

# Unix/Linux体系及编程

## Chap6 Linux文件系统

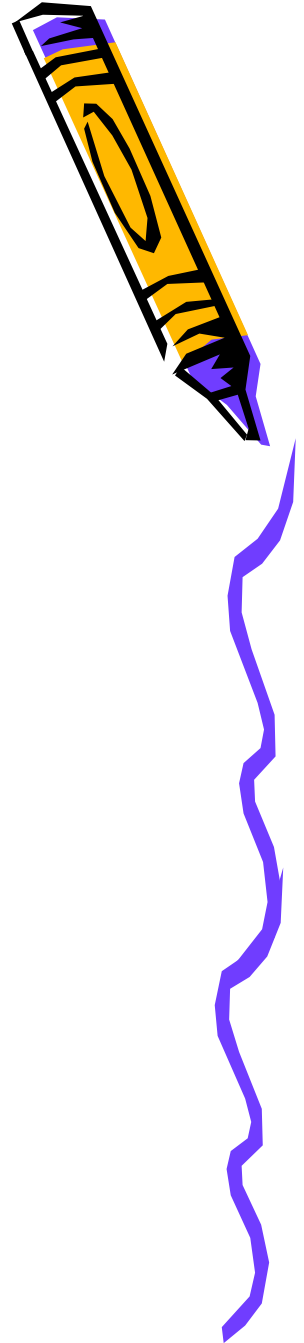
刘朝斌

zhbliu@gmail.com



# 主要内容

- 什么是文件系统
- Linux文件系统的结构
- Linux文件的类型和访问权限
- 如何安装和卸载文件系统
- 如何创建文件系统
- 如何进行ext之间文件系统的转换
- 交换分区和交换文件
- 关于文件的命令

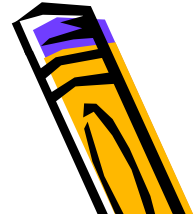


## 6.1 Linux文件系统

- **文件系统**：任何一个操作系统无论简单与否，都必须提供管理存放在这些存储设备上的程序和数据的手段。完成这一任务的内核程序就是操作系统的文件管理系统，简称文件系统。
- **“文件”**：是存放一切数据信息的仓库。

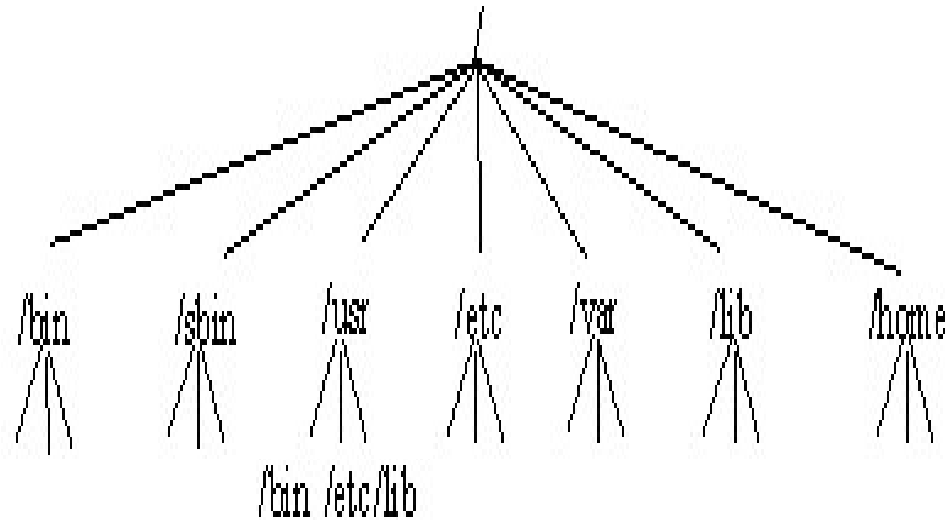


## 6.1.1 Linux与DOS文件系统的区别



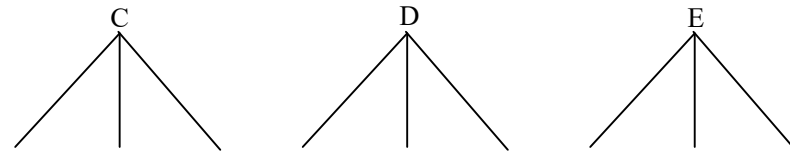
### ➤ Linux文件系统

- ✓ 单个目录树的结构
- ✓ 根是根目录“/”
- ✓ 往下连接各个分支



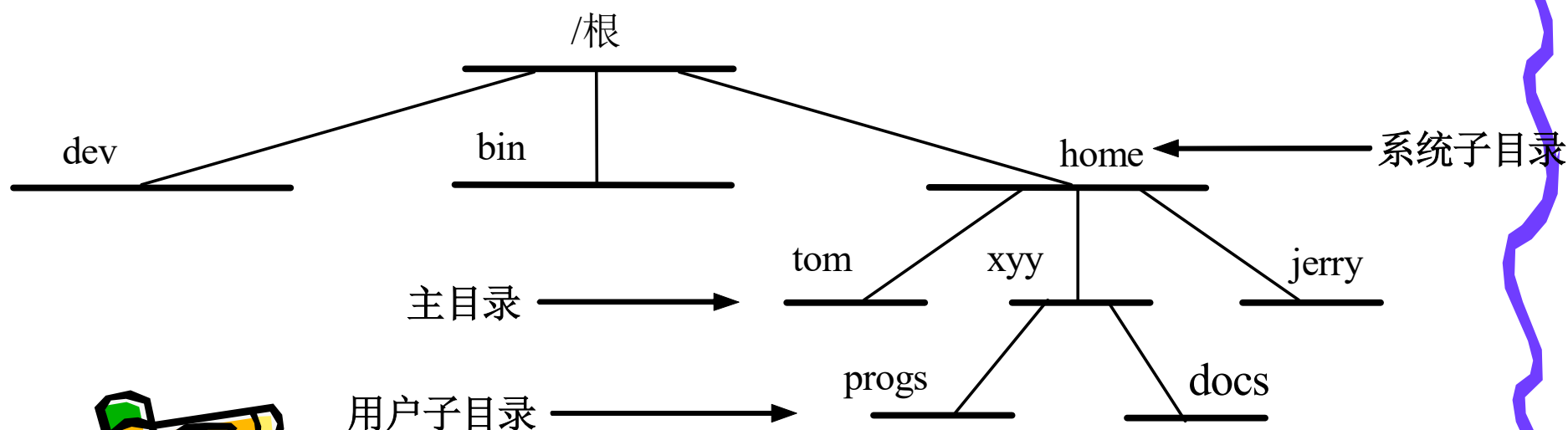
### ➤ DOS文件系统

- ✓ 每个分区为树根
- ✓ 多个分区，形成了多个树并列



## 6.1.2 Linux文件系统结构

- 用户的系统子目录
- 用户的登录子目录
- 路径名



**/bin**：提供用户使用的基本命令，存放二进制命令，不允许关联到独立分区，OS启动会用到里面的程序。

**/dev**：设备（device）文件目录，存放linux系统下的设备文件，访问该目录下某个文件，相当于访问某个设备，存放连接到计算机上的设备（终端、磁盘驱动器、光驱及网卡等）的对应文件（b 随机访问，c 线性访问）

**/proc**：用于输出内核与进程信息相关的虚拟文件系统，目录中的数据都在内存中，如系统核心，外部设备，网络状态，由于数据都存放于内存中，所以不占用磁盘空间

## 6.2 Linux的文件

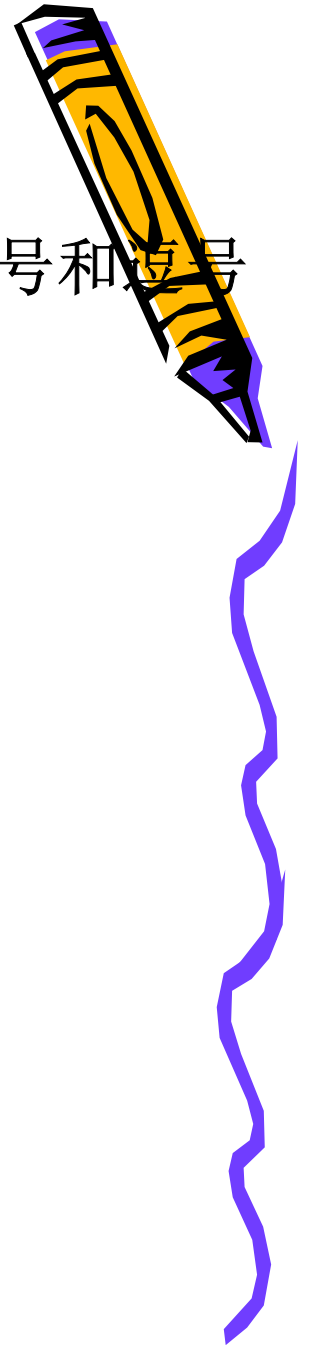
### ➤ 文件名

- ✓ 字母字符、下划线和数字组成，也可以加上句号和逗号
- ✓ 文件名和扩展名之间用句点隔开

### ➤ 文件类型

- ✓ 物理格式：字节流 (**byte stream**)
- ✓ 类型：**file** 命令查看

- - 普通文件：纯文本；二进制；数据格式
- **d** 目录文件
- **l** 连接文件
- **b** 块设备特殊文件
- **c** 字符设备特殊文件
- **s** 套接字文件 (**socket**)



## 6.2.3 文件权限

### ➤ 文件的详细信息

文件的权限

文件上次修改  
的日期和时间

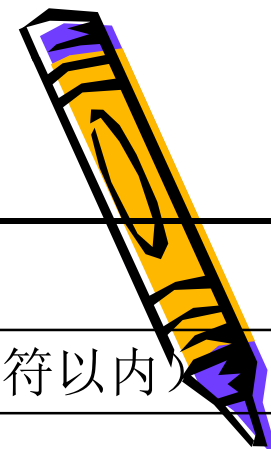
文件类型	链接个数	文件属主名	用户分组名	文件长度	文件上次修改的日期和时间	文件名
-rw-r--r--	1	chris	weather	207	Jul 20 11:55	mydata

- 三个安全级别：所有者级别、组访问级别、其他用户访问级别
- 文件和子目录：读（**Read**）、写（**Write**）、执行（**eXecute**）
- 访问权的分配：文件属主（**owner/user**）、用户分组（**group**）、其他人（**other**）





## 6.3 ext3/4文件系统

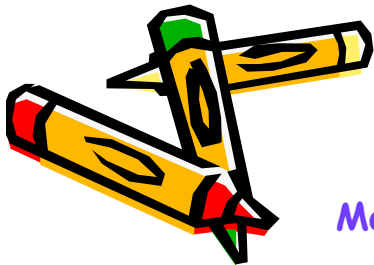
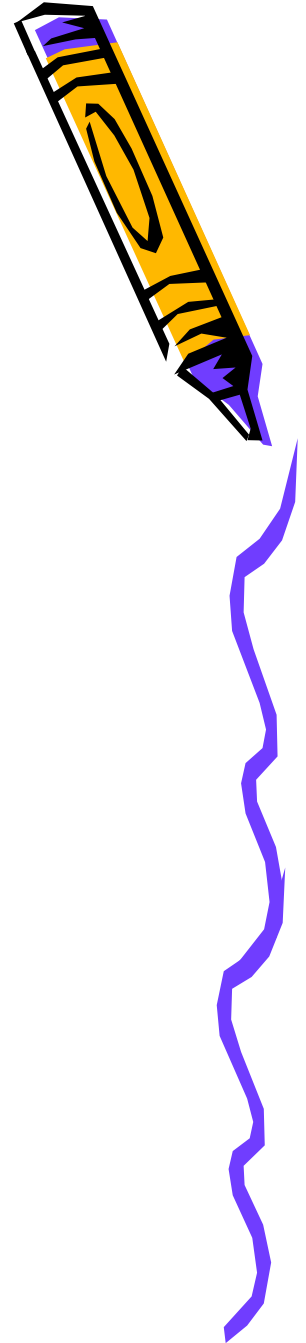


类 型	说 明
Minix	Minix文件系统（文件名的长度被限制在30个字符以内）
ext	Linux文件系统的早期版本，现已不再使用
ext2(3/4)	标准Linux文件系统，支持长文件名和大文件尺寸
xiaf	Xiaf文件系统
msdos	对应于MS-DOS分区的文件系统（16位）
vfat	对应于Windows分区的文件系统（32位）
proc	由操作系统使用，对应于进程
nfs	对应于来自远程系统挂装分区的NFS文件系统
swap	Linux的swap数据交换分区或者swap文件
iso9660	挂装光盘的文件系统



## 6.3.2 ext4 文件系统

- 可用性
- 数据完好性
- 速度
- 简易转换



## 6.4 安装和卸载文件系统

- 使用**mount**命令挂装文件系统

- ✓ 语法:

*mount -a[fnrsvm][-t vfstype]*

*mount [-fnrsvw][-o options..]device dir*

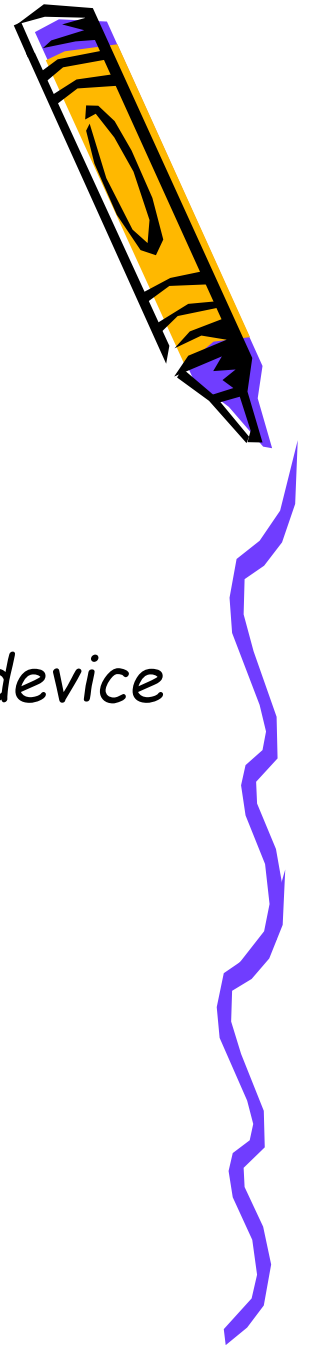
*mount[-fnrsvw][-t vfstype][-o options]device  
dir*

- 使用**umount**命令卸载文件系统

- ✓ 语法:

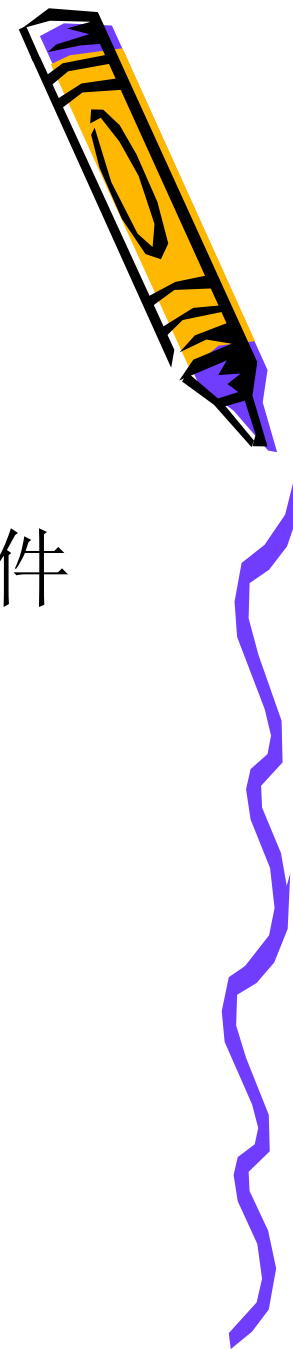
*umount device or dir*

- 手工编辑**/etc/fstab**文件



## 6.5 创建一个ext4文件系统

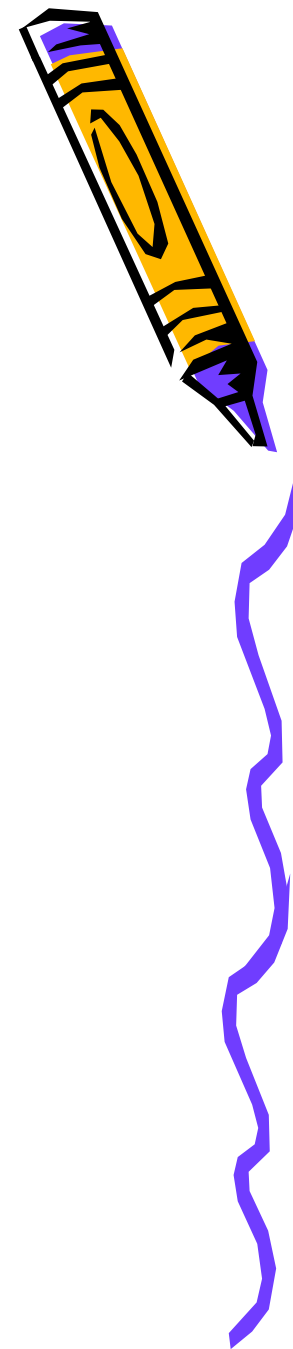
- 使用**parted** 或**fdisk**来创建分区；
- 使用**mkfs** 来把分区格式化为**ext3**文件系统；
- 使用**e2label**给分区标签；
- 创建挂载点；
- 把分区添加到**/etc/fstab**文件中。



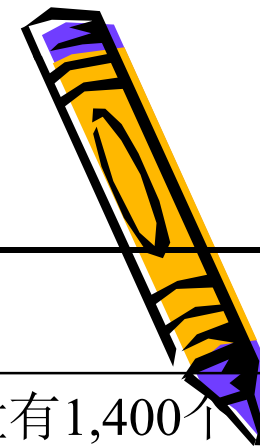
## 6.5.1 使用parted来创建磁盘分区

parted - a partition manipulation program

- 查看分区表
- 创建分区
- 制作分区
- 格式化分区
- 给分区注明标签
- 创建挂载点
- 添加到/etc/fstab



## 6.5.2 使用mkfs格式化分区



命 令	说 明
Blocks	文件系统使用的存储块数目。一张1.44MB软盘有1,400个存储块
-t file-system-type	指定格式化操作时使用的文件系统类型
fs -options	指定的文件系统类型的参数
-V	命令进展注释状态
-v	通知文件系统建立程序mkfs
-c	执行时要显示它采取的每个动作
-l filename	在对分区进行格式化操作之前先检查其上的坏数



## 6.5.3 删除分区

- (1) 首先启动**parted**程序，执行**parted /dev/hda**。
- (2) 查看当前的分区表来判定要删除的分区的首要号码：**print**。
- (3) 使用**rm**来删除分区。例如，要删除次要号码为3的分区：**rm 3**。只要你一按[Enter]键，改变就会发生，因此在执行前请检查一下命令。
- (4) 删除了分区后，使用**print**命令来确认分区在分区表中已被删除。你还应该查看输出来确定内核知道分区已被删除：**cat /proc/partitions**。 **/dev/sd?**
- (5) 最后一步是把它从**/etc/fstab**文件中删除。找到和已被删除的分区相应的行，然后从文件中删除它。

```
buyhorse@ubuntu:~/apue$ cat /proc/partitions
major minor  #blocks  name
    11         0    1048575  sr0
     8         0   583983104  sda
     8         1   273436672  sda1
     8         2    195584  sda2
     8         3   11718656  sda3
```



## 6.5.4 重新划分分区大小

- 引导入救援模式（或卸载设备上的所有分区并关闭设备上的交换空间）。
- 启动**parted**，**/dev/hda** 是要在其中重新划分分区大小的设备：**parted /dev/hda**
- 查看当前的分区表来判定要重划大小的分区的次要号码以及它的起止点
- 使用**resize**命令，然后跟随分区的次要号码，以**MB**为单位的起始点和终止点。
- 使用**print**命令来确认分区已被正确地重新划分了大小，并且具备正确的分区类型和文件系统类型。
- 在正常模式下重新引导了系统后，使用**df**命令来确定分区已被挂载

✓ **df -lh** (report file system disk space usage)





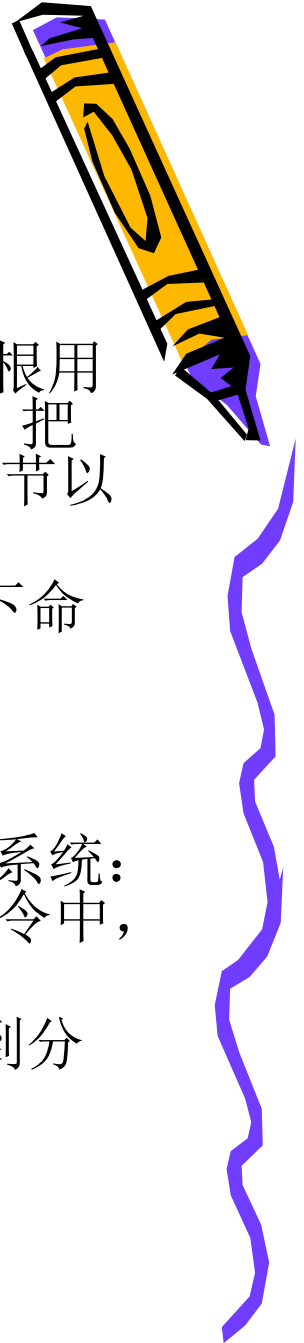
## 6.6 ext文件系统之间的转换



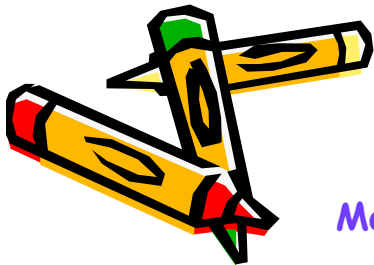
- (1) 要把 **ext2** 文件系统转换成 **ext3**，登录为根用户后键入：  
`/sbin/tune2fs -j /dev/hdbX`，具体命令中把 `/dev/hdb` 替换成设备名，把 **X** 替换成分区号码。
- (2) 命令执行完毕后，请确定把 `/etc/fstab` 文件中的 **ext2** 文件系统改成 **ext3** 文件系统。



## 6.7 还原到某个ext文件系统



- (1) 要还原分区，你必须首先卸载分区。方法是登录为根用户，然后键入：`umount /dev/hdbX`，在以上命令中，把 `/dev/hdb` 替换成设备名称，把 `X` 替换成分区号码。本节以后的示例命令将会使用 `hdb1` 来代表设备和分区。
- (2) 把文件系统类型改回 `ext2`，以根用户身份键入以下命令：`/sbin/tune2fs -O ^has_journal /dev/hdb1`。
- (3) 以根用户身份键入以下命令来检查分区的错误：`/sbin/e2fsck -y /dev/hdb1`。
- (4) 通过键入以下命令来把分区重新挂载为 `ext2` 文件系统：`mount -t ext2 /dev/hdb1 /mount/point`，在以上命令中，把 `/mount/point` 替换成分区的挂载点。
- (5) 最后是删除根目录下的 `.journal` 文件。方法是转换到分区的挂载目录中，然后键入：`rm -f .journal`



## 6.8 交换文件与交换分区

交换分区的优点（**vs MS**）：

1. 将整个分区用作交换，与用户文件系统分开。  
用户对交换分区不可见，这就不会影响交换分区的工作。
2. 既然是一个分区，那么它的磁盘空间是连续的。  
如果是交换文件则必须要划分出一个连续的磁盘空间，交换分区则省了这个问题。
3. 交换分区的磁盘空间比交换文件大得多。



## 6.8.1 交换文件

➤ 创建交换文件命令：**dd**

dd - convert and copy a file

➤ 语法：

`dd if=sourcefile of=destfile bs=BYTE count=BLOCKS`

✓ `dd if=/dev/hdx of=/dev/hdy` 整盘备份

✓ `dd if=/dev/hdx of=/path/to/image count=1 bs=512` 备份 MBR

✓ `dd if=/dev/urandom of=/dev/hda1` 销毁硬盘数据

✓ `dd if=/var/test of=/dev/null bs=8k` 测试纯读

✓ `dd if=/dev/zero of=/var/test bs=8k count=10000` 测试纯写



## 6.8.2 交换分区

➤ 使用**fdisk**命令生成一个交换分区，将分区的标志号设置为83，

➤ 记录分区的块数

➤ 设为系统的交换分区

fdisk - manipulate disk partition table

➤ 激活

➤ **sudo fdisk -l /dev/sda2**

➤ **sudo fdisk -l**

➤ **sudo sfdisk -g //show geometry infor**

➤ **sudo sfdisk -T // Print all supported types**



## 6.8.3 关闭交换区

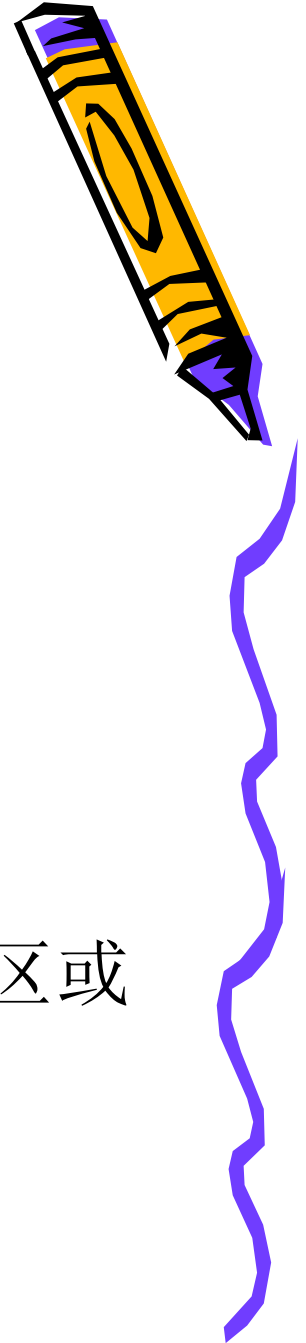
- 关闭交换分区或交换文件：

***swapoff /extra-swap***

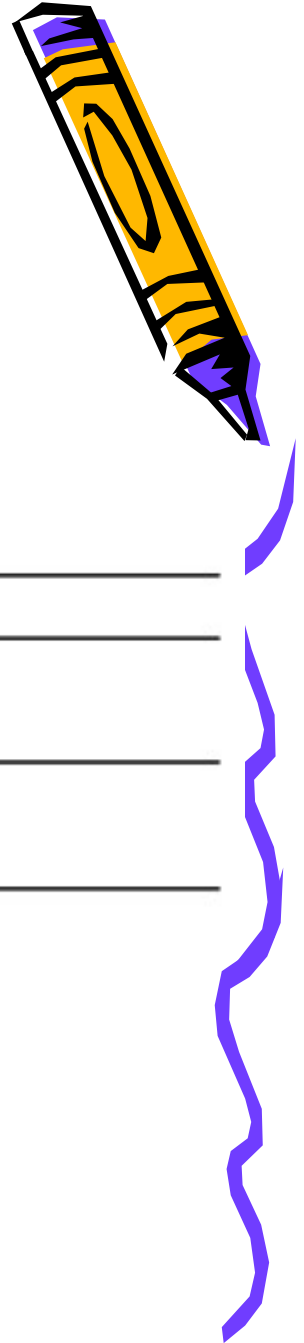
***swapoff /hda4***

- 把交换空间从某处移到另一处

- 首先删除交换空间
- 再遵循添加交换空间的说明，建立交换区或交换文件



# FHS: Filesystem Hierarchy Standard



➤ <http://www.pathname.com/fhs/>

	shareable	unshareable
static	/usr	/etc
	/opt	/boot
variable	/var/mail	/var/run
	/var/spool/news	/var/lock



# 根目录结构 /

➤ bin

➤ sbin

系统 ( 超级用户的 ) 可  
执行文件

➤ home

/home:

用户的主目录, 在 Linux 中, 每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名的, 如上图中的 alice、bob 和 eve。

➤ dev

➤ usr ( Unix Software Resource/Release )

➤ proc

➤ ...





### **/proc:**

proc 是 Processes(进程) 的缩写, /proc 是一种伪文件系统 (也即虚拟文件系统), 存储的是当前内核运行状态的一系列特殊文件, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。

这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件, 比如可以通过下面的命令来屏蔽主机的ping命令, 使别人无法ping你的机器:

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

### **/sbin:**

s 就是 Super User 的意思, 是 Superuser Binaries (超级用户的二进制文件) 的缩写, 这里存放的是系统管理员使用的系统管理程序。

### **/usr:**

usr 是 unix shared resources(共享资源) 的缩写, 这是一个非常重要的目录, 用户的很多应用程序和文件都放在这个目录下, 类似于 windows 下的 program files 目录。

## 6.9 其它文件相关命令 - **wc**

wc - print newline, word, and byte counts for each file

➤ `wc -l 1.txt`

➤ `wc -w 1.txt`

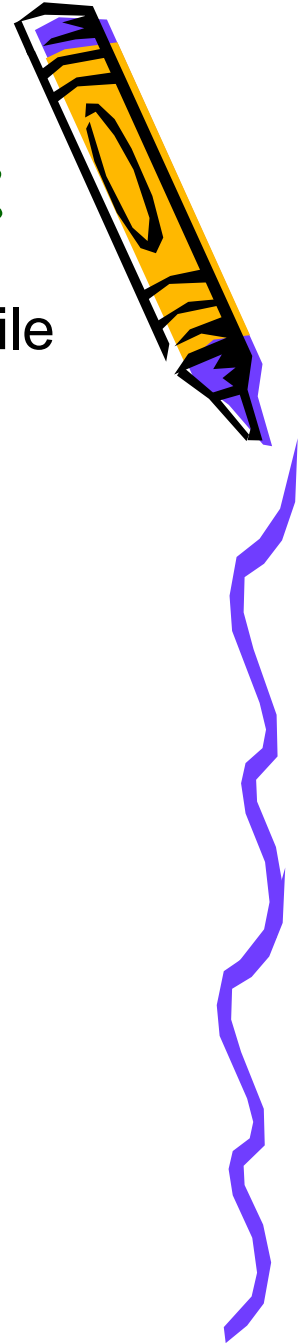
➤ `wc -c 1.txt`

➤ `ls |wc -l`

➤ `ls -a |wc -l`

➤ `ls /dev/tty* |wc -l`

- `-c`或`--bytes`或`--chars` 只显示Bytes数。
- `-l`或`--lines` 显示行数。
- `-w`或`--words` 只显示字数。



## 6.9 其它文件相关命令 - **grep**

grep, egrep, fgrep - print lines matching a pattern

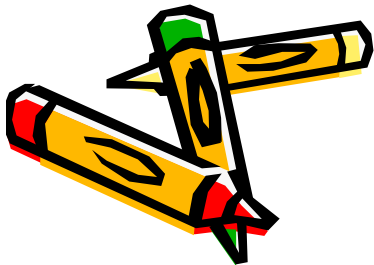
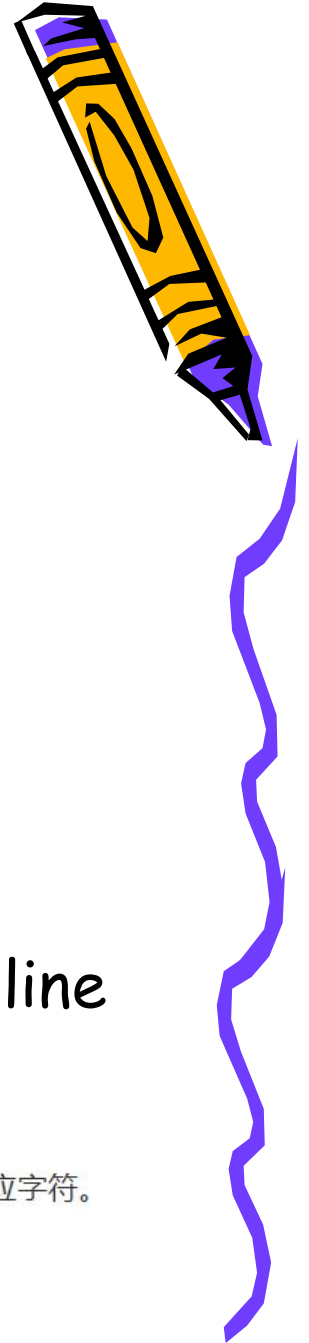
- `grep string file` 命令用于查找文件里符合条件的字符串
- `grep -n string file` (print line num)
- `grep -3 string file`
- `grep -c string file` (how many times)
- `grep -v string file` (non-matching lines)
- `grep -i string file` (without case-sensitivity)
- `grep "hello world" multifiles` (search in many files)
- `grep -n "hello world" multifiles`
- `cat /etc/passwd | grep stu`



## 6.9 其它文件相关命令 - **cmp, diff, sdiff**

用于比较两个文件是否有差异

- `$ cmp 1.txt 2.txt`
- `cmp` gives just the facts about **the first difference** between two files.
- `$ cp 1.txt 3.txt`
- `$ cmp 1.txt 3.txt`
- `$ vi 3.txt` //u can add/modify one line
- `$ cmp 1.txt 3.txt`
- `$ diff 1.txt 3.txt` //diff
- `$ diff 1.txt 2.txt`
- `$ diff 1.txt 4.txt` //4.txt contains blank line
- `$ sdiff 1.txt 4.txt`



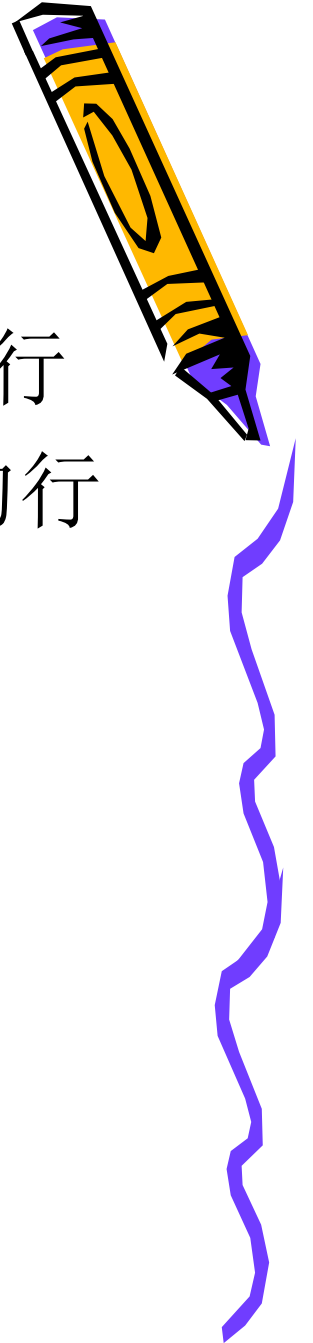
- `-c`或`--print-chars` 除了标明差异处的十进制字码之外，一并显示该字符所对应字符。
- `-i<字符数目>`或`--ignore-initial=<字符数目>` 指定一个数目。
- `-l`或`--verbose` 标示出所有不一样的地方。

## 6.9 其它文件相关命令 - sort

- `$ cat student_records`
- `$ sort student_records`
- `$ sort student_records > records.sorted`
- `$ sort -r student_records` // -r
- `$ sort -k 4 student_records` //以第4列为关键字排序
- `$ sort -k 4 -k 3 student_records`
- `$ sort -n sortnum.txt` //以数值排序，而不是以字符
- `$ sort -t ';' -k 2 com.txt` //以;为分隔符，第2列为关键字排序
- `$ sort -n -t ';' -k 2 com.txt`
- `$ sort -t ';' -k 1.3 com.txt`
  - -t<分隔字符> 指定排序时所用的栏位分隔字符。
  - //第一个域的第3个字符开始
- `$ sort 1.txt`
- `$ sort *.txt > txt.sorted.file`
- `$ cat txt.sorted.file`



## 6.9 其它文件相关命令 - **uniq**



- `$ uniq sorted.txt` //去除连续重复的行
- `$ uniq -d sorted.txt` //显示连续重复的行
- `$ sort student_records | uniq`
- `$ sort student_records | uniq -d`
- `$ sort -u student_records`  
= `$ sort student_recores | uniq`
- `$ uniq student_records`



## 6.9 其它文件相关命令 - sed

- **stream editor**, 以行为处理单位
  - **-e** 以选项中指定的参数格式来处理输入的文本文件。
  - **-f** 以选项中指定的**script**文件来处理输入的文本文件。
  - **-n** 仅显示模式匹配的结果。
- `$ sed '1' 1.txt`
- `$ sed -n '1' 1.txt` // 关闭输出
- `$ sed -n '9p' 1.txt` // 输出第9行
- `$ sed -n 'p' 1.txt` // 打印所有
- `$ sed -n '2~2p' 1.txt` // 打印2,4,6...
- `$ sed -n '1,3!p' 1.txt` // 打印1-3行以外的行("!"的作用)
- `$ sed '1,3d' 1.txt` // 删除1-3行
- `$ sed '1,3a *****' 1.txt` // 1-3行每一行后添加\*\*\*\*\*
- `$ sed '1,3y/abcde/ABCDE/' 1.txt` // 1-3行对应字母大写
- `$ sed '1d' 1.txt 2.txt` // 对多个文件进行处理
- `$ sed -e 's/hello/hi/g' 1.txt`
- `$ sed -e 's/popular/wellknown/g' sed.txt >sed.txt.new`
- `$ cat script.sed`
- `$ sed -f script.sed sed.txt >sed.txt.new`
- `$ sed -n '/hello/w hello.txt' 1.txt` // 包含hello的行写入hello.txt
- `$ sed -n 'l' 1.txt` // 字母l, 显示是否为tab键
- `$ sed -n 'l' student_records`
- `$ sed -i` // 直接对原始文件进行处理



## 6.9 其它文件相关命令- tac,rev,nl

➤ \$ tac rev-tac.txt

以行为单位倒着输出

➤ \$ rev rev-tac.txt

每一行以列为单位倒着输出

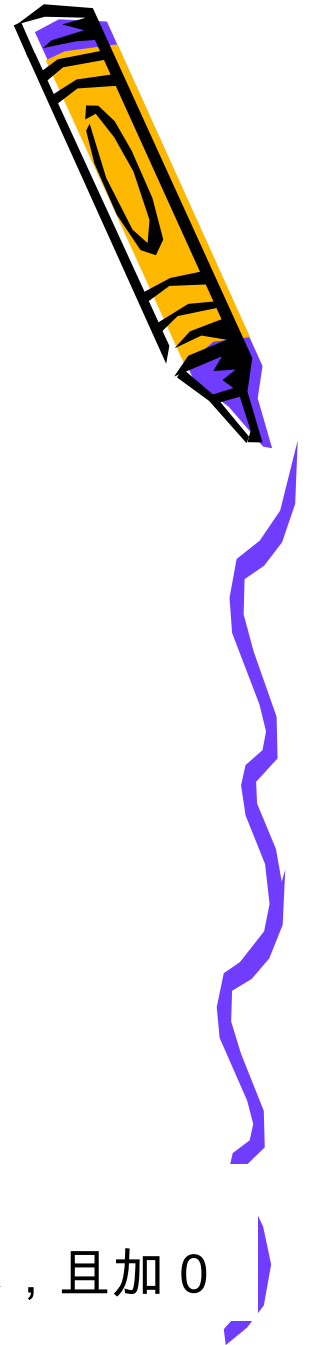
➤ \$ nl 1.txt

➤ \$ nl -n rz 1.txt

右对齐 带行号

-n rz

: 行号在自己栏位的最右方显示，且加 0





# stat 查看文件及文件的三个时间



- `$ stat 1.txt`
- `$ ls -l 1.txt` (Mtime: last modified)
- `$ ls -lc 1.txt` (Ctime: last changed)
- `$ ls -lu 1.txt` (Atime: last access)
- **Mtime**修改时间: (`ls -l`默认的文件时间), 当用**vim**对文件进行编辑之后保存, 它的**Mtime**就会相应的改变;
- **Atime**访问时间: 对文件最后一次读操作的访时间。例如**cat**、**more**等操作, 但是**stat**、**ls**等命令对其没有影响;
- **Ctime**状态时间: 当文件的属性状态被改变的时候, 例如当使用**chmod**、**chown**等改变文件属性的操作。



# stat 查看文件及文件的三个时间



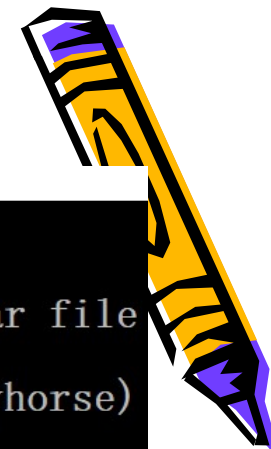
buyhorse@ubuntu-server-1804: ~/unix

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 192          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534785       Links: 1
Access: (0764/-rwxrw-r--)  Uid: ( 1000/buyhorse)  Gid: ( 1000/buyhorse)
Access: 2020-03-30 13:31:09.729057176 +0800
Modify: 2020-03-26 15:01:42.086466354 +0800
Change: 2020-04-02 09:31:42.922298854 +0800
Birth: -

buyhorse@ubuntu-server-1804:~/unix$ chmod u-x 1.txt
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 192          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534785       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)  Gid: ( 1000/buyhorse)
Access: 2020-03-30 13:31:09.729057176 +0800
Modify: 2020-03-26 15:01:42.086466354 +0800
Change: 2020-04-02 09:42:01.124233003 +0800
Birth: -

buyhorse@ubuntu-server-1804:~/unix$ ls -l 1.txt
-rw-rw-r-- 1 buyhorse buyhorse 192 Mar 26 15:01 1.txt
buyhorse@ubuntu-server-1804:~/unix$ ls -lu 1.txt
-rw-rw-r-- 1 buyhorse buyhorse 192 Mar 30 13:31 1.txt
buyhorse@ubuntu-server-1804:~/unix$ ls -lc 1.txt
-rw-rw-r-- 1 buyhorse buyhorse 192 Apr  2 09:42 1.txt
buyhorse@ubuntu-server-1804:~/unix$
```

# stat 查看文件及文件的三个时间



buyhorse@ubuntu-server-1804: ~/unix

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
```

```
File: 1.txt
Size: 192          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534785       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 09:53:02.528380664 +0800
Modify: 2020-03-26 15:01:42.086466354 +0800
Change: 2020-04-02 09:42:01.124233003 +0800
Birth: -
```

```
buyhorse@ubuntu-server-1804:~/unix$ cat 1.txt >1.txt.cat
```

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
```

```
File: 1.txt
Size: 192          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534785       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 09:53:02.528380664 +0800
Modify: 2020-03-26 15:01:42.086466354 +0800
Change: 2020-04-02 09:42:01.124233003 +0800
Birth: -
```

```
buyhorse@ubuntu-server-1804:~/unix$ vi 1.txt
```

```
buyhorse@ubuntu-server-1804: /unix$ stat 1.txt
```

```
File: 1.txt
Size: 193          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534787       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 10:03:10.651088861 +0800
Modify: 2020-04-02 10:03:10.651088861 +0800
Change: 2020-04-02 10:03:10.651088861 +0800
Birth: -
```





# touch filename

- 文件存在则修改其时间，否则创建该文件。

buyhorse@ubuntu-server-1804: ~/unix

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 193          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534787       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 10:25:13.606391893 +0800
Modify: 2020-04-02 10:25:13.606391893 +0800
Change: 2020-04-02 10:25:13.606391893 +0800
Birth: -
```

```
buyhorse@ubuntu-server-1804:~/unix$ touch -a 1.txt
```

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 193          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534787       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 10:30:30.554961195 +0800
Modify: 2020-04-02 10:25:13.606391893 +0800
Change: 2020-04-02 10:30:30.554961195 +0800
Birth: -
```

```
buyhorse@ubuntu-server-1804:~/unix$ touch -m 1.txt
```

```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 193          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534787       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)   Gid: ( 1000/buyhorse)
Access: 2020-04-02 10:30:30.554961195 +0800
Modify: 2020-04-02 10:31:19.988986452 +0800
Change: 2020-04-02 10:31:19.988986452 +0800
Birth: -
```

```
buyhorse@ubuntu-server-1804:~/unix$ touch 1.txt
```

```
buyhorse@ubuntu-server-1804:~/unix$
```

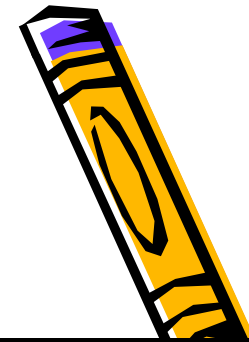
**touch -a //access time**

change time

**touch -m //modification time**

change time

**touch //all time**



```
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 217          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534786       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)  Gid: ( 1000/buyhorse)
Access: 2020-04-09 07:54:22.806877491 +0800
Modify: 2020-04-02 11:12:50.298482358 +0800
Change: 2020-04-02 11:12:50.298482358 +0800
Birth: -
buyhorse@ubuntu-server-1804:~/unix$ date
Thu Apr  9 10:44:09 CST 2020
buyhorse@ubuntu-server-1804:~/unix$ touch 1.txt
buyhorse@ubuntu-server-1804:~/unix$ stat 1.txt
File: 1.txt
Size: 217          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 534786       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/buyhorse)  Gid: ( 1000/buyhorse)
Access: 2020-04-09 10:44:15.195980820 +0800
Modify: 2020-04-09 10:44:15.195980820 +0800
Change: 2020-04-09 10:44:15.195980820 +0800
Birth: -
buyhorse@ubuntu-server-1804:~/unix$
```

# 其它文件相关命令

- ✓ 查看目录的容量

`du -h`

- ✓ 文件系统的磁盘使用情况统计

`df -i`

挂载点 Mounted on

- ✓ 查看Linux已载入到内存中的文件系统

`cat /proc/filesystems`

- ✓ 查看本地Linux支持的文件系统

`ls -l /lib/modules/$(uname -r)/kernel/fs`

- ✓ 挑选一个已格式化好的设备，查看文件的详细信息

`sudo dumpe2fs /dev/sda1 //了解`



# 小结&习题

## ➤ 思考题

- (1) 试描述Linux文件系统的结构。
- (2) Linux具有哪几种文件类型？
- (3) 什么是交换分区和交换文件。
- (4) 分区与文件系统是什么关系？

## ➤ 上机题

- (1) 如何对Linux中的文件的访问权限进行设置。
- (2) 安装和卸载CD-ROM的文件系统。
- (3) 创建一个独立的文件系统。
- (4) 查看系统的分区情况。
- (5) 编辑分区表，使得系统启动时就把MS-DOS分区/dev/hda1安装到/DOS目录下。
- (6) all available commands

