```

图 2.6.2.8 创建 a.c 文件

我们先将图 2.6.2.8 中的 a.c 这个文件做个备份，也就是复制到同文件夹 test1 里面，新的文件命名为 b.c。然后在将 test1 文件夹中的 a.c 和 b.c 这两个文件都复制到文件夹 test2 中，操作

如图 2.6.2.9 所示

```
zuozhongkai@ubuntu:~/test1$ cp a.c b.c //拷贝a.c到本文件夹中，并重命名为b.c
zuozhongkai@ubuntu:~/test1$ ls //查看当前目录下的所有文件
a.c b.c
zuozhongkai@ubuntu:~/test1$ cp *.c ../test2 //拷贝a.c和b.c到文件夹test2中
zuozhongkai@ubuntu:~/test1$ cd ../test2 //进入上级目录中的test2文件夹
zuozhongkai@ubuntu:~/test2$ ls //查看test2文件夹下的文件
a.c b.c
```

图 2.6.2.9 拷贝文件

在图 2.6.2.9 中，我们添加了一些高级使用技巧，首先是拷贝 a.c 和 b.c 文件到 test2 文件夹中，我们使用了通配符“*”，“*.c”就表示 test1 下的所有以“.c”结尾的文件，也就是 a.c 和 b.c。“../test2”中的“../”表示上级目录，因此“../test2”就是上级目录下的 test2 文件夹。

上面都是文件复制，我们接下来学习一下文件夹复制，我们将 test2 文件夹复制到同目录下，新拷贝的文件夹命名为 test3，操作如图 2.6.2.10 所示：

```
zuozhongkai@ubuntu:~$ cp -rf test2/ test3/ //复制test2文件夹为test3文件夹
zuozhongkai@ubuntu:~$ ls //查看当前目录下的所有文件
examples.desktop test2 test.txt 公共的 视频 文档 音乐
test1 test3 tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ cd test3 //进入test3文件夹
zuozhongkai@ubuntu:~/test3$ ls //查看test3中的所有文件
a.c b.c
```

图 2.6.2.10 拷贝文件夹

6、文件移动命令—mv

有时候我们需要将一个文件或者文件夹移动到另外一个地方去，或者给一个文件或者文件夹进行重命名，这个时候我们就可以使用命令“mv”了，此命令格式如下：

```
mv [参数] [源地址] [目的地址]
```

主要参数描述如下：

- b 如果要覆盖文件的话覆盖前先进行备份。
- f 若目标文件或目录与现在的文件重复，直接覆盖目的文件或目录。
- I 在覆盖之前询问用户。

使用上面讲解“cp”命令的时候创建了三个文件夹，在上面创建的 test1 文件夹里面创建一个 c.c 文件，然后将 c.c 这个文件重命名为 d.c。最后将 d.c 这个文件移动到 test2 文件夹里面，操作如图 2.6.2.11 所示：

```
zuozhongkai@ubuntu:~$ cd test1 //进入test1文件夹
zuozhongkai@ubuntu:~/test1$ touch c.c //创建c.c文件
zuozhongkai@ubuntu:~/test1$ ls //查看当前目录下的所有文件
a.c b.c c.c
zuozhongkai@ubuntu:~/test1$ mv c.c d.c //移动c.c到当前目录下，并重命名为d.c
zuozhongkai@ubuntu:~/test1$ ls //查看当前目录下的所有文件
a.c b.c d.c
```

图 2.6.2.11 移动文件操作

我们再将 test1 中的 d.c 文件移动到 test2 文件夹里面，操作如图 2.6.2.12 所示：

```
zuozhongkai@ubuntu:~/test2$ ls //查看test2下所有文件
a.c b.c
zuozhongkai@ubuntu:~/test2$ cd ../test1 //进入test1文件夹中
zuozhongkai@ubuntu:~/test1$ ls //查看test1文件夹所有文件
a.c b.c d.c
zuozhongkai@ubuntu:~/test1$ mv d.c ../test2 //将d.c移动到test2中
zuozhongkai@ubuntu:~/test1$ ls ../test2 //查看test2下的所有文件
a.c b.c d.c
```

图 2.6.2.12 移动文件操作

2.6.3 文件压缩和解压缩

文件的压缩和解压缩是非常常见的操作，在 Windows 下我们有很多压缩和解压缩的工具，比如 zip、360 压缩等等。在 Ubuntu 下也有压缩工具，本节我们学习 Ubuntu 下图形化以及命令行这两种压缩和解压缩操作。

1、图形化压缩和解压缩

图形化压缩和解压缩和 Windows 下基本一样，在要压缩或者解压的文件上点击鼠标右键，然后选择要进行的操作，我们先讲解一下如何进行文件的压缩。首先找到要压缩的文件，然后在要压缩的文件上点击鼠标右键，选择“压缩”选项，如图 2.6.3.1 所示：



图 2.6.3.1 文件压缩

在图 2.6.3.1 中我们要对 test2 这个文件夹进行压缩，点击“压缩”以后会弹出图 2.6.3.2 所

示界面让选择压缩后的文件名和压缩格式。



图 2.6.3.2 压缩命名和格式选择

在图 2.6.3.2 中，设置好压缩以后的文件名，然后选择压缩格式，可选的压缩格式如图 2.6.3.3 所示：

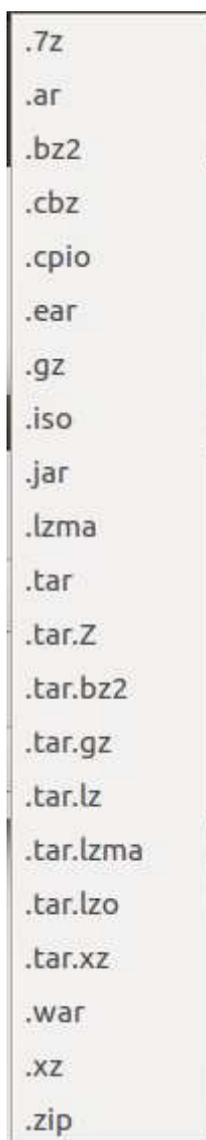


图 2.6.3.3 可选压缩格式

从图 2.6.3.3 中可以看出，可以选择的压缩格式还是有很多的，挑选一个格式进行压缩，比如我选择的“.zip”这个格式，压缩完成以后如图 2.6.3.4 所示：



图 2.6.3.4 压缩完成的文件

上面就是使用图形化进行文件压缩的过程，我们接下来对刚刚压缩的 `test2.zip` 进行解压缩，鼠标放到 `test2.zip` 上然后点击鼠标右键，选择“提取到此处”，如图 2.6.3.5 所示：



图 2.6.3.5 解压缩文件

点击图 2.6.3.5 中的“提取到此处”以后，系统就会自动进行解压缩，上面就是在 Ubuntu 中使用图形化工具进行文件的压缩和解压缩。

2、命令行进行文件的压缩和解压缩

上面我们学习了如何使用图形化工具在 Ubuntu 下进行文件的压缩和解压缩，本节我们学习如何使用命令行进行压缩和解压缩，我们后面的开发中所有涉及到压缩和解压缩的操作都是

在命令行下完成的。命令行下进行压缩和解压缩常用的命令有三个：zip、unzip 和 tar，我们依次来学习：

3、命令 zip

zip 命令看名字就知道是针对.zip 文件的，用于将一个或者多个文件压缩成一个.zip 结尾的文件，命令格式如下：

```
zip [参数] [压缩文件名.zip] [被压缩的文件]
```

主要参数函数如下：

- b<工作目录> 指定暂时存放文件的目录。
- d 从 zip 文件中删除一个文件。
- F 尝试修复已经损毁的压缩文件。
- g 将文件压缩入现有的压缩文件中，不需要新建压缩文件。
- h 帮助。
- j 只保存文件的名，不保存目录。
- m 压缩完成以后删除源文件。
- n<字尾符号> 不压缩特定扩展名的文件。
- q 不显示压缩命令执行过程。
- r 递归压缩，将指定目录下的所有文件和子目录一起压缩。
- v 显示指令执行过程。
- num 压缩率，为 1~9 的数值。

上面讲解了如何使用图形化压缩工具对文件夹 test2 进行压缩，这里我们使用命令“zip”对 test2 文件夹进行压缩，操作如图 2.6.3.6 所示：

```
zuozhongkai@ubuntu:~$ ls //显示当前目录下所有文件
examples.desktop test2 test.txt 公共的 视频 文档 音乐
test1 test3 tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ zip -rv test2.zip test2 //使用zip命令进行压缩
adding: test2/ (in=0) (out=0) (stored 0%)
adding: test2/d.c (in=0) (out=0) (stored 0%)
adding: test2/b.c (in=0) (out=0) (stored 0%)
adding: test2/a.c (in=0) (out=0) (stored 0%)
total bytes=0, compressed=0 -> 0% savings
zuozhongkai@ubuntu:~$ ls //显示压缩以后的当前目录下所有文件
examples.desktop test2 test3 tmp 模板 图片 下载 桌面
test1 test2.zip test.txt 公共的 视频 文档 音乐
```

图 2.6.3.6 使用 zip 进行文件压缩

图 2.6.3.6 就是使用 zip 命令进行 test2 文件夹的压缩，我们使用的命令如下：

```
zip -rv test2.zip test2
```

上述命令中，-rv 表示递归压缩并且显示压缩命令执行过程。

4、命令 unzip

unzip 命令用于对.zip 格式的压缩包进行解压，命令格式如下：

```
unzip [参数] [压缩文件名.zip]
```

主要参数如下：

- l 显示压缩文件内所包含的文件。
- t 检查压缩文件是否损坏，但不解压。
- v 显示命令显示的执行过程。
- Z 只显示压缩文件的注解。
- C 压缩文件中的文件名称区分大小写。

- j 不处理压缩文件中的原有目录路径。
- L 将压缩文件中的全部文件名改为小写。
- n 解压缩时不要覆盖原有文件。
- P<密码> 解压密码。
- q 静默执行，不显示任何信息。
- x<文件列表> 指定不要处理.zip 中的哪些文件。
- d<目录> 把压缩文件解到指定目录下。

对上面压缩的 test2.zip 文件使用 unzip 命令进行解压缩，操作如图 2.6.3.7 所示：

```

zuozhongkai@ubuntu:~$ rm -rf test2 //删除以前的test2文件夹，防止干扰解压过程
zuozhongkai@ubuntu:~$ ls //显示当前目录下所有文件
examples.desktop test2.zip test.txt 公共的 视频 文档 音乐
test1 test3 tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ unzip test2.zip //解压缩test2.zip文件
Archive: test2.zip
  creating: test2/
  extracting: test2/d.c
  extracting: test2/b.c
  extracting: test2/a.c
zuozhongkai@ubuntu:~$ ls //显示当前目录下所有文件
examples.desktop test2 test3 tmp 模板 图片 下载 桌面
test1 test2.zip test.txt 公共的 视频 文档 音乐

```

图 2.6.3.7 命令 unzip 使用演示

5、命令 tar

我们前面讲的 zip 和 unzip 这两个是命令只适用于.zip 格式的压缩和解压，其它压缩格式就用不了了，比如 Linux 下最常用的.bz2 和.gz 这两种压缩格式。其它格式的压缩和解压使用命令 tar，tar 将压缩和解压缩集合在一起，使用不同的参数即可，命令格式如下：

```
tar [参数] [压缩文件名] [被压缩文件名]
```

常用参数如下：

- c 创建新的压缩文件。
- C<目的目录> 切换到指定的目录。
- f<备份文件> 指定压缩文件。
- j 用 tar 生成压缩文件，然后用 bzip2 进行压缩。
- k 解开备份文件时，不覆盖已有的文件。
- m 还原文件时，不变更文件的更改时间。
- r 新增文件到已存在的备份文件的结尾部分。
- t 列出备份文件内容。
- v 显示指令执行过程。
- w 遭遇问题时先询问用户。
- x 从备份文件中释放文件，也就是解压缩文件。
- z 用 tar 生成压缩文件，用 gzip 压缩。
- Z 用 tar 生成压缩文件，用 compress 压缩。

使用 tar 命令来进行.zip 和.gz 格式的文件压缩，操作如图 2.6.3.8 所示：

```

zuozhongkai@ubuntu:~$ ls //显示当前目录下的所有文件
examples.desktop test2 test.txt 公共的 视频 文档 音乐
test1 test3 tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ tar -vcjf test1.tar.bz2 test1 //压缩为.bz2格式
test1/
test1/c.c
test1/b.c
test1/a.c
zuozhongkai@ubuntu:~$ ls //查看压缩后的文件是否存在
examples.desktop test1.tar.bz2 test3 tmp 模板 图片 下载 桌面
test1 test2 test.txt 公共的 视频 文档 音乐
zuozhongkai@ubuntu:~$ tar -vczf test1.tar.gz test1 //压缩为.gz格式
test1/
test1/c.c
test1/b.c
test1/a.c
zuozhongkai@ubuntu:~$ ls //查看压缩后的文件是否存在
examples.desktop test1.tar.bz2 test2 test.txt 公共的 视频 文档 音乐
test1 test1.tar.gz test3 tmp 模板 图片 下载 桌面

```

图 2.6.3.8 tar 命令进行压缩

在图 2.6.3.8 中，我们使用如下两个命令将 test1 文件夹压缩为.bz2 和.gz 这两个格式：

```

tar -vcjf test1.tar.bz2 test1
tar -vczf test1.tar.gz test1

```

在上面两行命令中，-vcjf 表示创建 bz2 格式的压缩文件，-vczf 表示创建.gz 格式的压缩文件。学习了如何使用 tar 命令来完成压缩，我们再来学习使用 tar 命令完成文件的解压，操作如图 2.6.3.9 所示：

```

zuozhongkai@ubuntu:~$ ls //显示当前目录下所有文件
examples.desktop test2.tar.gz test.txt 公共的 视频 文档 音乐
test1.tar.bz2 test3 tmp 模板 图片 下载 桌面
zuozhongkai@ubuntu:~$ tar -vxjf test1.tar.bz2 //解压缩.bz2格式文件
test1/
test1/c.c
test1/b.c
test1/a.c
zuozhongkai@ubuntu:~$ ls //检查test1.tar.bz2是否解压缩成功
examples.desktop test1.tar.bz2 test3 tmp 模板 图片 下载 桌面
test1 test2.tar.gz test.txt 公共的 视频 文档 音乐
zuozhongkai@ubuntu:~$ tar -vxzf test2.tar.gz //解压缩.gz格式文件
test2/
test2/d.c
test2/b.c
test2/a.c
zuozhongkai@ubuntu:~$ ls //检查test2.tar.gz是否解压缩成功
examples.desktop test2 test.txt 模板 文档 桌面
test1 test2.tar.gz tmp 视频 下载
test1.tar.bz2 test3 公共的 图片 音乐

```

图 2.6.3.9 tar 解压缩命令

图 2.6.3.9 中我们使用如下所示两行命令完成.bz2 和.gz 格式文件的解压缩：

```

tar -vxjf test1.tar.bz2
tar -vxzf test2.tar.gz

```

上述两行命令中，-vxjf 用来完成.bz2 格式压缩文件的解压，-vxzf 用来完成.gz 格式压缩文件的解压。关于 Ubuntu 下的命令行压缩和解压缩就讲解到这里，重点是 tar 命令，要熟练掌握使用 tar 命令来完成.bz2 和.gz 格式的文件压缩和解压缩。

2.6.4 文件查询和搜索

文件的查询和搜索也是最常用的操作，在嵌入式 Linux 开发中常常需要在 Linux 源码文件中查询某个文件是否存在，或者搜索哪些文件都调用了某个函数等等。本节我们就讲解两个最常用的文件查询和搜索命令：`find` 和 `grep`。

1、命令 `find`

`find` 命令用于在目录结构中查找文件，其命令格式如下：

```
find [路径] [参数] [关键字]
```

路径是要查找的目录路径，如果不写的话表示在当前目录下查找，关键字是文件名的一部分，主要参数如下：

-name<filename> 按照文件名称查找，查找与 `filename` 匹配的文件，可使用通配符。

-depth 从指定目录下的最深层的子目录开始查找。

-gid<群组识别码> 查找符合指定的群组识别码的文件或目录。

-group<群组名称> 查找符合指定的群组名称的文件或目录。

-size<文件大小> 查找符合指定文件大小的文件。

-type<文件类型> 查找符合指定文件类型的文件。

-user<拥有者名称> 查找符合指定的拥有者名称的文件或目录。

`find` 命令的参数有很多，常用的就这些，关于其它的参数大家可以自行上网查找，我们来看一下如何使用 `find` 命令进行文件搜索，我们搜索目录 `/etc` 中以“`vim`”开头的文件为例，操作如图 2.6.4.1 所示：

```
zuozhongkai@ubuntu:~$ find /etc/ -name vim*
/etc/vim
/etc/vim/vimrc
/etc/vim/vimrc.tiny
find: `/etc/cups/ssl': 权限不够
/etc/alternatives/vim
/etc/alternatives/vimdiff
find: `/etc/polkit-1/localauthority': 权限不够
find: `/etc/ssl/private': 权限不够
```

图 2.6.4.1 `find` 命令操作

从图 2.6.4.1 可以看出，在目录 `/etc` 下，包含以“`vim*`”开头的文件有 `/etc/vim`、`/etc/vim/vimrc` 等等，就不一一列出了。

2、命令 `grep`

`find` 命令用于在目录中搜索文件，我们有时候需要在文件中搜索一串关键字，`grep` 就是完成这个功能的，`grep` 命令用于查找包含指定关键字的文件，如果发现某个文件的内容包含所指定的关键字，`grep` 命令就会把包含指定关键字的这一行标记出来，`grep` 命令格式如下：

```
grep [参数] 关键字 文件列表
```

`grep` 命令一次只能查一个关键字，主要参数如下：

-b 在显示符合关键字的那一列前，标记处该列第 1 个字符的位编号。

-c 计算符合关键字的列数。

-d<进行动作> 当指定要查找的是目录而非文件时，必须使用此参数！否则 `grep` 指令将回报信息并停止搜索。

-i 忽略字符大小写。

-v 反转查找，只显示不匹配的行。

-r 在指定目录中递归查找。

比如我们在目录/usr 下递归查找包含字符“Ubuntu”的文件，操作如图 2.6.4.2 所示：

```
zuozhongkai@ubuntu:~$ grep -ir "Ubuntu" /usr
匹配到二进制文件 /usr/bin/fcitx-qimpanel-configtool
匹配到二进制文件 /usr/bin/nsupdate
/usr/bin/appport-bug:# Author: Martin Pitt <martin.pitt@ubuntu.com>
/usr/bin/appport-bug:# this so that confined applications using ubuntu-browsers.d
/ubuntu-integration
匹配到二进制文件 /usr/bin/locale
/usr/bin/fcitx-configtool: LXDE|Lubuntu)
匹配到二进制文件 /usr/bin/x86_64-linux-gnu-elfedit
```

图 2.6.4.2 命令 grep 演示

2.6.5 文件类型

这里的文件类型不是说这个文件是音乐文件还是文本文件，在用户根目录下使用命令“ls -l”来查看用户根目录下所有文件的详细信息，如图 2.6.5.1 所示：

```
zuozhongkai@ubuntu:~$ ls -l
总用量 72
-rw-r--r-- 1 zuozhongkai zuozhongkai 8980 12月 18 02:08 examples.desktop
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 24 21:56 test1
-rw-rw-r-- 1 zuozhongkai zuozhongkai 194 12月 24 21:58 test1.tar.bz2
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 23 01:00 test2
-rw-rw-r-- 1 zuozhongkai zuozhongkai 171 12月 24 22:09 test2.tar.gz
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 23 00:45 test3
-rw-rw-r-- 1 zuozhongkai zuozhongkai 68 12月 21 01:40 test.txt
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 19 21:41 tmp
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 公共的
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 模板
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 视频
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 图片
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 文档
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 下载
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 音乐
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 22 02:08 桌面
```

图 2.6.5.1 文件详细信息

在图 2.6.5.1 中，每个文件的详细信息占一行，每行最前面的符号标记了当前文件类型，比如 test1 的第一个字符是“d”，test1.tar.bz2 文件第一个字符是“-”。这些字符表示的文件类型如下：

- 普通文件，一些应用程序创建的，比如文档、图片、音乐等等。
- d 目录文件。
- c 字符设备文件，Linux 驱动里面的字符设备驱动，比如串口设备，音频设备等。
- b 块设备文件，存储设备驱动，比如硬盘，U 盘等。
- l 符号连接文件，相当于 Windows 下的快捷方式。
- s 套接字文件。
- p 管道文件，主要指 FIFO 文件。

我们后面学习 Linux 驱动开发的时候基本是在和字符设备文件和块设备文件打交道。

2.7 Linux 用户权限管理

2.7.1 Ubuntu 用户系统

Ubuntu 是一个多用户系统，我们可以给不同的使用者创建不同的用户账号，每个用户使用各自的账号登陆，使用用户账号的目的一是方便系统管理员管理，控制不同用户对系统的访问权限，另一方面是为用户提供安全性保护。

我们前面在安装 Ubuntu 系统的时候被要求创建一个账户，当我们创建好账号以后，系统会在目录/home 下以该用户名创建一个文件夹，所有与该用户有关的文件都会被存储在这个文件夹中。同样的，创建其它用户账号的时候也会在目录/home 下生成一个文件夹来存储该用户的文件，图 2.7.1.1 就是我的电脑上“zuozhongkai”这个账户的文件夹。



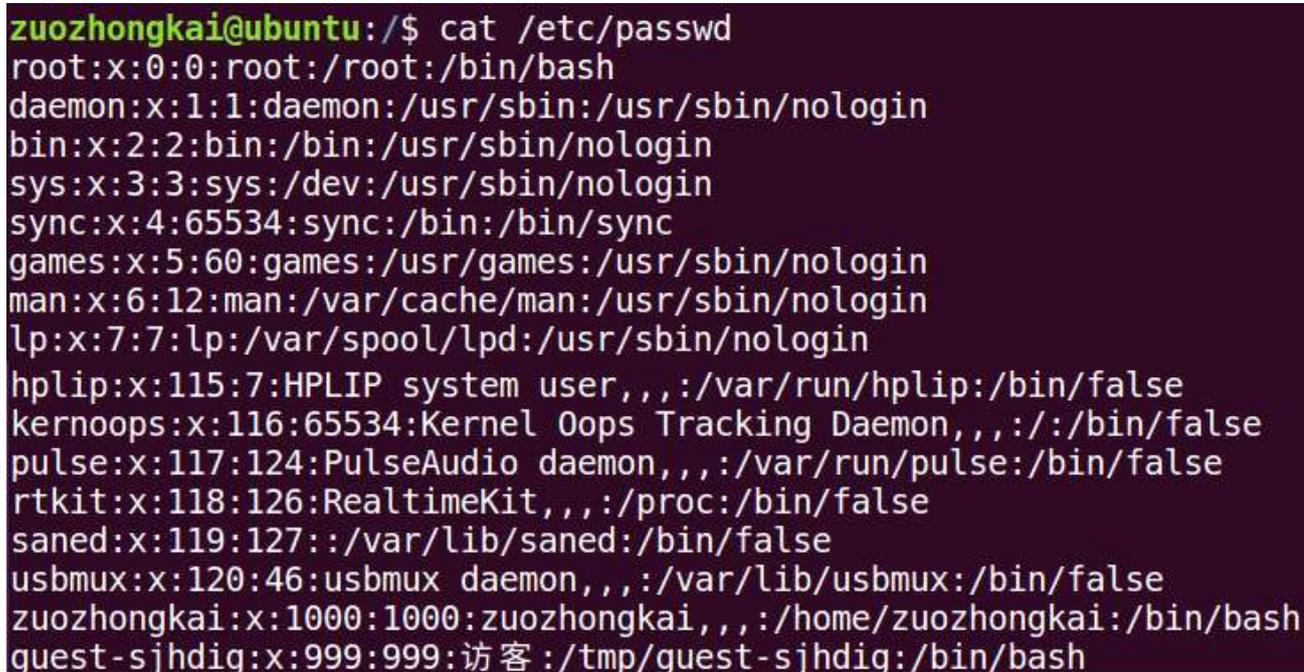
```
zuozhongkai@ubuntu:~$ ls
examples.desktop  tmp  公共  模板  视频  图片  文档  下载  音乐  桌面
```

图 2.7.1.1 用户账号根目录

装系统的时候创建的用户其权限比后面创建的用户大一点，但是没有 root 用户权限大，Ubuntu 下用户类型分为以下 3 类：

- 初次创建的用户，此用户可以完成比普通用户更多的功能。
- root 用户，系统管理员，系统中的玉皇大帝，拥有至高无上的权利。
- 普通用户，安装完操作系统以后被创建的用户。

以上三种用户，每个用户都有一个 ID 号，称为 UID，操作系统通过 UID 来识别是哪个用户，用户相关信息可以在文件/etc/passwd 中查看到，如图 2.7.1.2 所示：



```
zuozhongkai@ubuntu:/$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127:./var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
zuozhongkai:x:1000:1000:zuozhongkai,,,:/home/zuozhongkai:/bin/bash
guest-sjhdig:x:999:999:访客:/tmp/guest-sjhdig:/bin/bash
```

图 2.7.1.2 passwd 文件内容

从配置文件 passwd 中可以看到，每个用户名后面都有两个数字，比如用户“zuozhongkai”后面“1000:1000”，第一个数字是用户的 ID，另一个是用户的 GID，也就是用户组 ID。Ubuntu 里面每个用户都属于一个用户组里面，用户组就是一组有相同属性的用户集合。

2.7.2 权限管理

在使用 Windows 的时候我们很少接触到用户权限，最多就是打开某个软件出问题的时候会选择以“管理员身份”打开。Ubuntu 下我们会常跟用户权限打交道，权限就是用户对于系统资

源的使用限制情况，root 用户拥有最大的权限，可以为所欲为，装系统的时候创建的用户拥有 root 用户的部分权限，其它普通用户的权限最低。对于我们做嵌入式开发的人一般不关注用户的权限问题，因为嵌入式基本是单用户，做嵌入式开发重点关注的是文件的权限问题。

对于一个文件通常有三种权限：读(r)、写(w)和执行(x)，使用命令“ls -l”可以查看某个目录下所有文件的权限信息，如图 2.7.2.1 所示：

```
zuozhongkai@ubuntu:~$ ls -l
总用量 48
-rw-r--r-- 1 zuozhongkai zuozhongkai 8980 12月 18 02:08 examples.desktop
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 20:44 test.c
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 19 21:41 tmp
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 公共的
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 模板
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 视频
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 图片
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 文档
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 下载
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 音乐
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 22 02:08 桌面
```

图 2.7.2.1 文件权限信息

在图 2.7.2.1 中我们以文件 test.c 为例讲解，文件 test.c 文件信息如下：

```
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 20:44 test.c
```

其中“-rw-rw-r--”表示文件权限与用户和用户组之间的关系，第一位表示文件类型，上一小节已经说了。剩下的 9 位以 3 位为一组，分别表示文件拥有者的权限，文件拥有者所在用户组的权限以及其它用户权限。后面的“zuozhongkai zuozhongkai”分别代表文件拥有者(用户)和该用户所在的用户组，因此文件 test.c 的权限情况如下：

①、文件 test.c 的拥有者是用户 zuozhongkai，其对文件 test.c 的权限是“rw-”，也就是对该文件拥有读和写两种权限。

②、用户 zuozhongkai 所在的用户组也叫做 zuozhongkai，其组内用户对于文件 test.c 的权限是“rw-”，也是拥有读和写这两种权限。

③、其它用户对于文件 test.c 的权限是“r--”，也就是只读权限。

对于文件，可读权限表示可以打开查看文件内容，可写权限表示可以对文件进行修改，可执行权限就是可以运行此文件(如果是软件的话)。对于文件夹，拥有可读权限才可以使用命令 ls 查看文件夹中的内容，拥有可执行权限才能进入到文件夹内部。

如果某个用户对某个文件不具有相应的权限的话就不能进行相应的操作，比如根目录“/”下的文件只有 root 用户才有权限进行修改，如果以普通用户去修改的话就会提示没有权限。比如我们要在根目录“/”创建一个文件 mytest，使用命令“touch mytest”，结果如图 2.7.2.2 所示：

```
zuozhongkai@ubuntu:/$ touch mytest
touch: 无法创建 'mytest': 权限不够
```

图 2.7.2.2 创建文件

在图 2.7.2.2 中，我以用户“zuozhongkai”在根目录“/”创建文件 mytest，结果提示我无法创建“mytest”，因为权限不够，因为只有 root 用户才能在根目录“/”下创建文件。我们可以使用命令“sudo”命令暂时切换到 root 用户，这样就可以在根目录“/”下创建文件 mytest 了，如图 2.7.2.3 所示：

```

zuozhongkai@ubuntu:/$ sudo touch mytest
[sudo] zuozhongkai 的密码:
zuozhongkai@ubuntu:/$ ls
bin      dev      initrd.img      lib64      mnt      proc      sbin      sys      var
boot    etc      initrd.img.old  lost+found  mytest   root      snap      tmp      vmlinuz
cdrom   home    lib             media      opt      run      srv       usr      vmlinuz.old

```

图 2.7.2.3 使用 sudo 命令创建文件

在图 2.7.2.3 中我们使用命令“sudo”以后就可以在根目录“/”创建文件 mytest，在进行其它的操作的时候，遇到提示权限不够的时候都可以使用 sudo 命令暂时以 root 用户身份去执行。

上面我们讲了，文件的权限有三种：读(r)、写(w)和执行(x)，除了用 r、w 和 x 表示以外，我们也可以使用二进制数表示，三种权限就可以使用 3 位二进制数来表示，一种权限对应一个二进制位，如果该位为 1 就表示具备此权限，如果该位为 0 就表示不具备此权限，如表 2.7.2.1 所示：

字母	二进制	八进制
r	100	4
w	010	2
x	001	1

表 2.7.2.1 文件权限数字表示方法

如果做过单片机开发的话，就会发现和单片机里面的寄存器位一样，将三种权限 r、w 和 x 进行不同的组合，即可得到不同的二进制数和八进制数，3 位权限可以组出 8 种不同的权限组合，如表 2.7.2.2 所示：

权限	二进制数字	八进制数字
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

表 2.7.2.2 文件所有权限组合

表 2.7.2.2 中权限所对应的八进制数字就是每个权限对应的位相加，比如权限 rwx 就是 4+2+1=7。前面的文件 test.c 其权限为“rw-rw-r--”，因此其十进制表示就是：664。

另外我们也开始使用 a、u、g 和 o 表示文件的归属关系，用=、+和-表示文件权限的变化，如表 2.7.2.3 所示：

字母	意义
r	可读权限
w	可写权限
x	可执行权限
a	所有用户
u	归属用户
g	归属组
o	其它用户

=	具备权限
+	添加某权限
-	去除某权限

表 2.7.2.3 权限修改字母表示方式

对于文件 test.c，我们想要修改其归属用户(zuozhongkai)对其拥有可执行权限，那么就可以使用：u+x。如果希望设置归属用户及其所在的用户组都对其拥有可执行权限就可以使用：gu+x。

2.7.3 权限管理命令

我们也可以使用 Shell 来操作文件的权限管理，主要用到“chmod”和“chown”这两个命令，我们一个一个来看。

1、权限修改命令 chmod

命令“chmod”用于修改文件或者文件夹的权限，权限可以使用前面讲的数字表示也可以使用字母表示，命令格式如下：

```
chmod [参数] [文件名/目录名]
```

主要参数如下：

- c 效果类似“-v”参数，但仅回显更改的部分。
- f 不显示错误信息。
- R 递归处理，指定目录下的所有文件及其子文件目录一起处理。
- v 显示指令的执行过程。

我们先来学习以下如何使用命令“chmod”修改一个文件的权限，在用户根目录下创建一个文件 test，然后查看其默认权限，操作如图 2.7.3.1 所示：

```

zuozhongkai@ubuntu:~$ touch test
zuozhongkai@ubuntu:~$ ls -l
-rw-r--r-- 1 zuozhongkai zuozhongkai 8980 12月 18 02:08 examples.desktop
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 22:24 test
drwxrwxr-x 2 zuozhongkai zuozhongkai 4096 12月 19 21:41 tmp
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 公共的
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 模板
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 视频
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 图片
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 文档
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 18 02:21 下载
drwxr-xr-x 2 zuozhongkai zuozhongkai 4096 12月 22 02:08 桌面

```

图 2.7.3.1 创建文件 test

在图 2.7.3.1 中我们创建了一个文件：test，这个文件的默认权限为“rw-rw-r--”，我们将其权限改为“rwxrw-rw”，对应数字就是 766，操作如下：

```

zuozhongkai@ubuntu:~$ ls -l test //查看文件test权限
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 22:24 test
zuozhongkai@ubuntu:~$ chmod 766 test //修改文件test的权限为766
zuozhongkai@ubuntu:~$ ls -l test //查看修改以后的文件test权限
-rwxrw-rw- 1 zuozhongkai zuozhongkai 0 12月 25 22:24 test

```

图 2.7.3.2 修改权限

在图 2.7.3.2 中，我们修改文件 test 的权限为 766，修改完成以后的 test 文件权限为“rwxrw

-rw-”，和我们设置的一样，说明权限修改成功。

上面我们是通过数字来修改权限的，我们接下来使用字母来修改权限，操作如图 2.7.3.3 所示：

```
zuozhongkai@ubuntu:~$ touch a.c //创建文件a.c
zuozhongkai@ubuntu:~$ ls -l a.c //显示文件a.c的权限
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 22:33 a.c
zuozhongkai@ubuntu:~$ chmod u+x a.c //给文件a.c的归属用户添加可执行权限
zuozhongkai@ubuntu:~$ ls -l a.c //显示修改权限后的文件a.c
-rwxrw-r-- 1 zuozhongkai zuozhongkai 0 12月 25 22:33 a.c
```

图 2.7.3.3 使用字母修改文件权限

上面两个例子都是修改文件的权限，接下来我们修改文件夹的权限，新建一个 test 文件夹，在文件夹 test 里面创建 a.c、b.c 和 c.c 三个文件，如图 2.7.3.4 所示：

```
zuozhongkai@ubuntu:~$ ls -l test/
总用量 0
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 26 00:20 a.c
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 26 00:20 b.c
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 26 00:20 c.c
```

图 2.7.3.4 test 文件夹

在图 2.7.3.4 中 test 文件夹下的文件 a.c、b.c 和 c.c 的权限均为“rw-rw-r--”，我们将 test 文件夹下的所有文件权限都改为“rwxrwxrwx”，也就是数字 777，操作如图 2.7.3.5 所示：

```
zuozhongkai@ubuntu:~$ chmod -R 777 test/ //递归修改文件权限
zuozhongkai@ubuntu:~$ ls -l test/ //查看修改权限以后的文件
总用量 0
-rwxrwxrwx 1 zuozhongkai zuozhongkai 0 12月 26 00:20 a.c
-rwxrwxrwx 1 zuozhongkai zuozhongkai 0 12月 26 00:20 b.c
-rwxrwxrwx 1 zuozhongkai zuozhongkai 0 12月 26 00:20 c.c
```

图 2.7.3.5 递归修改文件夹权限

2、文件归属者修改命令 chown

命令 chown 用来修改某个文件或者目录的归属者用户或者用户组，命令格式如下：

```
chown [参数] [用户名.<组名>] [文件名/目录]
```

其中[用户名.<组名>]表示要将文件或者目录改为哪一个用户或者用户组，用户名和组名用“.”隔开，其中用户名和组名中的任何一个都可以省略，命令主要参数如下：

- c 效果同-v 类似，但仅显示更改的部分。
- f 不显示错误信息。
- h 只对符号连接的文件做修改，不改动其它任何相关的文件。
- R 递归处理，将指定的目录下的所有文件和子目录一起处理。
- v 显示处理过程。

在用户根目录下创建一个 test 文件，查看其文件夹所属用户和用户组，如图 2.7.3.6 所示：

```
zuozhongkai@ubuntu:~$ touch test
zuozhongkai@ubuntu:~$ ls -l test
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 26 00:36 test
```

图 2.7.3.6 test 文件信息查询

从图 2.7.3.6 中可以看出，文件 test 的归属用户为 zuozhongkai，所属的用户组为 zuozhongkai，将文件 test 归属用户改为 root 用户，所属的用户组也改为 root，操作如图 2.7.3.7 所示：

```
zuozhongkai@ubuntu:~$ ls -l test //显示文件test的归属用户和归属组
-rw-rw-r-- 1 zuozhongkai zuozhongkai 0 12月 26 00:45 test
zuozhongkai@ubuntu:~$ sudo chown root.root test //修改文件test的归属用户和归属组
zuozhongkai@ubuntu:~$ ls -l test //查看修改以后的文件test归属用户和归属组
-rw-rw-r-- 1 root root 0 12月 26 00:45 test
```

图 2.7.3.7 修改文件归属用户和归属组

命令 shown 同样也可以递归处理来修改文件夹的归属用户和用户组，用法和命令 chown 一样，这里就不演示了。

2.8 Linux 磁盘管理

2.8.1 Linux 磁盘管理基本概念

Linux 的磁盘管理体系和 Windows 有很大的区别，在 Windows 下经常会遇到“分区”这个概念，在 Linux 中一般不叫“分区”而叫“挂载点”。“挂载点”就是将一个硬盘的一部分做成文件夹的形式，这个文件夹的名字就是“挂载点”，不管在哪个发行版的 Linux 中，用户是绝对看不到 C 盘、D 盘这样的概念的，只能看到以文件夹形式存在的“挂载点”。

文件/etc/fstab 详细的记录了 Ubuntu 中硬盘分区的情况，如图 2.8.1.1 所示：

```
zuozhongkai@ubuntu:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=fcdae2a0-05f1-48fa-a6c8-b4efff06a180 / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=c9eb5d28-50fe-4e35-951d-ea8a1c4b7810 none swap sw 0 0
```

图 2.8.1.1 文件 fstab

在图 2.8.1.1 中有一行“/ was on /dev/sda1 during installation”，意思是根目录“/”是在/dev/sda1 上的，其中“/”是挂载点，“/dev/sda1”就是我们装 Ubuntu 系统的硬盘。由于我们的系统是安装在虚拟机中的，因此图 2.8.1.1 没有出现实际的硬盘。可以通过如下命令查看当前系统中的磁盘：

```
ls /dev/sd*
```

上述命令就是打印出所有以/dev/sd 开头的设备文件，如图 2.8.1.2 所示：

```
zuozhongkai@ubuntu:~$ ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5
```

图 2.8.1.2 查看硬盘设备文件

在图 2.8.1.2 中有四个磁盘设备文件，其中 sd 表示是 SATA 硬盘或者其它外部设备，最后面的数字表示该硬盘上的第 n 个分区，比如/dev/sda1 就表示磁盘 sda 上的第一个分区。图 2.8.1.2 中都是以/dev/sda 开头的，说明当前只有一个硬盘。如果再插上 U 盘、SD 卡啥的就可能会出现/dev/sdb、/dev/sdc 等等。如果你的 U 盘有两个分区那么可能会出现/dev/sdb1、dev/sdb2 这样的设备文件。比如我现在插入我的 U 盘，插入 U 盘会提示 U 盘是接到主机还是虚拟机，如图 2.8.1.3 所示：



图 2.8.1.3 U 盘连接选择

设置好图 2.8.1.3 以后，点击“确定”按钮 U 盘就会自动连接到虚拟机中，也就是连接到 Ubuntu 系统中，我们再次使用命令“ls /dev/sd*”来查看当前的“/dev/sd*”设备文件，如图 2.8.1.4 所示：

```
zuozhongkai@ubuntu:~$ ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5 /dev/sdb /dev/sdb1
```

图 2.8.1.4 插入 U 盘后的设备文件

从图 2.8.1.4 可以看出，相比图 2.8.1.2 多了 /dev/sdb 和 /dev/sdb1 这两个文件，其中 /dev/sdb 就是 U 盘文件， /dev/sdb1 表示 U 盘的第一个分区，因为我的 U 盘就一个分区。

2.8.2 磁盘管理命令

本节我们学习一下跟磁盘操作有关的命令，这些命令如下：

1、磁盘分区命令 fdisk

如果要对某个磁盘进行分区，可以使用命令 fdisk，命令格如下：

```
fdisk [参数]
```

主要参数如下：

-b<分区大小> 指定每个分区的大小。

-l 列出指定设备的分区表。

-s<分区编号> 将指定的分区大小输出到标准的输出上，单位为块。

-u 搭配“-l”参数，会用分区数目取代柱面数目，来表示每个分区的起始地址。

比如我要对 U 盘进行分区，千万不要对自己装 Ubuntu 系统进行分区!!! 可以使用如下命令：

```
sudo fdisk /dev/sdb
```

结果如图 2.8.2.1 所示:

```
zuozhongkai@ubuntu:~$ sudo fdisk /dev/sdb
[sudo] zuozhongkai 的密码:

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

命令(输入 m 获取帮助): █
```

图 2.8.2.1 U 盘分区界面

在图 2.8.2.1 中提示我们输入“m”可以查看帮助，因为 fdisk 还有一些子命令，通过输入“m”可以查看都有哪些子命令，如图 2.8.2.2 所示:

```
命令(输入 m 获取帮助): m

Help:

DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table
```

图 2.8.2.2 fdisk 命令的子命令

图 2.8.2.2 中常用的命令如下:

- p** 显示现有的分区
- n** 建立新分区

- t** 更改分区类型
- d** 删除现有的分区
- a** 更改分区启动标志
- w** 对分区的更改写入到硬盘或者存储器中。
- q** 不保存退出。

由于我的 U 盘里面还有一些重要的文件，所以现在不能进行分区，这里现在就不演示 `fdisk` 的分区操作了，后面我们讲解裸机例程的时候需要将可执行的 `bin` 文件烧写到 SD 卡中，烧写到 SD 卡之前需要对 SD 卡进行分区，到时候在详细讲解如何使用 `fdisk` 命令对磁盘进行分区。

2、格式化命令 `mkfs`

使用命令 `fdisk` 创建好一个分区以后，我们需要对其格式化，也就是在这个分区上创建一个文件系统，Linux 下的格式化命令为 `mkfs`，命令格式如下：

```
mkfs [参数] [-t 文件系统类型] [分区名称]
```

主要参数如下：

- fs** 指定建立文件系统时的参数
- V** 显示版本信息和简要的使用方法。
- v** 显示版本信息和详细的使用方法。

比如我们要格式化 U 盘的分区 `/dev/sdb1` 为 FAT 格式，那么就可以使用如下命令：

```
mkfs -t vfat /dev/sdb1
```

3、挂载分区命令 `mount`

我们创建好分区并且格式化以后肯定是要使用硬盘或者 U 盘的，那么如何访问磁盘呢？比如我的 U 盘就一个分区，为 `/dev/sdb1`，如果直接打开文件 `/dev/sdb1` 会发现根本就不是我们要的结果。我们需要将 `/dev/sdb1` 这个分区挂载到一个文件夹中，然后通过这个文件访问 U 盘，磁盘挂载命令为 `mount`，命令格式如下：

```
mount [参数] -t [类型] [设备名称] [目的文件夹]
```

命令主要参数有：

- V** 显示程序版本。
- h** 显示辅助信息。
- v** 显示执行过程详细信息。
- o ro** 只读模式挂载。
- o rw** 读写模式挂载。
- s-r** 等于 `-o ro`。
- w** 等于 `-o rw`。

挂载点是一个文件夹，因此在挂载之前先要创建一个文件夹，一般我们把挂载点放到“`/mnt`”目录下，在“`/mnt`”下创建一个 `tmp` 文件夹，然后将 U 盘的 `/dev/sdb1` 分区挂载到 `/mnt/tmp` 文件夹里面，操作如图 2.8.2.3 所示：

```

zuozhongkai@ubuntu:~$ ls /mnt //查看/mnt目录下的所有文件
zuozhongkai@ubuntu:~$
zuozhongkai@ubuntu:~$ sudo mkdir /mnt/tmp //创建文件夹/mnt/tmp
[sudo] zuozhongkai 的密码:
zuozhongkai@ubuntu:~$ ls /mnt //查看/mnt目录下的所有文件
tmp
zuozhongkai@ubuntu:~$ sudo mount -t vfat /dev/sdb1 /mnt/tmp //挂载U盘到/mnt/tmp中
zuozhongkai@ubuntu:~$ ls /mnt/tmp //查看/mnt/tmp中的文件,也就是U盘里面的文件
AD IMX6UL_ZERO_V1.0.zip 开发板光盘
ARM裸机与嵌入式Linux驱动开发V1.0.docx System Volume Information

```

图 2.8.2.3 挂载 U 盘

在图 2.8.2.3 中我们将 U 盘以 fat 格式挂载到目录/mnt/tmp 中，然后我们就可以通过访问/mnt/tmp 来访问 U 盘了。

4、卸载命令 umount

当我们不再需要访问已经挂载的 U 盘，可以通过 umount 将其从卸载点卸除，命令格式如下：

```

umount [参数] -t [文件系统类型] [设备名称]
-a  卸载/etc/mntab 中的所有文件系统。
-h  显示帮助。
-n  卸载时不要将信息存入到/etc/mntab 文件中
-r  如果无法成功卸载，则尝试以只读的方式重新挂载。
-t<文件系统类型>  仅卸载选项中指定的文件系统。
-v  显示执行过程。

```

上面我们将 U 盘挂载到了文件夹/mnt/tmp 里面，这里我们使用命令 umount 将其卸载掉，操作如图 2.8.2.4 所示：

```

zuozhongkai@ubuntu:~$ ls /mnt/tmp //查看卸载之前的文件夹/mnt/tmp
AD IMX6UL_ZERO_V1.0.zip 开发板光盘
ARM裸机与嵌入式Linux驱动开发V1.0.docx System Volume Information
zuozhongkai@ubuntu:~$ sudo umount -t vfat /dev/sdb1 //卸载U盘的/dev/sdb1分区
zuozhongkai@ubuntu:~$ ls /mnt/tmp //查看卸载以后的/mnt/tmp文件夹
zuozhongkai@ubuntu:~$
zuozhongkai@ubuntu:~$

```

图 2.8.2.4 卸载 U 盘

在图 2.8.2.4 中，我们使用命令 umount 卸载了 U 盘，卸载以后当我们再去访问文件夹/mnt/tmp 的时候发现里面没有任何文件了，说明我们卸载成功了。