

# GNU/Linux 文件管理命令



# 文件管理命令

## ls 目录及文件的列示

命令格式      ls [ 选项 ]      [ 目录 ]

### 参数

-l      以长格式显示目录文件

```
drwxr-sr-x  2 root src  4096 2006-11-16 10:56 ati
drwxr-xr-x 19 root root 4096 2006-10-25 21:30 linux-headers-2.6.17-10
drwxr-xr-x  4 root root 4096 2006-10-25 21:30 linux-headers-2.6.17-10-generic
drwxr-xr-x  7 root root 4096 2006-11-09 13:41 rpm
```

权限 硬连接 所属用户及用户组 文件大小 创建和修改日期 文件名

# 文件管理命令

-a 这个选项能显示 . 开头的隐藏文件  
-i 显示每个文件的 inode 号  
( 关于 inode 号的详细解释还会在后面课程里介绍 )

-m 所有项目以逗号分隔 , 并填满整行行宽  
-R 同时列出所有子目录层  
-h 将列出文件的大小以人性化格式输出 🗨️

--color=tty ( 2 个 - 号 ) 显示文件的时候以色彩提示  
目录      可执行文件      连接文件      一般文件  
蓝色      绿色 🗨️      淡蓝色      白色

# 文件管理命令

在使用 ls 的时候还对显示出来的内容按要求排序

--sort

time          按时间

size          按大小

示例

ls --sort time

ls --sort size




# 文件管理命令

对文件的状态修改时间、访问时间排序

状态修改情况：

1. 改变属主、属组
2. 改变文件权限操等
3. 对 i 节点进行的的操作

示例

ls --time=atime 

ls --time=ctime



# 文件管理命令

查看文件状态修改时间

#ls -lc [ 文件名 ]

查看文件的访问时间

#ls -lu [ 文件名 ]

查看文件的最后修改时间

#ls -l



# 文件管理命令

如果需要按要求排序

`--full-time` 显示文件的详细访问时间

示例

`ls -full-time`

显示本地目录下以“.”开头的文件

`ls -d .*`




# 文件管理命令



清屏： 命令 `clear`

小诀窍：可以使用 `ctrl+l` 来实现 `clear` 命令的功能





# 文件管理命令

命令：date 显示系统当前时间

命令格式：date [选项] 

选项：-R 按照 RFC822 文档中的时间格式  
来实现时间

-r 文件名 显示文件的最后修改日期

-u 显示 utc 时间（通用时间）

# 文件管理命令

date 还可以按照管理员给定的格式来显示时间  
使用方法 date +" 变量 "

变量 : %a 简写星期 (sat,sun)

%D 按 月 / 日 / 年格式显示

%A 星期 (sunday)

%d 显示日期

%b 简写月 (Jan)

%H 小时 (24 小时制)

%B 月 (January)

%I 小时 (12 小时制)

%j 显示一年中的第几天

%M 分钟 (00-59)

%U 显示一年中的第几周

%S 秒

# 文件管理命令

%y 年简写

%Y 年

%r 以 12 小时制的时间格式显示 小时 / 分钟 / 秒 上午 / 下午

%s 以 1970 年 1 月 1 日 0 时 0 分开始计算到目前所经过的时间

%x 按照 月 / 日 / 年 显示

%X 按照 H:M:S 显示

%c 显示当前时区和详细的时间

示例： date + "time %H 小时 %A 星期"

## 文件管理命令

使用 date 修改 Linux 上的系统时间

date 命令不光能显示时间，还能修改系统上的时间。

修改方法：

date MMDDhhmmyy

MM 月 0-12

DD 日 1-31

hh 小时 00-23

mm 分钟 0-59

yy 年 1970---

注意：修改系统时间需要管理员权限来做



# 文件管理命令

使用 date 查看过去 / 将来的时间

1. 查看 1945 年 8 月 15 日是星期几 ( 当前时间为 2014-9-27)

```
#date -d "-69 year -1 month -12 days"
```

2. 显示 1945 年 8 月 15 日 12:00:00( 当前时间为 2014-9-27)

```
#date -d "-69 year -2 month -12 days -7 hour  
-5 mins -10 sec"
```

# 文件管理命令

使用 date 查看过去 / 将来的时间

1. 查看 2045 年 8 月 15 日是星期几 ( 当前时间为 2014-9-27)

```
#date -d "+31year -1 month -12 day"
```

2. 显示 2014 年 1 月 1 日星期几

```
#date -d "+3month +5day"
```

或

```
#date -d "+4month -26day"
```



## 文件管理命令

命令：hwclock 显示硬件时钟

将系统时钟同步至硬件时钟

#hwclock --systohc

将硬件时钟同步至系统时钟

#hwclock --hctosys



# 文件管理命令

命令：tzselect

功能：设置系统时区

语法格式：tzselect

示例

1. 设置当前系统的时区  
#tzselect





# 文件管理命令

命令：timedatectl

功能：设置系统时间及日期

语法格式：timedatectl [ 选项 ] [ 日期 ]

示例：

1. 显示各项当前时间

#timedatectl

2. 显示系统所支持的时间区域

#timedatectl list-timezones



## 文件管理命令

### 3. 设置当前系统的时间区域

```
#timedatectl set-timezone  
Asia/Shanghai
```

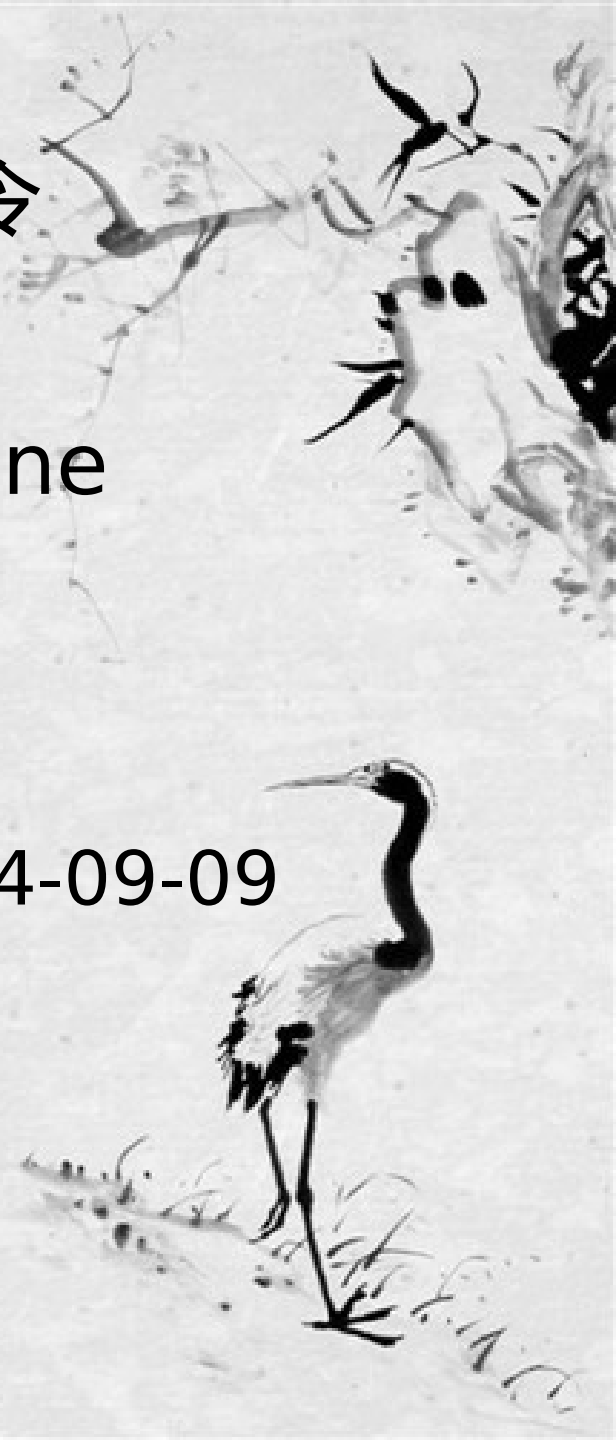
### 4. 设置当前系统时间

```
#timedatectl set-time "2014-09-09  
12:29:30"
```

### 5. 设置 NTP 支持

```
#timedatectl set-ntp true
```

### 6. 取消 NTP 支持



## 文件管理命令

命令：cal 日历

显示指定年份之全年信息

#cal 2009 ←-2009 为指定年份

显示指定年、月的信息

#cal 10 2009

显示当前年、当前月信息

#cal



# 文件管理命令

命令：mkdir      创建目录

命令格式： mkdir [ 选项 ] 新建目录名  
选项：

-p 建立多级目录

-m 在建立目录的时候给目录赋予权限值

示例：mkdir -p /name/a/b/c



# 文件管理命令

命令：`rmdir` 删除空目录

命令格式：`rmdir [ 选项 ] 目录名`

选项：

`-p` 删除多级空目录

示例：`mkdir -p /name/a/b/c`

注意：目录本身一定要是空的



# 文件管理命令

命令 :rm 删除文件 / 目录

命令格式        rm [选项]    文件名

选项： -i    在删除文件之前需要手工确认

      -v    在删除文件的时候显示信息

      -r    删除目录    -f 忽略提示

示例： rm -v filename

常用选项： rm -rf 目录名



# 文件管理命令

命令 :cp 复制命令

命令格式 :

cp [ 选项 ] 文件名 ( 原件 ) 目的文件名 ( 复印件 )

选项 : -R 复制整个目录里的内容

-p 复制完后保持目录的权限值

-v 在复制文件的时候显示进度

-f 在复制的时候如果碰到目的文件名有重复

就将原先的删除

示例 : cp -vRn /etc/ /home/usr/

# 文件管理命令

命令：mv      移动文件或目录 / 修改文件或目录名

命令格式： mv [ 选项 ] 源文件 目的路径

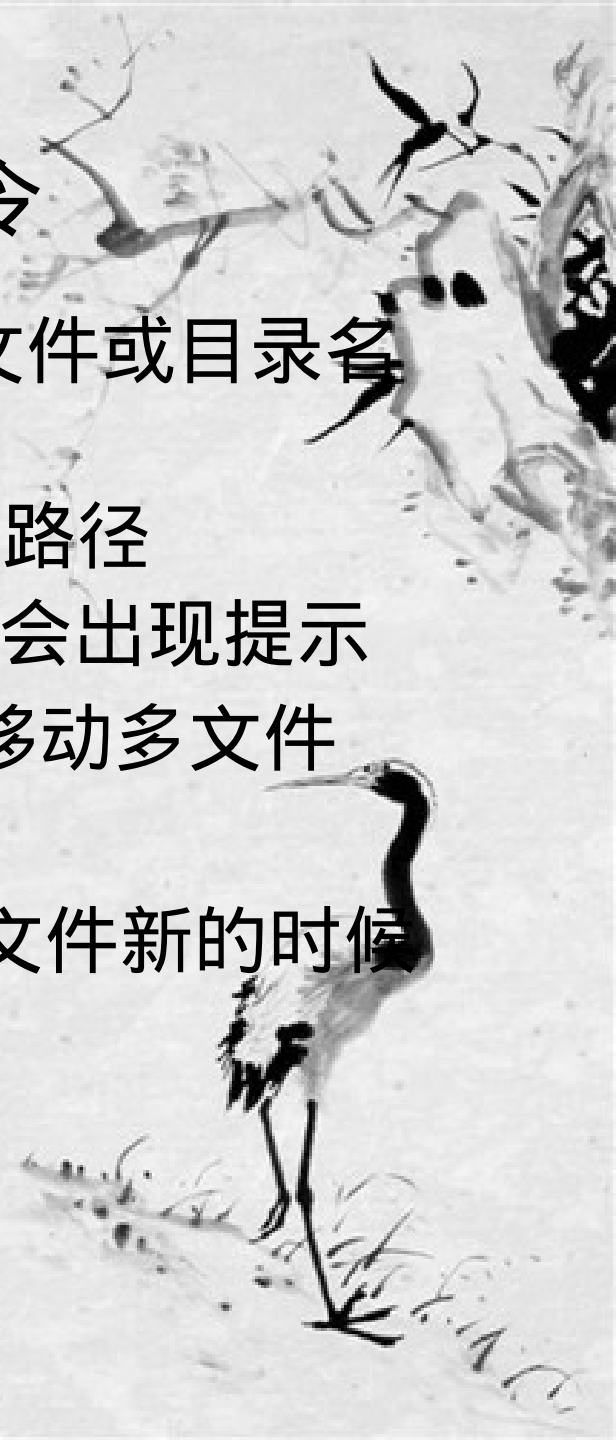
选项： -i 如果目的地有相同文件名时会出现提示

-v 在搬移文件时显示进度，在移动多文件时非常有用

-u 当移动时只有源文件比目的文件新的时候才会移动

-f 强制覆盖已有的文件

示例：mv -i 1.jpg /home/2.jpg





# 文件管理命令

命令：pwd 显示当前工作路径



# 文件管理命令

命令：cd            切换工作目录

命令格式： cd   路径

术语：

1. 相对路径
2. 绝对路径



# 文件管理命令

命令 :cat 查看文本文件

命令格式： cat [ 选项 ] 文件名

选项： -b 显示文件内容的时候显示行数

-n 显示文件内容包括空行

-s 将多个空行合并成一个空行输出



# 文件管理命令

命令：more 查看文本文件命令

命令格式： more [ 选项 ] 文件名

选项： + 行数 直接从给定的行数开始显示

-s 将多个空行压缩成一个空行

-p 清除屏幕后再显示

提示：用 more 打开一个文件后用空格向下翻页，用 b 向上翻页



# 文件管理命令

命令：less 查看文本文件命令

命令格式： less 文件名



# 文件管理命令

命令：head      查看文件头部

命令格式：head    [ 选项 ]    文件

选项：

- n < 行数 >      显示文件的最前指定的行
- c < 字节数 >    显示文件前 N 个字节数里的内容
- q                不输出文件头的内容
- v                输出文件头的内容

示例： head -n 15 test.file



# 文件管理命令

命令：tail 查看文件尾部

命令格式：tail [选项] 文件

选项：

-f 循环读取

-c < 字节数 > 显示文件前 N 个字节数里的内容

-q 不输出文件头的内容

-n < 行数 > 指定所显示的行数

-v 输出文件头的内容

示例：tail -n 15 test.file



# 文件管理命令

命令：nano

功能：文本编辑器

命令格式：nano [选项][行号]

文件名





# 文件管理命令

命令：strings      查看二进制

命令格式：strings      文件



# 文件管理命令

命令 :ln 生成链接文件

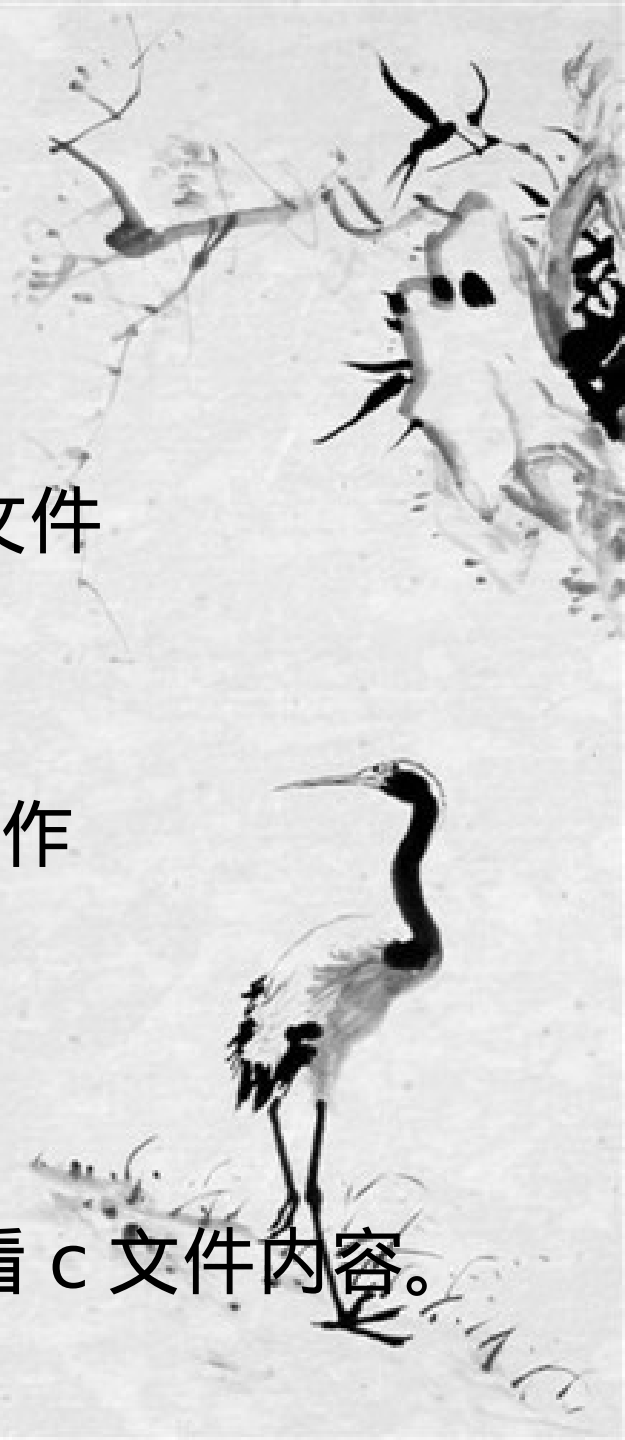
命令格式 : ln [ 选项 ] 源文件 链接文件  
选项

- f 删除已存在的目的文件
- i 如果碰到有重复名字的提示如何操作
- v 显示操作信息
- s 软链接选项

软链接 ln -s a c 然后删除 a 后 , 查看 c 文件内容。

区别出软链接和硬链接的不同

硬链接和软链接有什么区别使用?



# 文件管理命令

示例：

1. 改变 CentOS7 的运行级别，由 multi-user.target 改为 graphical.target

```
#ln -sf /lib/systemd/system/graphical.target /etc/systemd/system/default.target
```



# 文件管理命令

命令 :diff 文件逐行对比

命令格式 : diff [ 选项 ] file1 file2

显示信息 :

a	为需要附加
d	为需要删除
c	为需要修改

示例 :

1. 对比 test.txt 与 test1.txt  
#diff test.txt test1.txt



# 文件管理命令

命令：tree 以树型结构查看文件

命令格式：tree

需求：需要另行安装 tree 软件包



# 文件管理命令

命令：file

功能：查看文件的类型

命令格式：file [ 选项 ] < 文件 / 目录 >



# 文件管理命令

命令：stat

功能：显示文件或文件系统状态

命令格式：stat [ 选项 ]      < 文件 / 目录 >

选项：

-f: 查看指定的文件系统



# 文件管理命令

示例：

1. 查看 / 分区

```
#stat -f /
```

2. 查看 test.txt 文件信息

```
#stat test.txt
```





# 文件管理命令

stat 可查看

1. 文件名

2. 文件尺寸

3. l 节点号

4. 创建时间 / 访问时间 / 状态（属主、组、权限）修改时间



# 文件管理命令

stat 可查看

5. 权限

6. 链接文件个数

7. 属主及属组等



# 文件管理命令

命令 :ldd

功能：查找某个二进制文件所使用的共享库（动态库）

语法格式 :ldd [ 选项 ] 文件

示例：

```
#ldd /bin/ls
```



# 文件管理命令

## 1. 静态函数库

这类库的名字一般是 `libxxx.a` ；利用静态函数库编译成的文件比较大，因为整个函数库的所有数据都会被整合进目标代码中，他的优点就显而易见了，即编译后的执行程序不需要外部的函数库支持，因为所有使用的函数都已经被编译进去了。当然这也会成为他的缺点，因为如果静态函数库改变了，那么你的程序必须重新编译。

# 文件管理命令

## 2. 动态函数库

这类库的名字一般是 `libxxx.so`; 相对于静态函数库, 动态函数库在编译的时候 并没有被编译进目标代码中, 你的程序执行到相关函数时才调用该函数库里的相应函数, 因此动态函数库所产生的可执行文件比较小。由于函数库没有被整合进你的程序, 而是程序运行时动态的申请并调用, 所以程序的运行环境中必须提供相应的库。动态函数库的改变并不影响你的程序, 所以动态函数库的升级比较方便。

# 文件管理命令

Linux 系统有几个重要的目录存放相应的函数库：

/lib

/usr/lib。



# 文件管理命令

多命令配合或结果导入、导出符号

“|” 管道符号

“>” 输出重定向

“>>” 输出重定向

“<” 输入重定向



# 文件管理命令

多命令配合或结果的引用符号

“ ” 双引号

' ' 单引号

` ` 反引号

