

Linux操作系统编程

常用UNIX/Linux命令  
- 文件目录命令

- 文件目录类命令
- 进程控制类命令
- 用户及权限管理类命令



- 命令格式： 命令名称
- -选项
- 参数
- 例： `ls -la /etc`



- 浏览目录命令: `ls`    `pwd`
- 目录操作命令: `cd`    `mkdir`    `rmdir`
- 浏览文件命令: `cat`    `more`    `less`    `head`    `tail`
- 文件操作命令: `cp`    `rm`    `mv`    `find`    `grep`    `tar`



- 命令名称: ls
  - 命令英文原意: list
  - 功能描述: 显示目录文件
  - 语法: ls 选项[-ald] [文件或目录]
- a 显示所有文件，包括隐藏文件
- l 详细信息显示
- d 查看目录属性

```
[root@linux root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@linux root]# ls -a
.                .bash_logout  .cshrc         install.log.syslog  .Xresources
..               .bash_profile .gtkrc         .kde
anaconda-ks.cfg  .bashrc       install.log     .tcshrc
[root@linux root]# ls -l
total 32
-rw-r--r--  1 root    root      3811 Mar 10  2005 anaconda-ks.cfg
-rw-r--r--  1 root    root     24636 Mar 10  2005 install.log
-rw-r--r--  1 root    root        0 Mar  9  2005 install.log.syslog
[root@linux root]# _
```

- 命令名称: pwd
- 命令英文原意: print working directory
- 语法: pwd
- 功能描述: 显示当前所在的工作目录

```
root@ubuntu:/usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4# pwd
/usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4
root@ubuntu:/usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4#
```



- 浏览目录命令: ls pwd
- 目录操作命令: cd mkdir rmdir
- 浏览文件命令: cat more less head tail
- 文件操作命令: cp rm mv find grep tar



- 命令名称: cd
- 命令英文原意: change directory
- 语法: cd [目录]
- 功能描述: 切换目录
- 范例: `$ cd /` 切换到根目录  
`$ cd ..` 回到上一级目录

- 命令名称: mkdir
- 命令英文原意: make directory
- 语法: mkdir [目录名]
- 功能描述: 创建新目录
- 范例: `$ mkdir newdir`



- 命令名称: `rmdir`
- 命令英文原意: `remove directory`
- 语法: `rmdir [目录名]`
- 功能描述: 删除空目录
- 范例: `$ rmdir newdir`



- 浏览目录命令: ls pwd
- 目录操作命令: cd mkdir rmdir
- 浏览文件命令: cat more less head tail
- 文件操作命令: cp rm mv find grep tar



- 命令名称: cat
- 命令英文原意: concatenate and display files
- 功能描述: 连接文件并打印到标准输出设备上
- 选项: -E 在每一行的末尾显示\$  
-n 为显示行添加行号
- 参数: 指定要连接的文件列表。
- 范例: cat m1 (在屏幕上显示文件m1的内容)  
cat m1 m2 (同时显示文件m1和m2的内容)



- 命令名称: more
- 语法: more [文件名]  
(空格) 或 f 显示下一页  
(Enter) 显示下一行  
q 或 Q 退出
- 功能描述: 分页显示文件内容
- 范例:  
\$ more /etc/services

less命令的用法与more命令类似，也可以用来浏览超过一页的文件。所不同的是less命令除了可以按空格键向下显示文件外，还可以利用上下键来卷动文件。



■ **head命令**：用于显示文件前几行的内容

■ **格式**：head [参数] <文件名>

- -n：显示前n行，不指定此参数显示前10行

```
[root@linux root]# head /etc/passwd
```

■ **tail命令**：用于显示文件后几行的内容

■ **格式**：tail [参数] <文件名>

- -n：显示后n行，不指定此参数显示后10行

- +n：从第n行显示到文件尾

- -F：用于跟踪显示不断增长的文件结尾内容（通常用于显示日志文件）。

```
[root@linux root]# tail /etc/passwd
```



- 浏览目录命令: ls pwd
- 目录操作命令: cd mkdir rmdir
- 浏览文件命令: cat more less head tail
- 文件操作命令: cp rm mv find grep tar



- **cp (copy) 命令** 可以将文件从一处复制到另一处。一般在使用cp命令时将一个文件复制成另一个文件或复制到某目录时，需要指定源文件名与目标文件名或目录。
- **格式：** cp [参数] <源文件路径> <目标文件路径>
  - **-p** : 连同文件的属性一起复制，而非使用默认方式，常用于备份
  - **-i** : 若目标文件已经存在时，在覆盖时会先询问操作的进行
  - **-r** : 递归持续复制，用于目录的复制行为
  - **-u** : 目标文件与源文件有差异时才会复制



**将test1.txt复制成test2.txt:**

```
[test@linux test]$ cp test1.txt test2.txt
```

**将test3.txt复制到/tmp目录中:**

```
[test@linux test]$ cp test3.txt /tmp (目录)
```

**目录拷贝:**

```
[test@linux test]$ cp -r test1(目录) test2(目录)
```



- **rm(remove) 命令：** 删除文件或目录
- **格式：** rm [参数] <目标文件路径>
  - f：就是**force**的意思，忽略不存在的文件，不会出现警告消息
  - i：互动模式，在删除前会询问用户是否操作
  - r：递归删除，最常用于目录删除，它是一个非常危险的参数



删除某个文件:

```
[test@linux test]$ rm myfiles (文件名)
```

删除当前目录下的所有文件:

```
[test@linux test]$ rm *
```

递归删除某个目录（包含该目录下所有文件及子目录）:

```
[test@linux test]$ rm -r myfolder (目录名)
```

强迫删除所有后缀名为txt文件:

```
[test@linux test]$ rm -f *.txt
```

删除当前目录下的所有文件:

```
[test@linux test]$ rm -i *
```

（删除文件时会询问,可按Y或N键表示允许或拒绝删除文件）



■ **find命令**：用来寻找文件或目录

■ **格式**：find 路径 [参数]

- **-name filename**：找出文件名为**filename**的文件
- **-size [+ -]SIZE**：找出比**SIZE**还要大（+）或小（-）的文件
- **-tpye TYPE**：查找文件的类型为**TYPE**的文件，**TYPE**的值主要有：一般文件（**f**）、设备文件（**b**、**c**）、目录（**d**）、连接文件（**l**）、**socket**（**s**）、**FIFO**管道文件（**p**）；
- **-perm mode**：查找文件权限刚好等于**mode**的文件，**mode**用数字表示，如**0755**；



# 文件操作类命令

```
root@ubuntu: /usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4
root@ubuntu: /usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4# find ctime
ctime
root@ubuntu: /usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4# find -name cti
me
./ctime
./tr1/ctime
root@ubuntu: /usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4# find -name ctime
./ctime
./tr1/ctime
root@ubuntu: /usr/local/genode-gcc/arm-elf-eabi/include/c++/4.7.4#
```

```
root@ubuntu: /dev
./net
./cpu
root@ubuntu: /dev# find -type c
./hidraw0
./vmci
./fb0
./dri/card0
./dri/controlD64
./vcsa5
./vcs5
./vcsa4
./vcs4
./vcsa3
./vcs3
./vcsa2
./vcs2
./vcsa6
./vcs6
./lp0
./parport0
```



- **grep命令**（Globally search a Regular Expression and Print）  
功能：在文件中搜索匹配的字符并进行输出
- **格式**：grep[参数] <要找的字串> <要寻找字串的源文件>
  - **-a**：将**binary**文件以**text**文件的方式查找数据
  - **-c**：计算找到“查找字符串”的次数
  - **-i**：忽略大小写的区别，即把大小写视为相同
  - **-v**：反向选择，即显示出不包含‘查找字符串’内容的那一行



# 文件操作命令

```
root@ubuntu:/usr/include# grep errno errno.h
*      ISO C99 Standard: 7.5 Errors      <errno.h>
#include <bits/errno.h>
/* Declare the `errno' variable, unless it's defined as a macro by
   bits/errno.h. This is the case in GNU, where it is a per-thread
#ifdef errno
extern int errno;
/* The Hurd <bits/errno.h> defines `error_t' as an enumerated type so
root@ubuntu:/usr/include#
```

```
root@ubuntu: /usr/include
root@ubuntu:/usr/include# clear

root@ubuntu:/usr/include# grep errno *.h
argp.h:#include <errno.h>
argz.h:#include <errno.h>
envz.h:#include <errno.h>
err.h:/* Print "program: ", FORMAT, ": ", the standard error string for errno,
errno.h: *      ISO C99 Standard: 7.5 Errors      <errno.h>
errno.h:#include <bits/errno.h>
errno.h:/* Declare the `errno' variable, unless it's defined as a macro by
errno.h: bits/errno.h. This is the case in GNU, where it is a per-thread
errno.h:#ifdef errno
errno.h:extern int errno;
```



■ **tar (tape archive)命令**能够将用户所指定的文件或目录打包成一个文件，也可以通过指定参数开启压缩/解压缩功能

■ **格式：**tar [参数] <文件>

- **-c**：新建打包文件
- **-t**：查看打包文件的内容含有哪些文件名
- **-x**：解打包或解压缩的功能，可以搭配**-C**（大写）指定解压的目录，注意**-c,-t,-x**不能同时出现在同一条命令中
- **-j**：通过**bzip2**的支持进行压缩/解压缩
- **-z**：通过**gzip**的支持进行压缩/解压缩
- **-v**：在压缩/解压缩过程中，将正在处理的文件名显示出来
- **-f filename**：**filename**为要处理的文件
- **-C dir**：指定压缩/解压缩的目录**dir**



**将当前目录的所有文件打包成test.tar:**

```
[root@linux test]# tar -cvf test.tar *
```

**将当前目录的所有文件打包成test.tar,再用gzip命令压缩**

```
[root@linux test]# tar -czvf test.tar.gz *
```

**查看test.tar文件中包括了哪些文件:**

```
[root@linux ljr]# tar -tf test.tar
```

**将test.tar解打包:**

```
[root@linux test]# tar -xvf test.tar
```

**将foo.tar.gz解压缩并解打包:**

```
[root@linux test]# tar -xzvf foo.tar.gz
```



