

Shell



操作系统Kernel与Shell

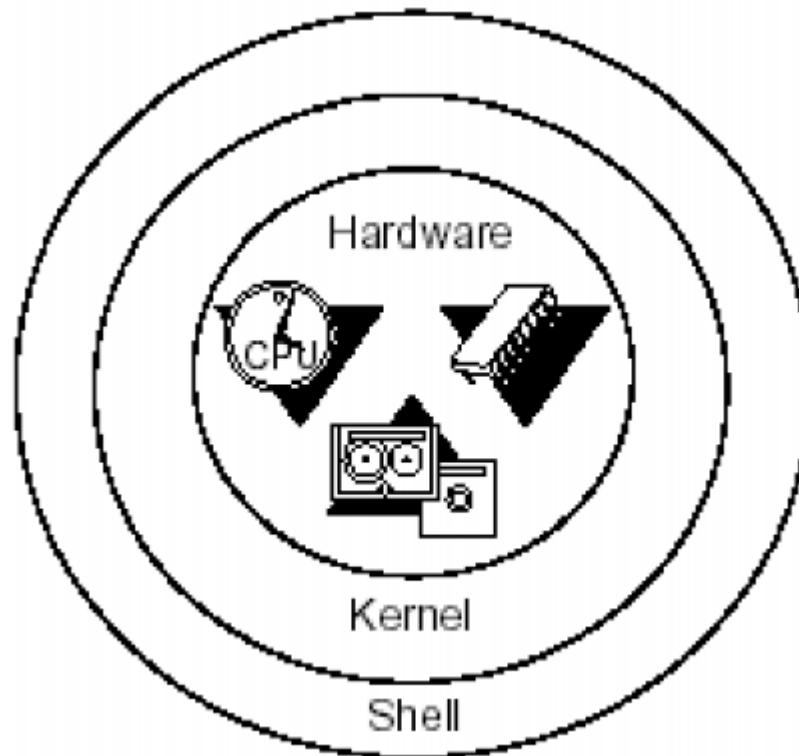
目前，我们知道了一台正常运行的计算机由硬件和软件组成，其中硬件最基本应该包括：最为珍贵的资源——CPU，代码运行的场所——内存，批量数据存储的场所——外部存储，人机交互的部分——I/O(输入/输出部件)。

但是，我们对软件部分仍然感到迷茫，本质上，整个软件部分应该分为两大部分：应用软件和操作系统。其中，应用软件运行于操作系统之上，或者说，应用软件依赖于操作系统而和底层的硬件交互。比如：我们经常使用的QQ、Firefox、WPS等等属于应用软件。而Windows XP、Windows Vista、Mac OS、Ubuntu等等属于操作系统。

更进一步，操作系统可以分为kernel和shell，其中kernel是操作系统内核的意思，操作系统内核是指大多数操作系统的核心部分。它由操作系统中用于管理存储器、文件、外设和系统资源的那些部分组成。操作系统内核通常运行进程，并提供进程间的通信（IPC）机制。

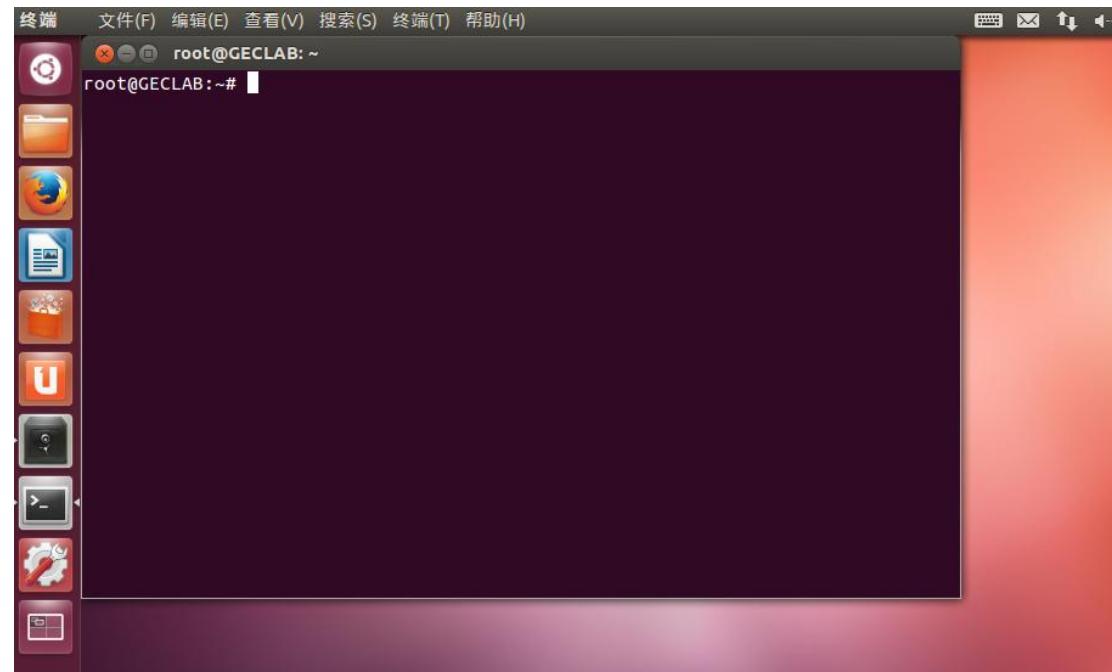
下面，我们来说说shell，这里的shell并不是希尔排序，在计算机科学中，shell俗称外壳（用来区别于内核），是指“提供使用界面”的软件（或者叫做命令解析器）。它类似于DOS下的command.com，接收用户命令，然后调用相应的应用程序。同时它又是一种程序设计语言。作为命令语言，它交互式解释和执行用户输入的命令或者自动解释和执行预先设定好的一连串的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才有的控制结构，包括循环和分支。目前阶段，我们知道，在shell下输入命令就会得到执行。

联系



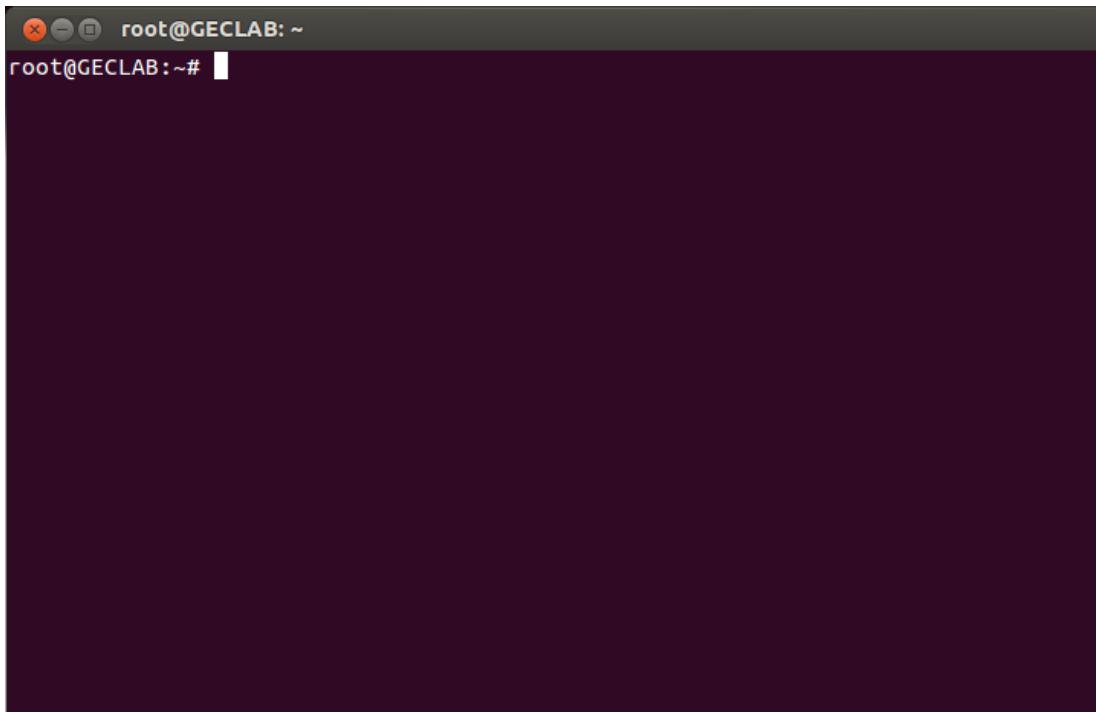
Shell位置

按CTRL+ALT+T，就可以弹出窗口了。中断运行之后会启动一个shell为我们服务。



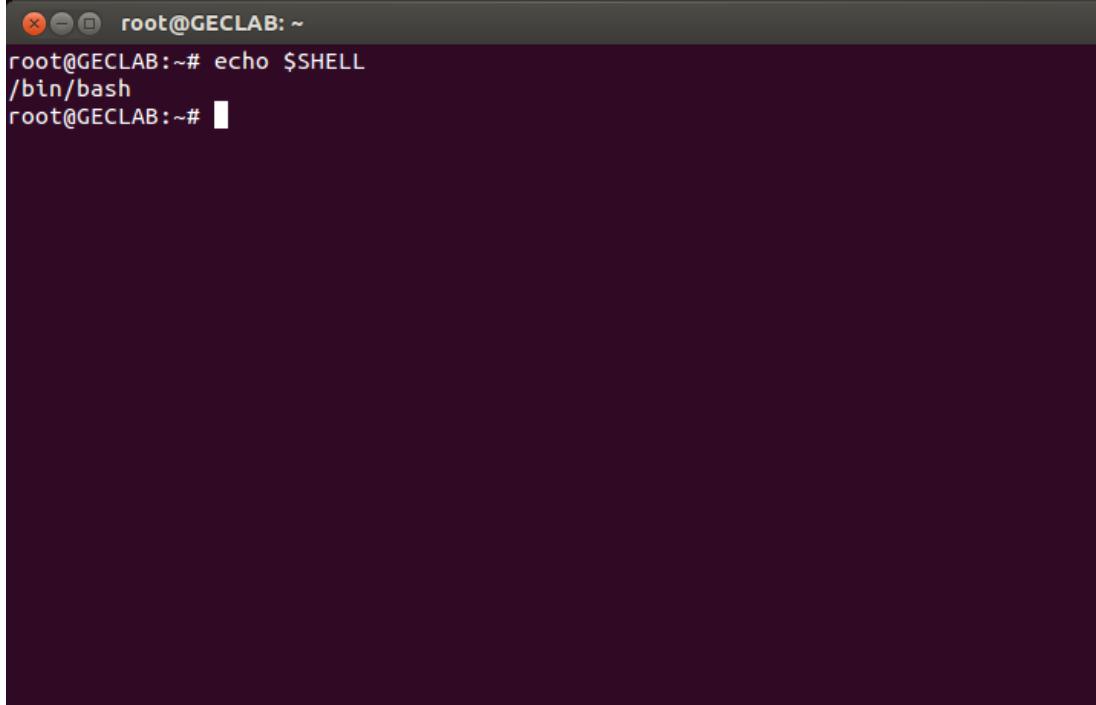
Shell位置

(续) 终端运行之后, 光标闪烁, 等待我们的输入, 鼠标功能也收到限制:



Shell位置

我们输入第一条命令: echo \$SHELL , 注意大小写及中间的空格:



```
root@GECLAB:~# echo $SHELL
/bin/bash
root@GECLAB:~#
```

Shell类型

命令顺利地得到了执行，结果提示我们正在使用bash，其实，bash是shell类型的一种，我们来看一下经常使用的shell有那几个类型：

Bourne Shell:

- 由斯蒂芬. 伯恩为AT&T UNIX系统编写；
- 是最早和最知名的Unix Shell；
- 拥有大多数shell脚本的共同点；
- 提示符是“#”为根用户，普通用户是“\$”；
- 下一个版本是bash，增强了b-sh。

C shell:

- 由文本编辑器vi 的作者比尔. 乔伊在美国加州大学伯克利分校作为BSD UNIX的一部分创建；
- 提供命令行的历史和编程环境。
- 类似于C编程语言；
- 不兼容Bourne shell语法；
- 其中tcsh是csh的增强，完全兼容；
提示符是“#”为根用户，普通用户使用“%”。

Linux命令

概述

Linux提供了大量的命令，利用它可以有效地完成大量的工作，如磁盘操作、文件存取、目录操作、进程管理、文件权限设定等。所以，在Linux系统上工作离不开使用系统提供的命令。要想真正理解Linux系统，就必须从Linux命令学起，通过基础的命令学习可以进一步理解Linux系统。

不同Linux发行版的命令数量不一样，但Linux发行版本最少的命令也有200多个。这里把比较重要和使用频率最多的命令一一进行讲解。



文件管理

磁盘管理

文档编辑

文件传输

磁盘维护

网络通讯

系统管理

系统设置

备份压缩

设备管理

预览

文件管理

cat	chattr	chggrp	c hmod	chown	cksum	cmp
diff	diffstat	file	find	git	gitview	indent
cut	ln	less	locate	isattr	mattrib	mc
mdel	mdir	mktemp	more	mmove	mread	mren
mtools	mtoolstest	mv	od	paste	patch	rcp
rm	slocate	split	tee	tmpwatch	touch	umask
which	cp	in	mcopy	mshowfat	rhmask	whereis

磁盘管理

cd	df	dirs	du	edquota	eject
mc	mdeltree	mdu	mkdir	mlabel	mmd
mrd	mzip	pwd	quota	mount	mmount
rmdir	rmt	stat	tree	umount	ls
quotacheck	quotaoff	lndir	repquota	quotaon	

文档编辑

col	colrm	comm	csplit	ed	egrep	ex	fgrep
fmt	fold	grep	ispell	jed	joe	join	look
mtype	pico	rgrep	sed	sort	spell	tr	expr
uniq	wc						

文件传输

lprm	lpr	lpq	lpd	bye	ftp
uuto	uupick	uucp	uucico	tftp	ncftp
ftpshut	ftpwho	ftpcount			

预览

磁盘维护

badblocks	cfdisk	dd	e2fsck	ext2ed
fsck	fsck	fsconf	fdformat	hdparm
mformat	mkbootdisk	mkdosfs	mke2fs	mkfs.ext2
mkfs.msdos	mkinitrd	mkisofs	mkswap	mpartition
swapon	symlinks	sync	mbadblocks	mkfs
fsck.ext2	fdisk	losetup	mkfs	sfdisk
swapoff				

网络通讯

apachectl	arpwatch	dip	getty	mingetty	uux
telnet	uulog	uustat	ppp-off	netconfig	nc
httpd	ifconfig	minicom	mesg	dnsconf	wall
netstat	ping	pppstats	samba	setserial	talk
traceroute	tty	newaliases	uname	netconf	write
statserial	efax	pppsetup	tcpdump	ytalk	cu
smbd	testparm	smbd	smbclient	shapecfg	

系统管理

adduser	chfn	useradd	date	exit	finger
fwhois	sleep	suspend	groupdel	groupmod	halt
kill	last	lastb	login	logname	logout
ps	nice	procinfo	top	pstree	reboot
rlogin	rsh	sliplogin	screen	shutdown	rwwho
sudo	gitps	swatch	tload	logrotate	kill
uname	chsh	userconf	userdel	usermod	vlock
who	whoami	whois	newgrp	renice	su
skill	w	id	free		

系统设置

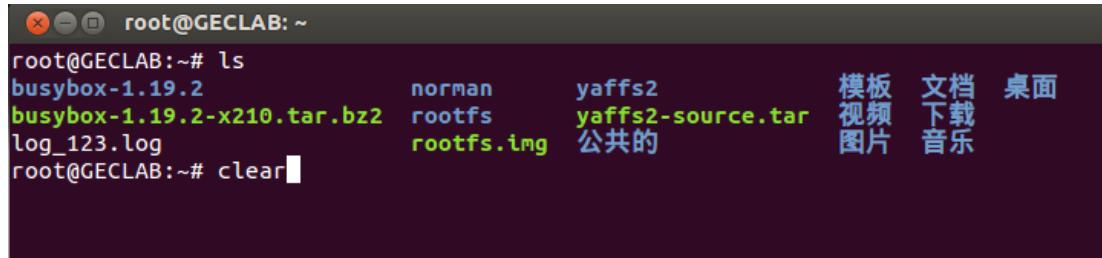
reset	clear	alias	dircolors	aumix	bind
chroot	clock	crontab	declare	depmod	dmesg
enable	eval	export	pwunconv	grpconv	rpm
insmod	kbdconfig	lilo	liloconfig	lsmod	minfo
set	modprobe	ntsysv	moouseconfig	passwd	pwconv
rdate	resize	rmmod	grpunconv	modinfo	time
setup	sndconfig	setenv	setconsole	timeconfig	ulimit
unset	chkconfig	apmd	hwclock	mkkickstart	fbset
unalias	SVGAText Mode				

预览

备份压缩					
ar	bunzip2	bzip2	bzip2recover	gunzip	unarj
compress	cpio	dump	uuencode	gzexe	gzip
lha	restore	tar	uudecode	unzip	zip
zipinfo					

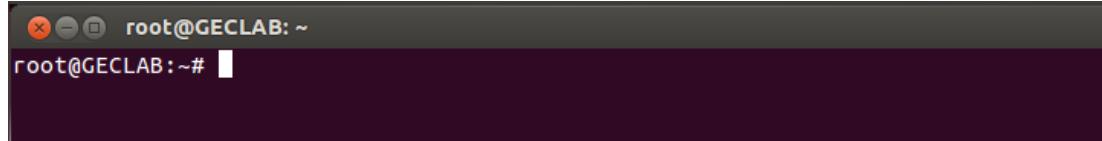
setleds	loadkeys	rdev	dumpkeys	MAKEDEV

清屏-clear



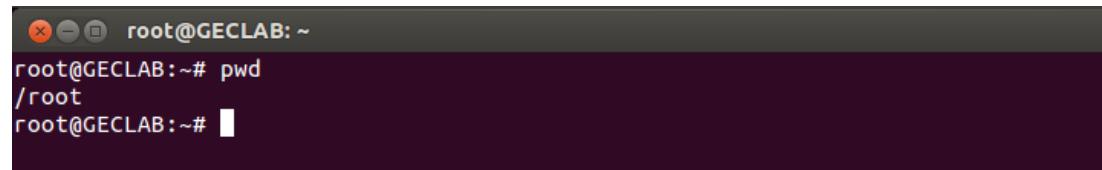
```
root@GECLAB:~# ls
busybox-1.19.2          norman      yaffs2           模板   文档   桌面
busybox-1.19.2-x210.tar.bz2 rootfs      yaffs2-source.tar  视频   下载
log_123.log               rootfs.img  公共的          图片   音乐
root@GECLAB:~# clear
```

命令执行之后的结果：



```
root@GECLAB:~#
```

打印当前路径—pwd



A screenshot of a terminal window titled "root@GECLAB: ~". The window contains the following text:
root@GECLAB:~# pwd
/root
root@GECLAB:~# █

The terminal has a dark background with white text. The title bar is dark grey with white text. The cursor is represented by a small white square at the end of the last line.

改变路径—cd

路径（目录、文件位置）的描述方式：

相对路径 – 从当前所在的目录出发描述路径的方式，不以/开始的就是相对路径。

绝对路径 – 从根目录出发描述路径的方法，以/开始的就是绝对路径。

几个特殊的路径：

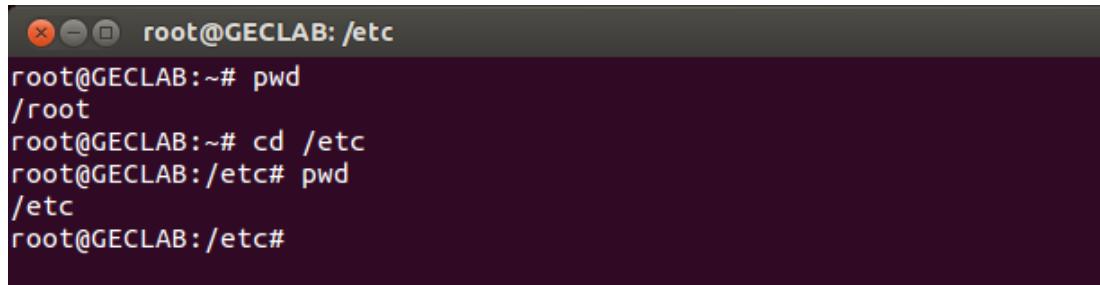
/：根目录

..：上一级目录

.：当前目录

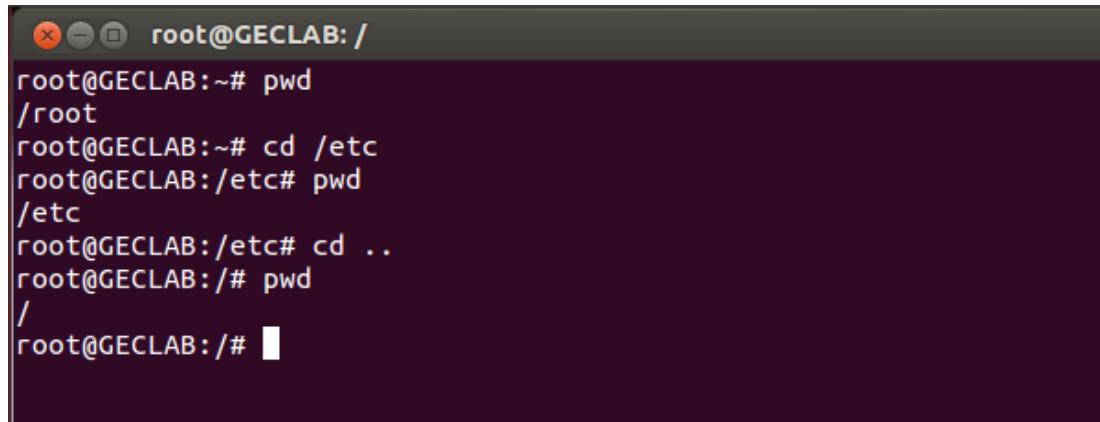
~：用户主目录

改变路径—cd



```
root@GECLAB: /etc
root@GECLAB:~# pwd
/root
root@GECLAB:~# cd /etc
root@GECLAB:/etc# pwd
/etc
root@GECLAB:/etc#
```

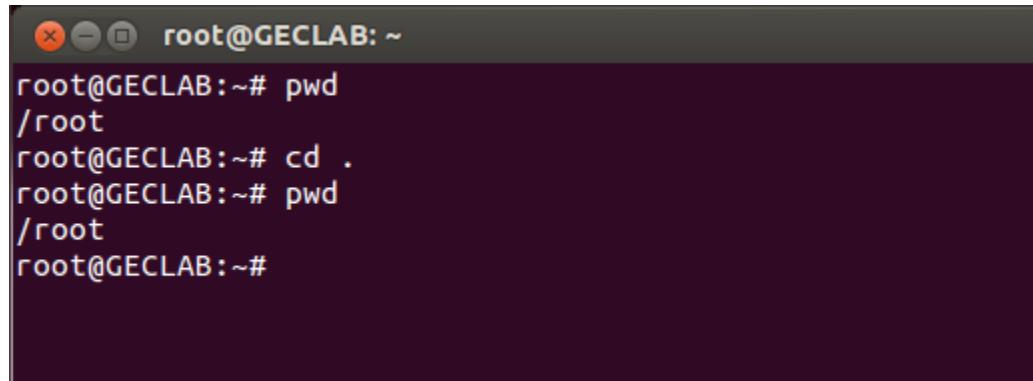
使用上一级目录（..）



```
root@GECLAB: /
root@GECLAB:~# pwd
/root
root@GECLAB:~# cd /etc
root@GECLAB:/etc# pwd
/etc
root@GECLAB:/etc# cd ..
root@GECLAB:/# pwd
/
root@GECLAB:/#
```

改变路径—cd

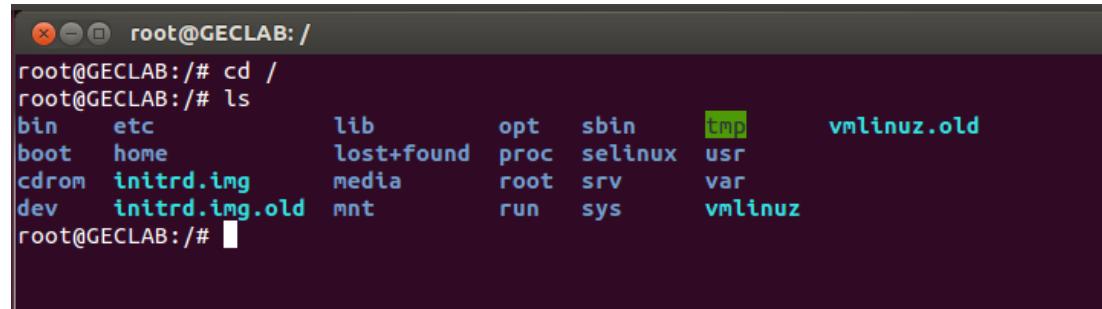
使用当前路径（..）



```
root@GECLAB:~# pwd
/root
root@GECLAB:~# cd .
root@GECLAB:~# pwd
/root
root@GECLAB:~#
```

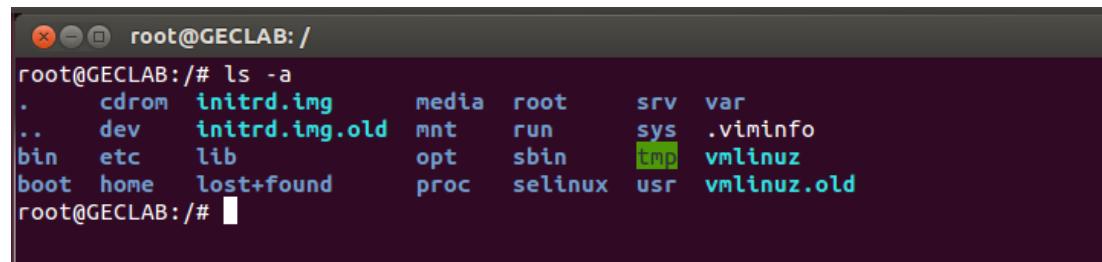
显示内容—ls

显示根目录下面的内容：



```
root@GECLAB: /  
root@GECLAB:/# cd /  
root@GECLAB:/# ls  
bin  etc      lib      opt    sbin   tmp      vmlinuz.old  
boot home    lost+found  proc  selinux  usr  
cdrom initrd.img  media    root   srv    var  
dev   initrd.img.old  mnt     run    sys    vmlinuz  
root@GECLAB:/#
```

参数1：ls -a 显示全部的内容，包括隐藏文件



```
root@GECLAB: /  
root@GECLAB:/# ls -a  
.  cdrom  initrd.img  media  root    srv  var  
..  dev    initrd.img.old  mnt   run    sys   .viminfo  
bin  etc   lib       opt    sbin   tmp  vmlinuz  
boot home  lost+found  proc   selinux  usr  vmlinuz.old  
root@GECLAB:/#
```

显示内容—ls

参数2：详细形式显示文件 ls -l

```
root@GECLAB:/# ls -l
总用量 85
drwxr-xr-x  2 root root  4096 10月 28  2013 bin
drwxr-xr-x  4 root root  1024 10月 29  2013 boot
drwxr-xr-x  2 root root  4096 10月 28  2013 cdrom
drwxr-xr-x  15 root root  4340 7月   8 16:44 dev
drwxr-xr-x 130 root root 12288 7月   8 16:55 etc
drwxr-xr-x  3 root root  4096 3月   3 11:14 home
lrwxrwxrwx  1 root root    33 10月 28  2013 initrd.img -> /boot/initrd.img-3.5.
0-42-generic
lrwxrwxrwx  1 root root    32 10月 28  2013 initrd.img.old -> boot/initrd.img-3.
5.0-23-generic
drwxr-xr-x  22 root root  4096 10月 29  2013 lib
drwx----- 2 root root 16384 10月 28  2013 lost+found
drwxr-xr-x  3 root root  4096 10月 29  2013 media
drwxr-xr-x  3 root root  4096 10月 29  2013 mnt
drwxr-xr-x  3 root root  4096 3月   3 11:23 opt
dr-xr-xr-x 180 root root     0 7月   8 16:43 proc
drwx----- 28 root root  4096 7月   8 16:55 root
drwxr-xr-x  23 root root  900 7月   8 17:22 run
drwxr-xr-x  2 root root  4096 10月 29  2013 sbin
drwxr-xr-x  2 root root  4096 3月   5  2012 selinux
drwxr-xr-x  2 root root  4096 2月   14  2013 srv
```

d rwxr-xr-x 4 user nobody 1024 Feb 22 09:20 a.txt

①② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

显示内容—ls

d rwx r-x r-x 4 user nobody 1024 Feb 22 09:20 a.txt

①② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

文件类型：目录（“d”）；普通文件（“-”）；链接文件（“l”）

② 文件拥有者对该文件的访问权限：r：读 w：写 x：执行 -：无相应权限

③ 同组者对该文件的访问权限

④ 其他人对该文件的访问权限

⑤ 文件的硬链接数（参见“ln”命令）

⑥ 文件的拥有者

⑦ 文件的属组

⑧ 文件大小（单位：byte）

⑨ 文件的最后一次修改时间

⑩ 文件名

r 读权限

w 写权限，对目录来说，可生成文件与子目录或删除文件与子目录

x 执行权限，对目录来说，可查找该目录下内容

- 没有权限

如： -rw -r- -r--

拥 同 其

有 组 它

者 人 人

其中 拥有者- 生成文件时登陆的，权限最高，u表示

同组人- 系统管理员分配的同组的一个或几个人，g表示

其他人- 除拥有者，同组人外的人，o表示

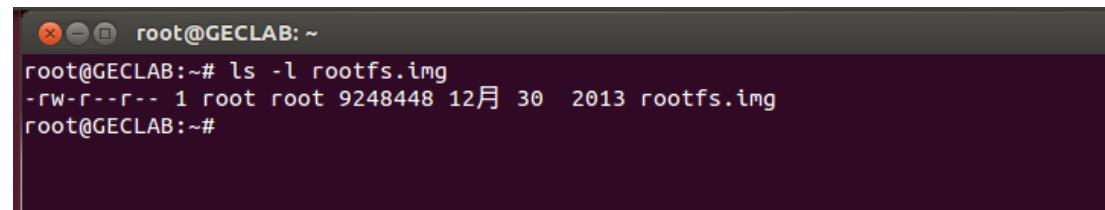
所有人- 包括拥有者、同组人及其他，a表示

修改权限——chmod

1: **符号模式**: 总结起来符号模式可以总结为下面的这个表达式:

chmod u/g/o/a = /+/- r/w/x filename

u ---所有者
g ---同组用户
o ---其他用户
a ---所有用
+ ---增加权限
- ---减去权限
= ---最终权限
r ---读取权限
w ---写入权限
x ---执行权限



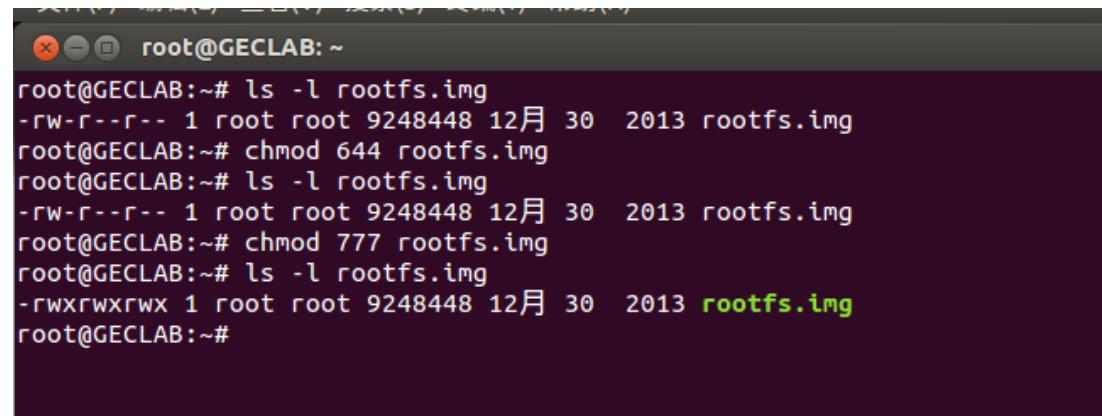
```
root@GECLAB: ~
root@GECLAB:~# ls -l rootfs.img
-rw-r--r-- 1 root root 9248448 12月 30 2013 rootfs.img
root@GECLAB:~#
```

修改权限——chmod

绝对模式：

相对的文件或者目录的权限由三位数字表示，从左向右，第一位是代表文件所有者的权限，第二位代表同组其他用户的权限，第三位是其他用户的权限。而每一个数字都是1, 2, 4三位数字组成的和，可以是0, 1, 2, 3, 4, 5, 6, 7。

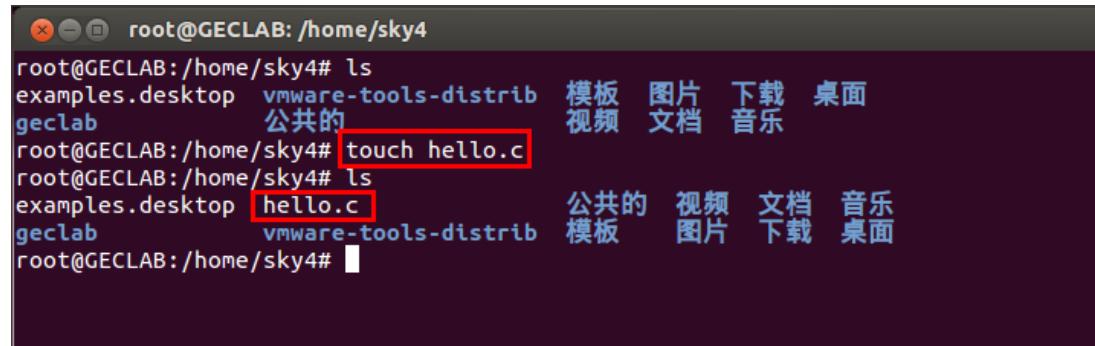
- 1 —— 执行权限
- 2 —— 写入权限
- 3 —— 读取权限



```
root@GECLAB:~# ls -l rootfs.img
-rw-r--r-- 1 root root 9248448 12月 30 2013 rootfs.img
root@GECLAB:~# chmod 644 rootfs.img
root@GECLAB:~# ls -l rootfs.img
-rw-r--r-- 1 root root 9248448 12月 30 2013 rootfs.img
root@GECLAB:~# chmod 777 rootfs.img
root@GECLAB:~# ls -l rootfs.img
-rwxrwxrwx 1 root root 9248448 12月 30 2013 rootfs.img
root@GECLAB:~#
```

文件创建命令—touch

例子命令： touch filename



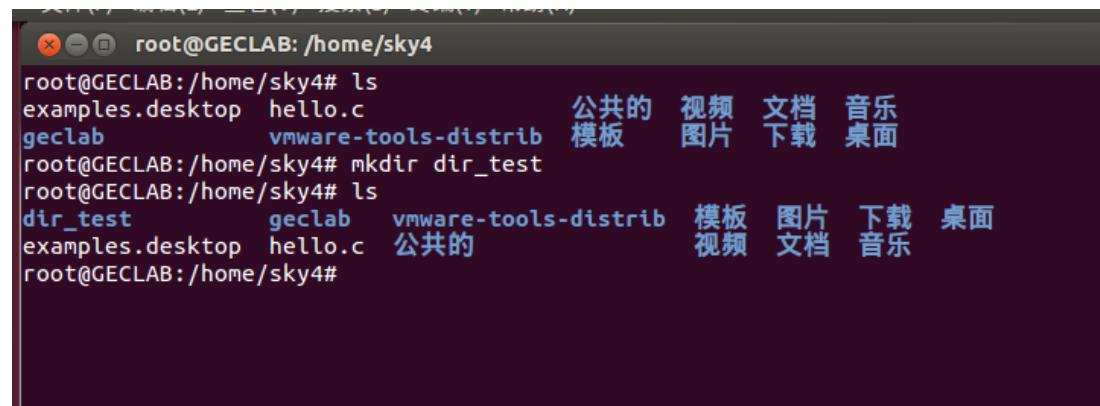
A screenshot of a Linux terminal window titled "root@GECLAB: /home/sky4". The terminal shows the following session:

```
root@GECLAB:/home/sky4# ls
examples.desktop  vmware-tools-distrib  模板  图片  下载  桌面
geclab            公共的                视频  文档  音乐
root@GECLAB:/home/sky4# touch hello.c
root@GECLAB:/home/sky4# ls
examples.desktop  hello.c             公共的  视频  文档  音乐
geclab           vmware-tools-distrib  模板    图片  下载  桌面
root@GECLAB:/home/sky4#
```

The command `touch hello.c` is highlighted with a red box. The resulting file "hello.c" is also highlighted with a red box in the second `ls` command output.

创建目录命令—mkdir

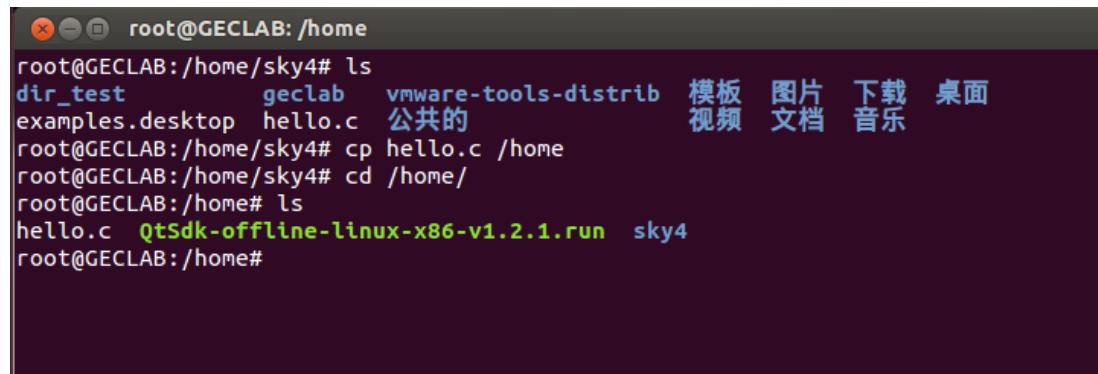
例子命令 mkdir dir_test



```
root@GECLAB:/home/sky4# ls
examples.desktop  hello.c      公共的  视频  文档  音乐
geclab           vmware-tools-distrib 模板  图片  下载  桌面
root@GECLAB:/home/sky4# mkdir dir_test
root@GECLAB:/home/sky4# ls
dir_test        geclab      vmware-tools-distrib 模板  图片  下载  桌面
examples.desktop  hello.c  公共的      视频  文档  音乐
root@GECLAB:/home/sky4#
```

文件拷贝命令—cp

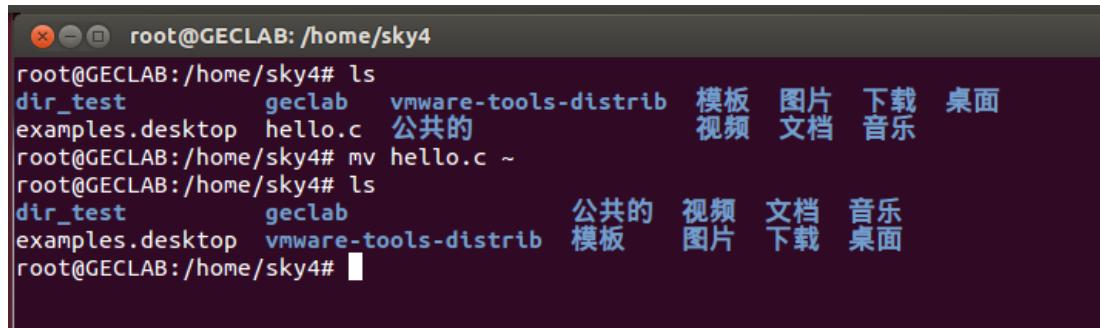
例子命令 cp hello.c /home



```
root@GECLAB: /home
root@GECLAB: /home/sky4# ls
dir_test      geclab  vmware-tools-distrib 模板 图片 下载 桌面
examples.desktop  hello.c 公共的          视频 文档 音乐
root@GECLAB:/home/sky4# cp hello.c /home
root@GECLAB:/home/sky4# cd /home/
root@GECLAB:/home# ls
hello.c QtSdk-offline-linux-x86-v1.2.1.run sky4
root@GECLAB:/home#
```

文件移动命令—mv

例子命令 mv hello.c ~



```
root@GECLAB: /home/sky4# ls
dir_test      geclab  vmware-tools-distrib 模板 图片 下载 桌面
examples.desktop  hello.c 公共的          视频 文档 音乐
root@GECLAB:/home/sky4# mv hello.c ~
root@GECLAB:/home/sky4# ls
dir_test      geclab          公共的  视频  文档  音乐
examples.desktop  vmware-tools-distrib 模板  图片  下载  桌面
root@GECLAB:/home/sky4#
```

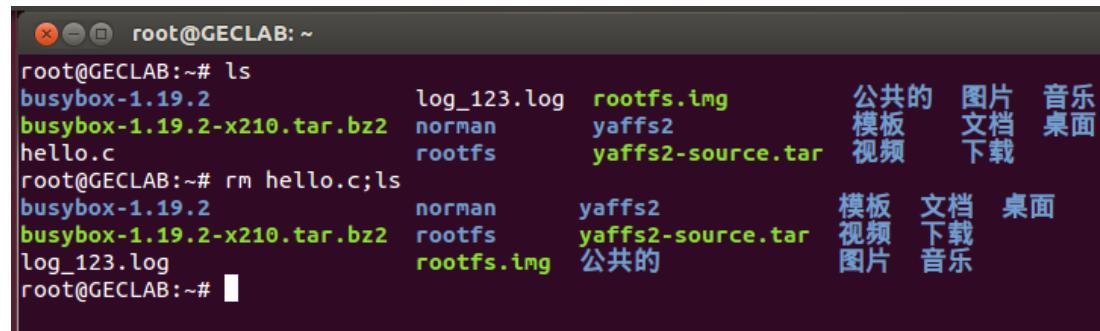
观察程序的运行结果可以发现，本来在/home/sky4目录下的hello.c文件已经被移动到用户的主目录下面去了。



```
root@GECLAB: ~
root@GECLAB:/home/sky4# ls
dir_test      geclab  vmware-tools-distrib 模板 图片 下载 桌面
examples.desktop  hello.c 公共的          视频 文档 音乐
root@GECLAB:/home/sky4# mv hello.c ~
root@GECLAB:/home/sky4# ls
dir_test      geclab          公共的  视频  文档  音乐
examples.desktop  vmware-tools-distrib 模板  图片  下载  桌面
root@GECLAB:/home/sky4# cd /root
root@GECLAB:~# ls
busybox-1.19.2          log_123.log  rootfs.img          公共的  图片  音乐
busybox-1.19.2-x210.tar.bz2  norman    yaffs2            模板  文档  桌面
hello.c                  rootfs     yaffs2-source.tar  视频  下载
root@GECLAB:~#
```

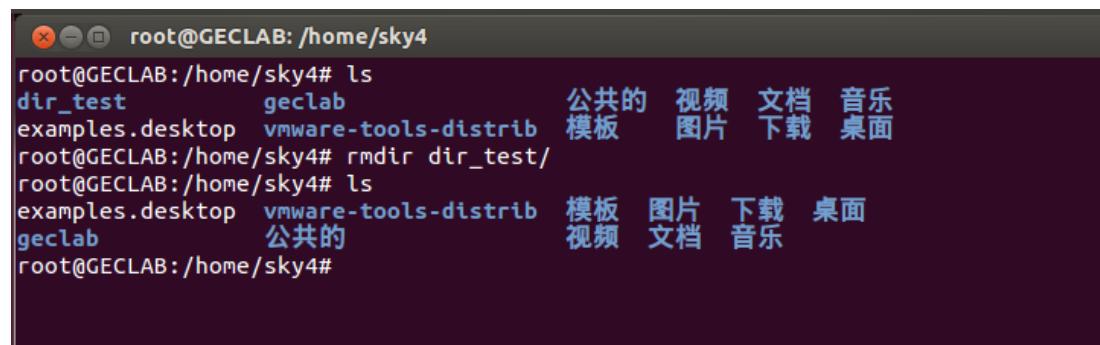
删除命令rm和rmdir

我们首先来看一下命令 rm



```
root@GECLAB:~# ls
busybox-1.19.2      log_123.log  rootfs.img    公共的  图片  音乐
busybox-1.19.2-x210.tar.bz2  norman     yaffs2        模板  文档  桌面
hello.c              rootfs      yaffs2-source.tar  视频  下载
root@GECLAB:~# rm hello.c;ls
busybox-1.19.2      norman     yaffs2        模板  文档  桌面
busybox-1.19.2-x210.tar.bz2  rootfs      yaffs2-source.tar  视频  下载
log_123.log          rootfs.img   公共的      图片  音乐
root@GECLAB:~#
```

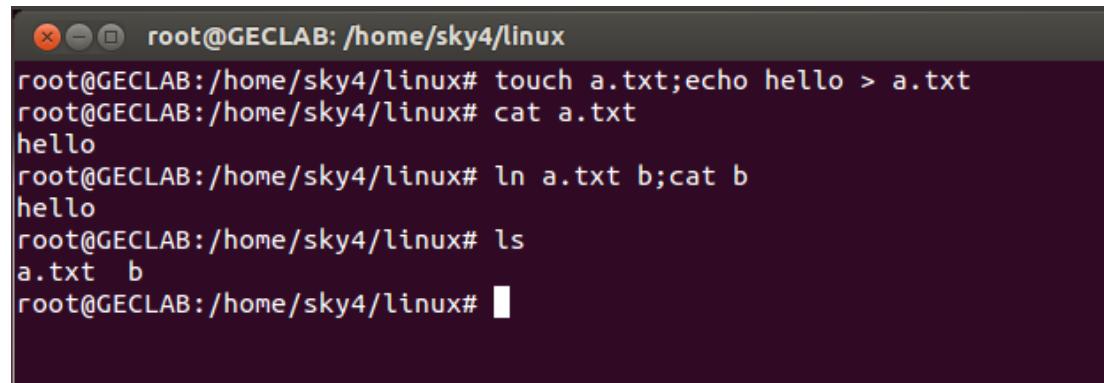
接着看一下删除目录的命令rmdir



```
root@GECLAB:/home/sky4#
root@GECLAB:/home/sky4# ls
dir_test      geclab      公共的  视频  文档  音乐
examples.desktop  vmware-tools-distrib  模板  图片  下载  桌面
root@GECLAB:/home/sky4# rmdir dir_test/
root@GECLAB:/home/sky4# ls
examples.desktop  vmware-tools-distrib  模板  图片  下载  桌面
geclab          公共的      视频  文档  音乐
root@GECLAB:/home/sky4#
```

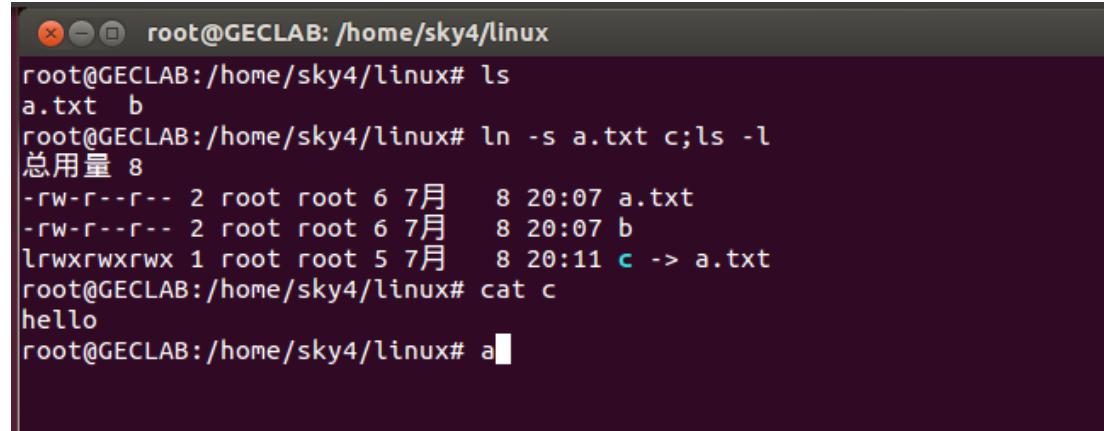
创建连接 —ln

命令ln用来创建一个硬连接或者软连接，下面使用一下ln命令：



```
root@GECLAB: /home/sky4/linux
root@GECLAB:/home/sky4/linux# touch a.txt;echo hello > a.txt
root@GECLAB:/home/sky4/linux# cat a.txt
hello
root@GECLAB:/home/sky4/linux# ln a.txt b;cat b
hello
root@GECLAB:/home/sky4/linux# ls
a.txt  b
root@GECLAB:/home/sky4/linux#
```

图中的第一条命令创建文件a. txt并向其中写入数据hello，第二条命令创建了一个连接b指向a. txt。使用命令cat观察b中的内容，发现和a. txt中的内容相同。上面的例子里面，我们使用命令ln创建了一个硬链接指向文件a. txt，下面我们来介绍另外一个参数-s，用来创建一个软连接指向一个文件；

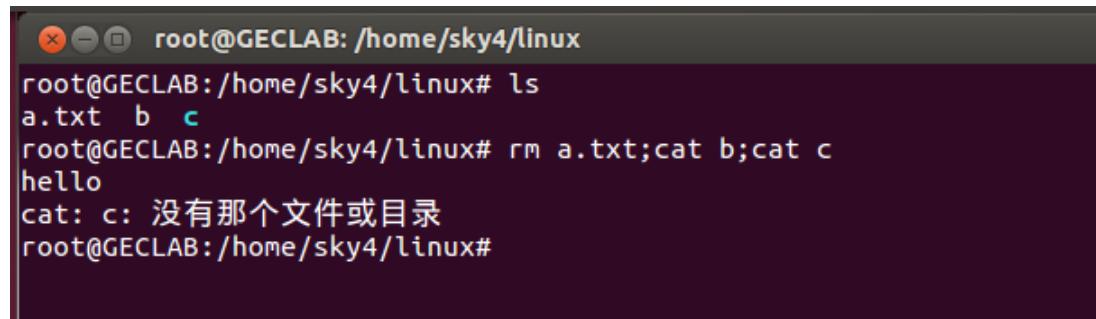


```
root@GECLAB: /home/sky4/linux
root@GECLAB:/home/sky4/linux# ls
a.txt  b
root@GECLAB:/home/sky4/linux# ln -s a.txt c;ls -l
总用量 8
-rw-r--r-- 2 root root 6 7月  8 20:07 a.txt
-rw-r--r-- 2 root root 6 7月  8 20:07 b
lrwxrwxrwx 1 root root 5 7月  8 20:11 c -> a.txt
root@GECLAB:/home/sky4/linux# cat c
hello
root@GECLAB:/home/sky4/linux# a
```

创建连接 —ln

我们创建了一个软连接c指向a.txt，那么硬链接和软连接有什么区别呢？

硬链接本质上硬盘上文件数据的另外一个访问路径，删除本来的a.txt，系统关闭了a.txt对硬盘数据访问的路径，不影响硬链接b，而软连接里面只保存了目标文件的位置，当删除目标文件的时候，软连接文件也失去了访问硬盘数据的通路。



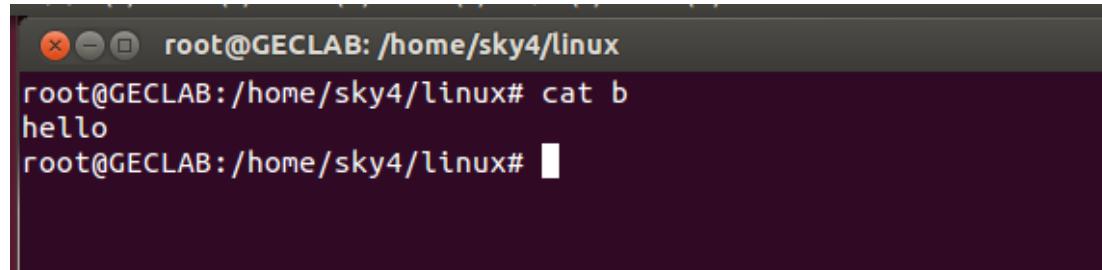
The screenshot shows a terminal window with the following session:

```
root@GECLAB: /home/sky4/linux
root@GECLAB: /home/sky4/linux# ls
a.txt b c
root@GECLAB: /home/sky4/linux# rm a.txt;cat b;cat c
hello
cat: c: 没有那个文件或目录
root@GECLAB: /home/sky4/linux#
```

The terminal window has a dark background and light-colored text. It shows the user is root and is in the directory /home/sky4/linux. The user runs 'ls' to list files, then 'rm a.txt' to delete the file 'a.txt'. After this, they run 'cat b' and 'cat c' to read the contents of files 'b' and 'c'. The output for 'cat b' is 'hello', but for 'cat c', it says 'cat: c: 没有那个文件或目录' (cat: c: No such file or directory), indicating that the soft link 'c' no longer points to a valid file.

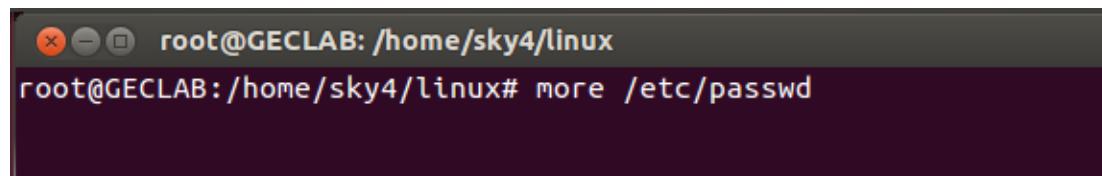
文件查看命令——cat more head tail

关于命令cat，我们已经使用过很多次了，我们再使用命令cat看一下/home/sky4/linux/b



```
root@GECLAB: /home/sky4/linux
root@GECLAB: /home/sky4/linux# cat b
hello
root@GECLAB: /home/sky4/linux#
```

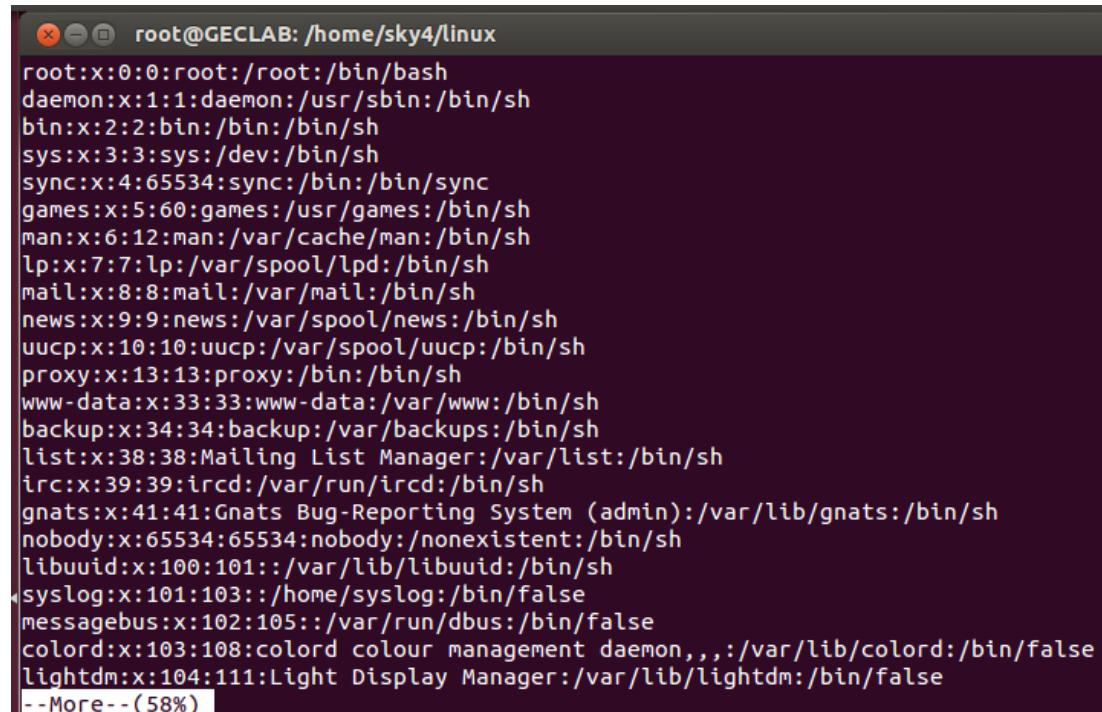
命令more 也是来查看文件的内容的，那么more和命令cat有什么区别呢？命令more也是分屏查看文件的内容，也可以说一行一行地显示内容，使用命令more来看/etc/passwd文件：



```
root@GECLAB: /home/sky4/linux
root@GECLAB: /home/sky4/linux# more /etc/passwd
```

文件查看命令——cat more head tail

(续) 按下回车之后, 命令的执行结果:

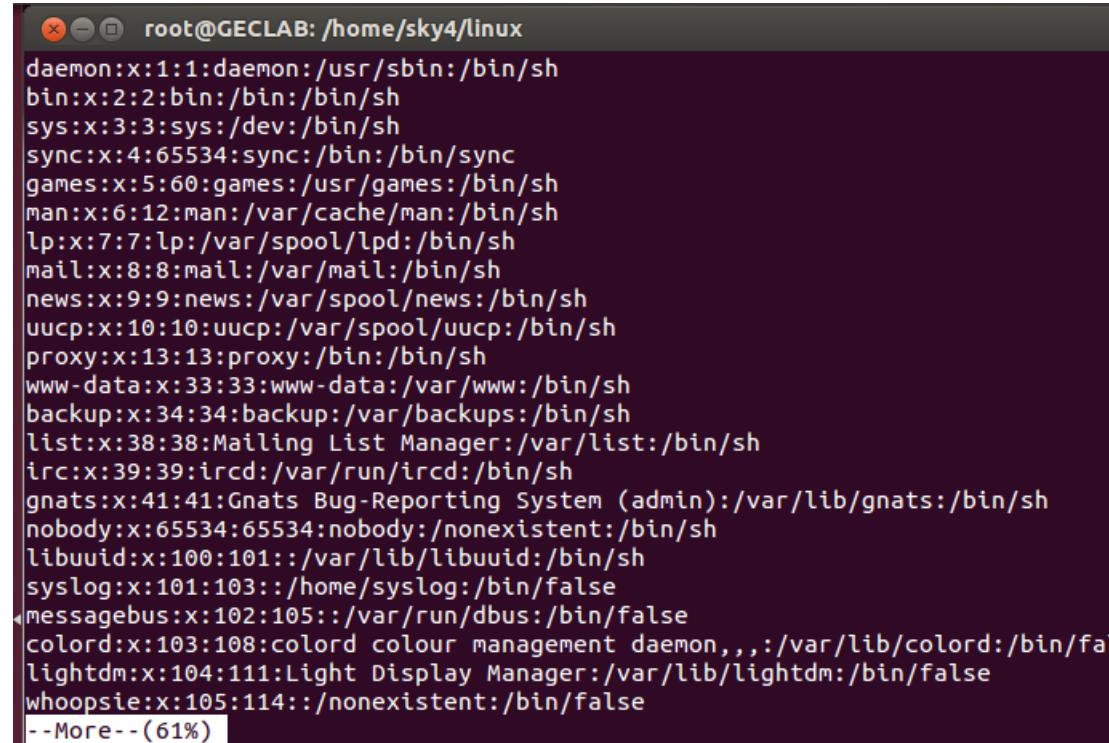


The screenshot shows a terminal window with the title "root@GECLAB: /home/sky4/linux". The window displays the contents of the /etc/passwd file. The output is as follows:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
--More--(58%)
```

文件查看命令——cat more head tail

(续) 在这种情况下，使用回车按键可以实现查看下一行内容：

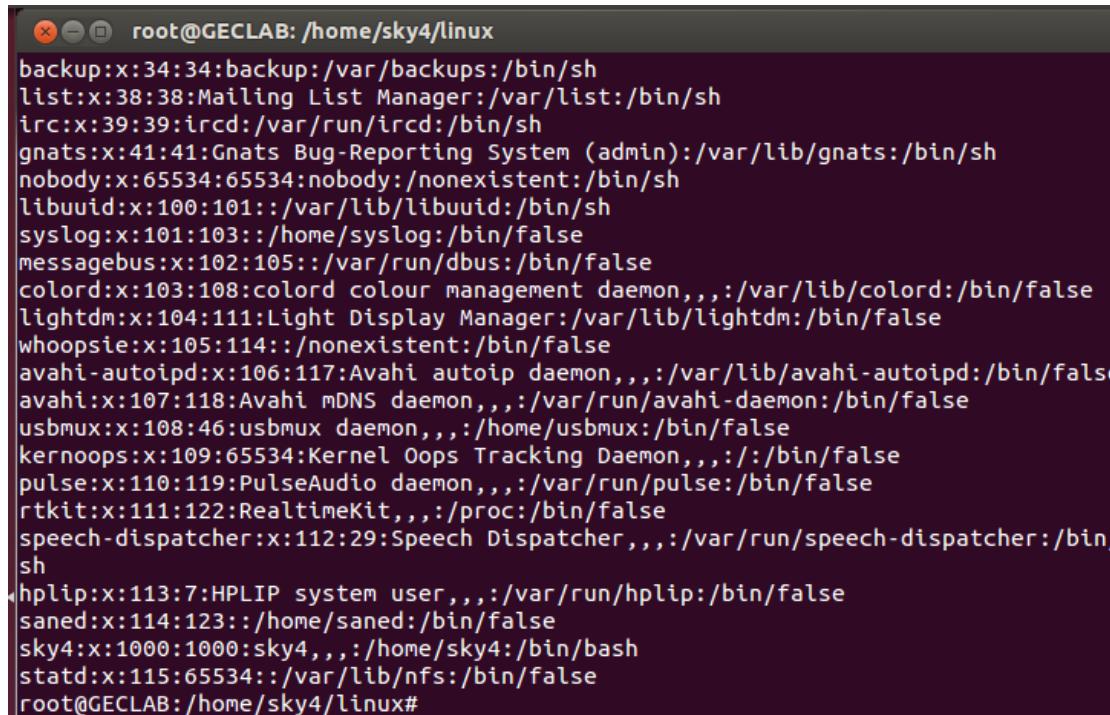


```
root@GECLAB: /home/sky4/linux
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:105:114::/nonexistent:/bin/false
--More-- (61%)
```

按键q退出。

文件查看命令——cat more head tail

(续) 空格按键实现查看下一屏内容:



A screenshot of a terminal window titled "root@GECLAB: /home/sky4/linux". The window displays a list of processes, each consisting of a colon-separated list of fields. The fields typically include the process ID (PID), type (e.g., 'x' for a daemon), user (e.g., 'root'), command name (e.g., 'hplip'), and the full path to the executable (e.g., '/var/run/hplip'). The list is quite long, showing many system daemons and user sessions. The terminal has a dark background with light-colored text.

```
root@GECLAB: /home/sky4/linux
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:105:114::/nonexistent:/bin/false
avahi-autoipd:x:106:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:107:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
usbmux:x:108:46:usbmux daemon,,,:/home/usbmux:/bin/false
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:110:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:111:122:RealtimeKit,,,:/proc:/bin/false
speech-dispatcher:x:112:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
sh
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:x:114:123::/home/saned:/bin/false
sky4:x:1000:1000:sky4,,,:/home/sky4:/bin/bash
statd:x:115:65534::/var/lib/nfs:/bin/false
root@GECLAB:/home/sky4/linux#
```

文件查看命令——cat more head tail less

less命令与 more 类似，但是比 more 更好的是，它可以往前翻页：less

```
gec@ubuntu:~$ less /etc/passwd
```



```
gec@ubuntu:~
```

```
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:105:114::/nonexistent:/bin/false
avahi-autoipd:x:106:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:107:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
usbmux:x:108:46:usbmux daemon,,,:/home/usbmux:/bin/false
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:110:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:111:122:RealtimeKit,,,:/proc:/bin/false
speech-dispatcher:x:112:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
sh
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:x:114:123::/home/saned:/bin/false
gec:x:1000:1000:www.yueqian.com.cn,,,:/home/gec:/bin/bash
statd:x:115:65534::/var/lib/nfs:/bin/false
(END)
```

文件查看命令——cat more head tail less

(续) less命令常用快捷键:

- 空格键 : 向下翻动一页 ;
- [pagedown] : 向下翻动一页 ;
- [pageup] : 向上翻动一页 ;
- /字符串 : 向下搜寻『字符串』的功能 ;
- ?字符串 : 向上搜寻『字符串』的功能 ;
- n : 重复前一个搜寻 (与 / 或 ? 有关 !)
- N : 反向的重复前一个搜寻 (与 / 或 ? 有关 !)
- q : 离开 less 这个程序 ;

寻找特定档案-whereis

whereis命令用于寻找特定文档

```
gec@ubuntu:~$ whereis cp
cp: /bin/cp /usr/share/man/man1/cp.1.gz /usr/share/man/man1/cp.1posix.gz
gec@ubuntu:~$ whereis mv
mv: /bin/mv /usr/share/man/man1/mv.1posix.gz /usr/share/man/man1/mv.1.gz
gec@ubuntu:~$ whereis clear
clear: /usr/bin/clear /usr/bin/X11/clear /usr/share/man/man1/clear.1.gz
```

压缩文件和读取压缩文件: gzip, zcat

减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip是在Linux系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。gzip不仅可以用来压缩大的、较少使用的文件以节省磁盘空间，还可以和tar命令一起构成Linux操作系统中比较流行的压缩文件格式。gzip命令对文本文件有**60%~70%的压缩率**。

1. 命令格式:

gzip[参数][文件或者目录]

2. 命令功能:

gzip是个使用广泛的压缩程序，文件经它压缩过后，其名称后面会多出".gz"的扩展名。

压缩文件和读取压缩文件: gzip, zcat

(续) 3. 命令参数:

- a或--ascii 使用ASCII文字模式。
- c或--stdout或--to-stdout 把压缩后的文件输出到标准输出设备，不去更动原始文件。
- d或--decompress或----uncompress 解开压缩文件。
- f或--force 强行压缩文件。不理会文件名称或硬连接是否存在以及该文件是否为符号连接。
- h或--help 在线帮助。
- l或--list 列出压缩文件的相关信息。
- L或--license 显示版本与版权信息。
- n或--no-name 压缩文件时，不保存原来的文件名称及时间戳记。
- N或--name 压缩文件时，保存原来的文件名称及时间戳记。
- q或--quiet 不显示警告信息。
- r或--recursive 递归处理，将指定目录下的所有文件及子目录一并处理。
- S<压缩字尾字符串>或----suffix<压缩字尾字符串> 更改压缩字尾字符串。
- t或--test 测试压缩文件是否正确无误。
- v或--verbose 显示指令执行过程。
- V或--version 显示版本信息。
- num 用指定的数字num调整压缩的速度，-1或--fast表示最快压缩方法（低压缩比），-9或--best表示最慢压缩方法（高压缩比）。系统缺省值为6。

压缩文件和读取压缩文件: gzip, zcat

gzip的压缩与解压:

```
gec@ubuntu:~/io$ ls
buff_types.c  buff_types.o  gettimeofday.c  line      lseek.c      read.c
buff_types.c~ f_pos.c        gettimeofday.o  lseek2.c  printf_color_demo  times
gec@ubuntu:~/io$ gzip -v gettimeofday.c
gettimeofday.c: 53.8% -- replaced with gettimeofday.c.gz
gec@ubuntu:~/io$ ls
buff_types.c  buff_types.o  gettimeofday.c.gz  line      lseek.c      read.c
buff_types.c~ f_pos.c        gettimeofday.o  lseek2.c  printf_color_demo  times
gec@ubuntu:~/io$ gzip -d gettimeofday.c.gz
gec@ubuntu:~/io$ ls
buff_types.c  buff_types.o  gettimeofday.c  line      lseek.c      read.c
buff_types.c~ f_pos.c        gettimeofday.o  lseek2.c  printf_color_demo  times
gec@ubuntu:~/io$
```

压缩文件和读取压缩文件: gzip, zcat

zcat查看压缩包里面的内容:

```
gec@ubuntu:~/io$ gzip -v gettimeofday.c
gettime.c:      53.8% -- replaced with gettimeofday.c.gz
gec@ubuntu:~/io$ zcat gettimeofday.c.gz
#include <stdio.h>
#include <sys/stat.h>
#include <time.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>

#define BUFSIZE 64

static int line_num = 0;

FILE *Fopen(const char *path, const char *mode)
{
    FILE *fp.
```



压缩和解压缩-tar

语法：tar [主选项+辅选项] 文件或者目录

使用该命令时，主选项是必须要有的，它告诉tar要做什么事情，辅选项是辅助使用的，可以选用。

主选项：

c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。相当于打包。

x 从档案文件中释放文件。相当于拆包。

t 列出档案文件的内容，查看已经备份了哪些文件。

特别注意，在参数的下达中，c/x/t 仅能存在一个！不可同时存在！因为不可能同时压缩与解压缩。

辅助选项：

-z : 是否同时具有 gzip 的属性？亦即是否需要用 gzip 压缩或解压？一般格式为 xx.tar.gz 或 xx.tgz

-j : 是否同时具有 bzip2 的属性？亦即是否需要用 bzip2 压缩或解压？一般格式为 xx.tar.bz2

-v : 压缩的过程中显示文件！这个常用

-f : 使用档名，请留意，在 f 之后要立即接档名喔！不要再加其他参数！

-p : 使用原文件的原来属性（属性不会依据使用者而变）

--exclude FILE: 在压缩的过程中，不要将 FILE 打包！

压缩和解压缩-tar

范例一：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar

【不压缩】 tar -cvf /tmp/etc.tar /etc

```
root@ubuntu:~# tar -cvf /tmp/etc.tar /etc
```

```
/etc/ssl/certs/cacert.org.pem
/etc/ssl/certs/e775ed2d.0
root@ubuntu:/# cd /
root@ubuntu:/# cd tmp
root@ubuntu:/tmp# ls
at-spi2          pulse-NFvdKVaTVmXx  unity_support_test.0  vmware-root
etc.tar          pulse-PKdhtXMmr18n  VMwareDnD        vmware-root-3630090662
keyring-LXyFKN  ssh-vUcNmIPD2058  vmware-gec
root@ubuntu:/tmp# ls -l etc.tar
-rw-rw-r-- 1 gec gec 10147840 Jul 15 09:14 etc.tar
```

压缩和解压缩-tar

范例二：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar.gz

【 gzip压缩】 tar -zcvf /tmp/etc.tar.gz /etc

```
root@ubuntu:/tmp
root@ubuntu:/tmp# tar -zcvf /tmp/etc.tar.gz /etc
```

```
/etc/ssl/certs/f060240e.0
/etc/ssl/certs/SecureSign_RootCA11.pem
/etc/ssl/certs/0b759015.0
/etc/ssl/certs/111e6273.0
/etc/ssl/certs/cacert.org.pem
/etc/ssl/certs/e775ed2d.0
root@ubuntu:/tmp# ls -l
total 12216
drwxrwxrwt 2 lightdm lightdm    4096 Jul 15 05:27 at-spi2
-rw-rw-r-- 1 gec      gec     10147840 Jul 15 09:14 etc.tar
-rw-r--r-- 1 root     root    2322285 Jul 15 09:21 etc.tar.gz
drwx----- 2 gec      gec     4096 Jul 15 05:27 keyring-LXyFKN
drwx----- 2 gec      gec     4096 Jul 15 05:27 pulse-NFvdKVaTVmXx
drwx----- 2 lightdm lightdm    4096 Jul 15 05:27 pulse-PKdhtXMmr18n
drwx----- 2 gec      gec     4096 Jul 15 05:27 ssh-vUcNmiPD2058
-rw-rw-r-- 1 lightdm lightdm      0 Jul 15 05:27 unity_support_test.0
drwxrwxrwt 2 root     root    4096 Jul 15 05:27 VMwareDnD
drwx----- 2 gec      gec     4096 Jul 15 05:27 vmware-gec
drwxr-xr-x 2 root     root    4096 Jul 15 05:27 vmware-root
drwx----- 2 root     root    4096 Jul 15 05:27 vmware-root-3630090662
```

压缩和解压缩-tar

范例三：将etc. tar进行解压缩

【解压缩】 tar -xvf /tmp/etc. tar.gz

```
root@ubuntu:/tmp# ls  
at-spi2      keyring-LXyFKN      ssh-vUcNmIPD2058      vmware-gec  
etc.tar      pulse-NFvdKVaTVmXx  unity_support_test.0  vmware-root  
etc.tar.gz   pulse-PKdhtXMmr18n  VMwareDnD           vmware-root-3630090662  
root@ubuntu:/tmp# tar -xvf etc.tar
```

```
----  
etc/ssl/certs/f060240e.0  
etc/ssl/certs/SecureSign_RootCA11.pem  
etc/ssl/certs/0b759015.0  
etc/ssl/certs/111e6273.0  
etc/ssl/certs/cacert.org.pem  
etc/ssl/certs/e775ed2d.0  
root@ubuntu:/tmp# ls  
at-spi2  etc.tar.gz          pulse-PKdhtXMmr18n  VMwareDnD      vmware-root-3630090662  
etc     keyring-LXyFKN        ssh-vUcNmIPD2058  vmware-gec  
etc.tar  pulse-NFvdKVaTVmXx  unity_support_test.0  vmware-root
```

压缩和解压缩-tar

范例四：将etc. tar. gz进行解压缩

【解压缩】 tar -zxvf /tmp/etc.tar.gz

```
root@ubuntu:/tmp# ls
at-spi2      keyring-LXyFKN      ssh-vUcNmiPD2058      vmware-gec
etc.tar      pulse-NFvdKVaTVmXx  unity_support_test.0  vmware-root
etc.tar.gz   pulse-PKdhtXMmr18n  VMwareDnD          vmware-root-3630090662
root@ubuntu:/tmp# tar -zxvf /tmp/etc.tar.gz
```

```
etc/ssl/certs/SecureSign_RootCA11.pem
etc/ssl/certs/0b759015.0
etc/ssl/certs/111e6273.0
etc/ssl/certs/cacert.org.pem
etc/ssl/certs/e775ed2d.0
root@ubuntu:/tmp# ls
at-spi2      etc.tar.gz        pulse-PKdhtXMmr18n  VMwareDnD      vmware-root-3630090662
etc         keyring-LXyFKN    ssh-vUcNmiPD2058    vmware-gec
etc.tar     pulse-NFvdKVaTVmXx  unity_support_test.0  vmware-root
```

文件查看命令——cat more head tail

命令head和tail查看文件的开头n行和结尾n行，很多时候文件很大，而我们只希望查看文件最后几行，我们使用命令tail：

```
root@GECLAB:/home/sky4/linux# tail -5 /var/log/dmesg
[ 50.459861] type=1400 audit(1404809072.119:10): apparmor="STATUS" operation="profile_replace" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=918 comm="apparmor_parser"
[ 50.460294] type=1400 audit(1404809072.119:11): apparmor="STATUS" operation="profile_replace" name="/usr/lib/connman/scripts/dhclient-script" pid=918 comm="apparmor_parser"
[ 51.143671] microcode: Microcode Update Driver: v2.00 <tigran@aivazian.fsnet.co.uk>, Peter Oruba
[ 52.898336] NFSD: Using /var/lib/nfs/v4recovery as the NFSv4 state recovery directory
[ 52.898613] NFSD: starting 90-second grace period
root@GECLAB:/home/sky4/linux#
```

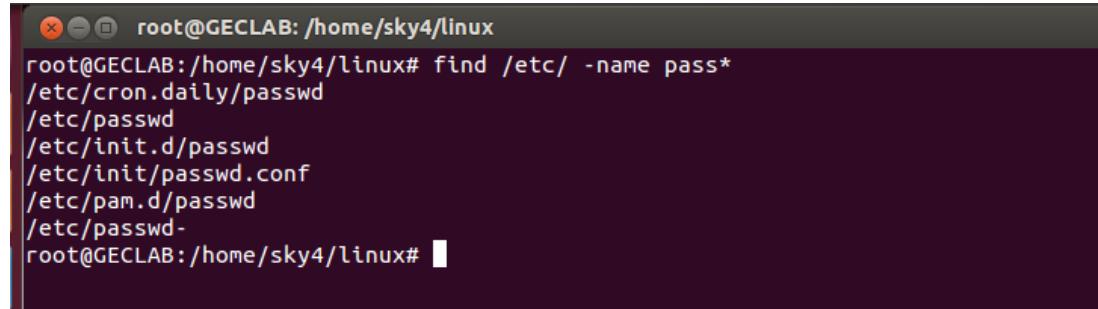
其中/var/log/dmesg文件保存了系统启动过程，内核打印的信息。

参数-5表示只观察这个文件的最后5行。同理，命令head用来查看文件的开始n行：

```
root@GECLAB:/home/sky4/linux# head -5 /var/log/dmesg
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffe0000-0x00000000ffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.4 present.
root@GECLAB:/home/sky4/linux#
```

文件查找命令—find

命令find可以根据指定的条件在指定的目录下面查找文件，其中文件 -name 比较常用，我们来观察一下命令的运行结果：



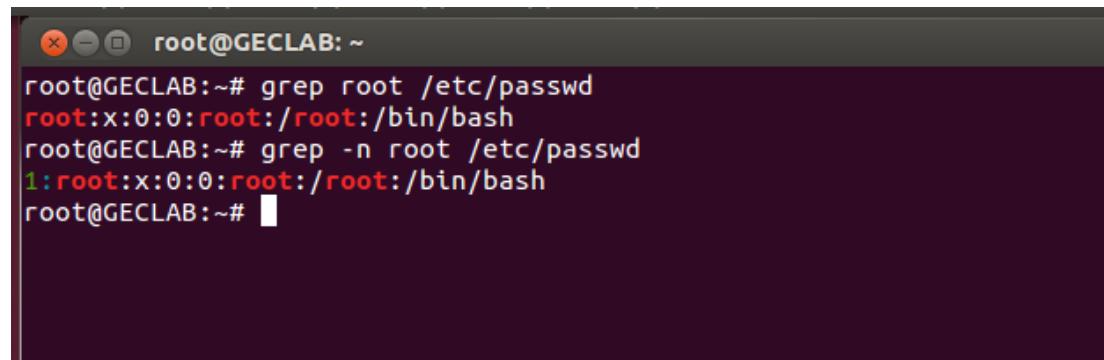
```
root@GECLAB:/home/sky4/linux
root@GECLAB:/home/sky4/linux# find /etc/ -name pass*
/etc/cron.daily/passwd
/etc/passwd
/etc/init.d/passwd
/etc/init/passwd.conf
/etc/pam.d/passwd
/etc/passwd-
root@GECLAB:/home/sky4/linux#
```

内容查找命令—grep

grep (gloal search regular expression (RE) and print out the line, 全面搜索正则表达式并把行打印出来) 是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来，Unix的grep家族包括grep、egrep和fgrep。egrep和fgrep的命令只跟grep有很小不同。

目前，我们只关心grep命令。

举例：在/etc/passwd文件中搜索和root字符有关的内容：

A screenshot of a terminal window titled "root@GECLAB: ~". The window contains the following text:

```
root@GECLAB:~# grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
root@GECLAB:~# grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
root@GECLAB:~#
```

The text is white on a dark background, with the grep command and its output clearly visible.

检测文件类型—file

file是检测文件类型命令。

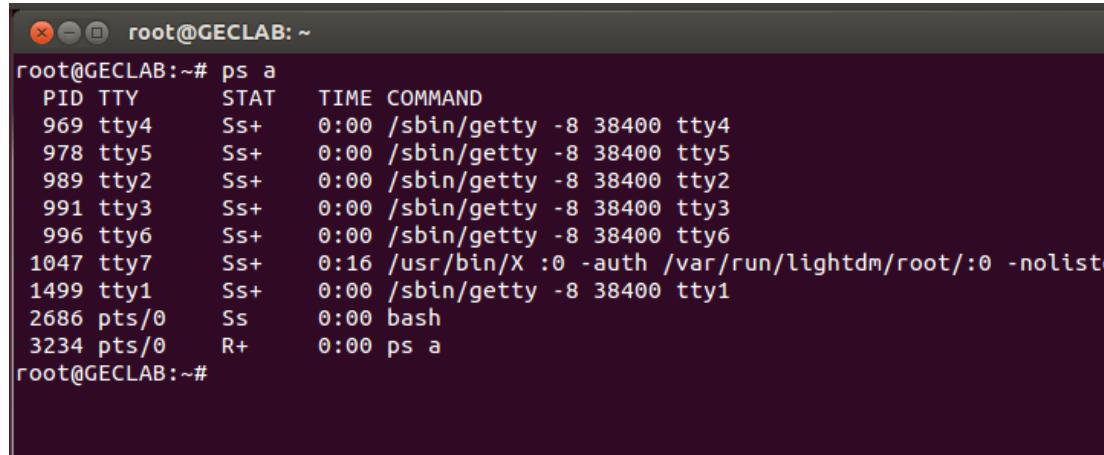
```
gec@ubuntu:~$ file 1.jpg
1.jpg: JPEG image data, JFIF standard 1.01
gec@ubuntu:~$ file b.txt
b.txt: ASCII text
gec@ubuntu:~$ file test.c
test.c: ASCII text
gec@ubuntu:~$ file test.o
test.o: ELF 32-bit LSB relocatable, ARM, version 1 (SYSV), not stripped
```

显示目标文件段大小—size

size显示一个**目标文件**或者链接库文件中的目标文件的各个段的大小命令。

```
gec@ubuntu:~$ size 1.jpg
size: 1.jpg: File format not recognized
gec@ubuntu:~$ size test.c
size: test.c: File format not recognized
gec@ubuntu:~$ size test.o
  text      data      bss      dec      hex filename
    120        0        0     120      78  test.o
```

查看进程—ps



```
root@GECLAB:~# ps a
 PID TTY      STAT   TIME COMMAND
 969 tty4    Ss+   0:00 /sbin/getty -8 38400 tty4
 978 tty5    Ss+   0:00 /sbin/getty -8 38400 tty5
 989 tty2    Ss+   0:00 /sbin/getty -8 38400 tty2
 991 tty3    Ss+   0:00 /sbin/getty -8 38400 tty3
 996 tty6    Ss+   0:00 /sbin/getty -8 38400 tty6
 1047 tty7    Ss+   0:16 /usr/bin/X :0 -auth /var/run/lightdm/root/:0 -nolisten
 1499 tty1    Ss+   0:00 /sbin/getty -8 38400 tty1
 2686 pts/0    Ss   0:00 bash
 3234 pts/0    R+   0:00 ps a
root@GECLAB:~#
```

常用选项：

ps 的参数非常多，在此仅列出几个常用的参数并大略介绍含义

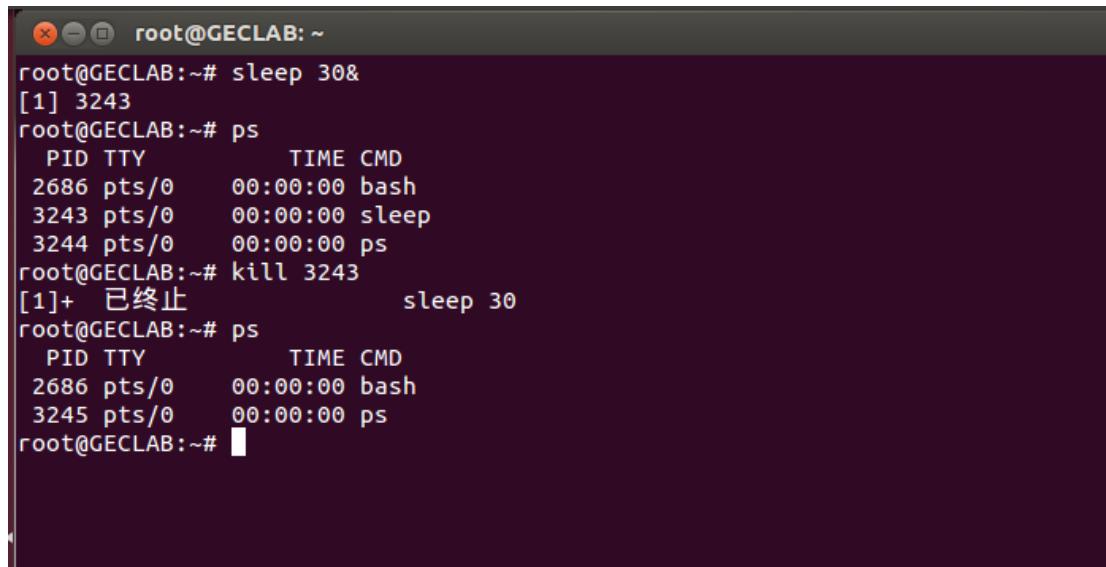
-a 列出所有的进程

-w 显示加宽可以显示更多的资讯

-au 显示较详细的资讯

-aux 显示所有包含其他使用者的进程.

杀掉指定的进程——kill



```
root@GECLAB:~# sleep 30&
[1] 3243
root@GECLAB:~# ps
  PID TTY      TIME CMD
 2686 pts/0    00:00:00 bash
 3243 pts/0    00:00:00 sleep
 3244 pts/0    00:00:00 ps
root@GECLAB:~# kill 3243
[1]+  已终止                  sleep 30
root@GECLAB:~# ps
  PID TTY      TIME CMD
 2686 pts/0    00:00:00 bash
 3245 pts/0    00:00:00 ps
root@GECLAB:~#
```

命令kill可以用来终结一个进程，其实还可以用来发送信号，我们来观察一下这个参数：

-s : 指定发送的信号。例如#kill -s SIGKILL 进程ID

杀掉指定的进程——kill

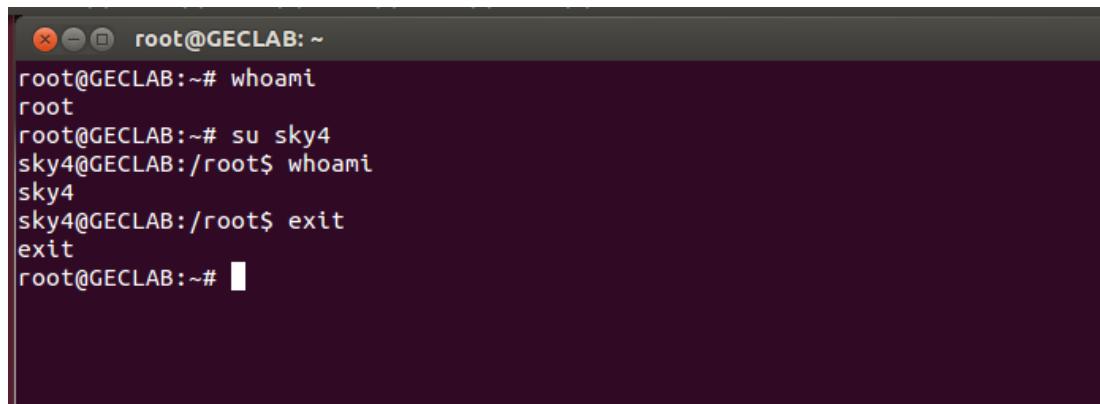


```
root@GECLAB:/home/sky4/linux# kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
root@GECLAB:/home/sky4/linux#
```

命令kill可以用来终结一个进程，其实还可以用来发送信号，我们来观察一下这个参数：

-l:查看所有信号

切换用户—su



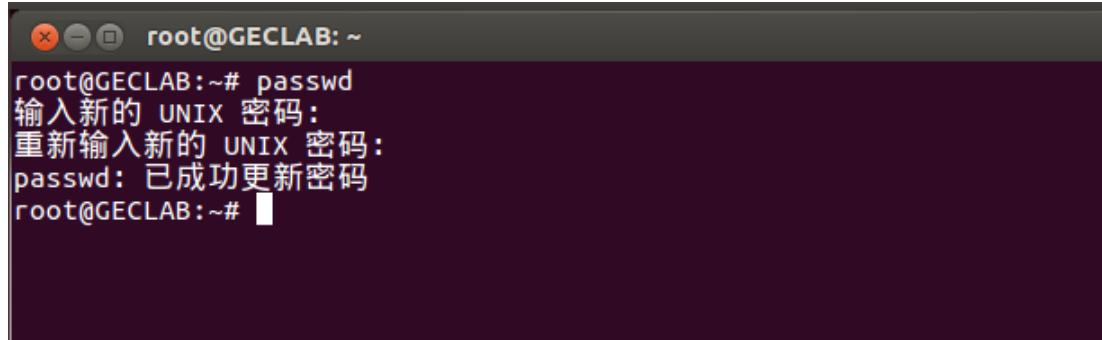
```
root@GECLAB: ~
root@GECLAB:~# whoami
root
root@GECLAB:~# su sky4
sky4@GECLAB:/root$ whoami
sky4
sky4@GECLAB:/root$ exit
exit
root@GECLAB:~#
```

临时切换到root -s

The screenshot shows a terminal window titled "gec@ubuntu: ~". The session starts with the user "gec" at the prompt. The user runs the command "whoami", which outputs "gec". Then, the user runs "sudo -s", which prompts for a password. After entering the password, the user runs "whoami" again, which now outputs "root". Finally, the user runs "exit" to return to the original "gec" prompt.

```
gec@ubuntu:~$ whoami  
gec  
gec@ubuntu:~$ sudo -s  
[sudo] password for gec:  
root@ubuntu:~# whoami  
root  
root@ubuntu:~# exit  
exit  
gec@ubuntu:~$
```

修改密码——passwd



A screenshot of a terminal window titled "root@GECLAB: ~". The window contains the following text:

```
root@GECLAB:~# passwd
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
root@GECLAB:~#
```

文件通配符

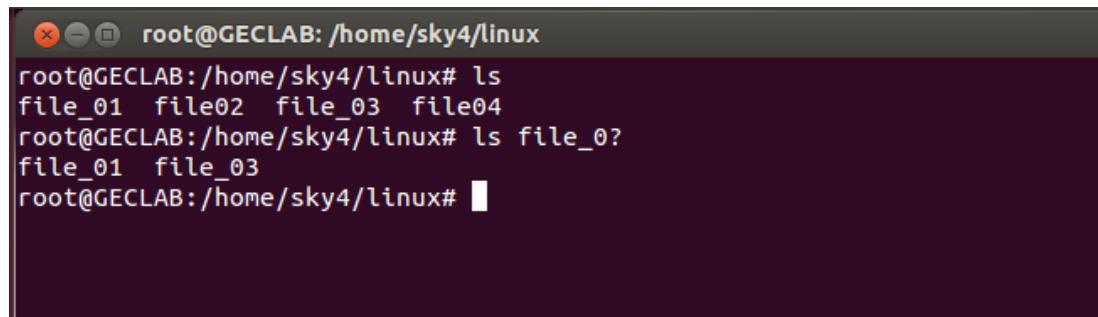
*通配任意多个任意字符

举例：创建四个文件，使用通配符*删除符合制定规则的文件：

```
root@GECLAB:/home/sky4/linux
root@GECLAB:/home/sky4/linux# touch file_01 file02 file_03 file04
root@GECLAB:/home/sky4/linux# ls
file_01  file02  file_03  file04
root@GECLAB:/home/sky4/linux# rm -f file0*
root@GECLAB:/home/sky4/linux# ls
file_01  file_03
root@GECLAB:/home/sky4/linux#
```

？ 通配一个任意字符

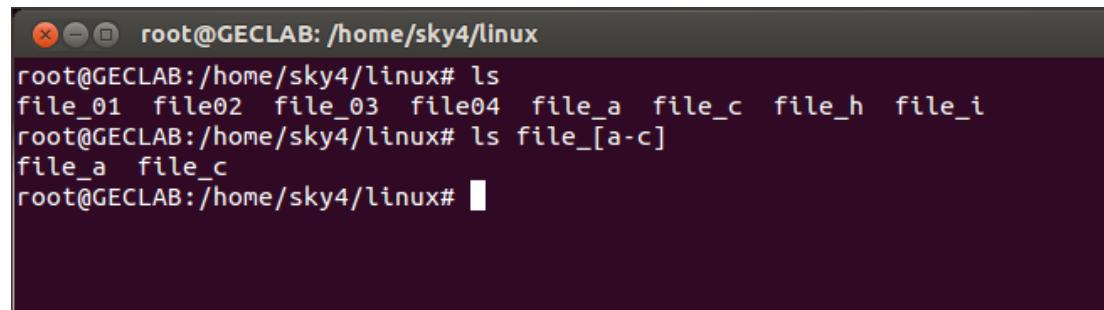
举例：显示通配符？ 符合制定规则的文件：



```
root@GECLAB:/home/sky4/linux
root@GECLAB:/home/sky4/linux# ls
file_01  file02  file_03  file04
root@GECLAB:/home/sky4/linux# ls file_0?
file_01  file_03
root@GECLAB:/home/sky4/linux#
```

[] 通配一个字符集或者范围

举例：显示通配符[] 符合制定规则的文件：

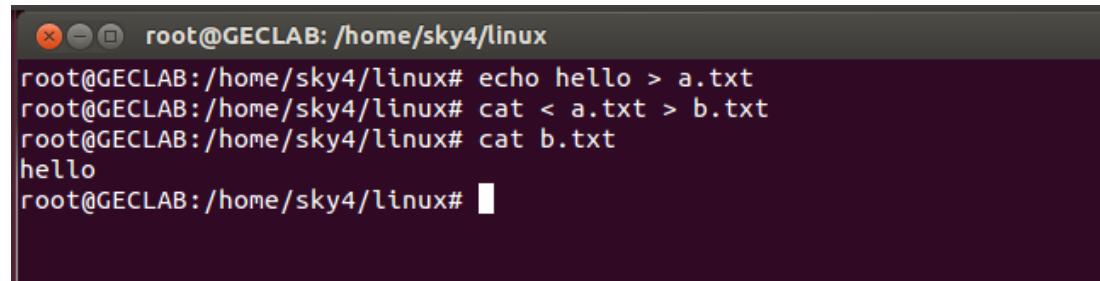


```
root@GECLAB:/home/sky4/linux
root@GECLAB:/home/sky4/linux# ls
file_01  file02  file_03  file04  file_a  file_c  file_h  file_i
root@GECLAB:/home/sky4/linux# ls file_[a-c]
file_a  file_c
root@GECLAB:/home/sky4/linux#
```

输入输出重定向

输入重定向

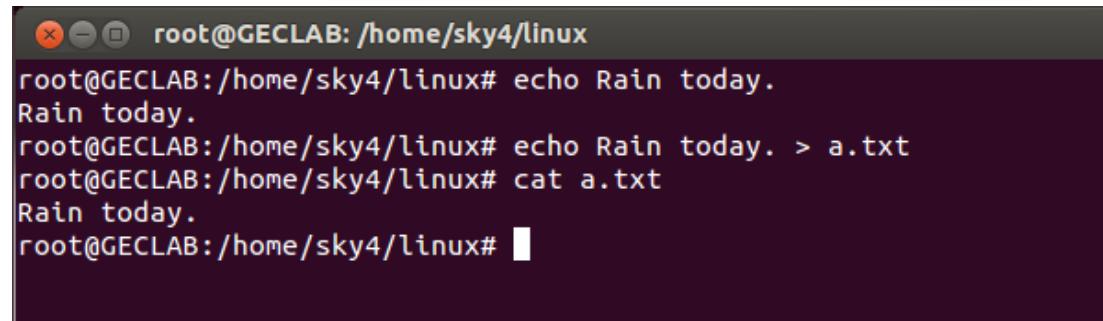
由键盘输入的改变到由文件输入



```
root@GECLAB:/home/sky4/linux
root@GECLAB:/home/sky4/linux# echo hello > a.txt
root@GECLAB:/home/sky4/linux# cat < a.txt > b.txt
root@GECLAB:/home/sky4/linux# cat b.txt
hello
root@GECLAB:/home/sky4/linux#
```

输出重定向

输出到屏幕上的改变为输出到文件



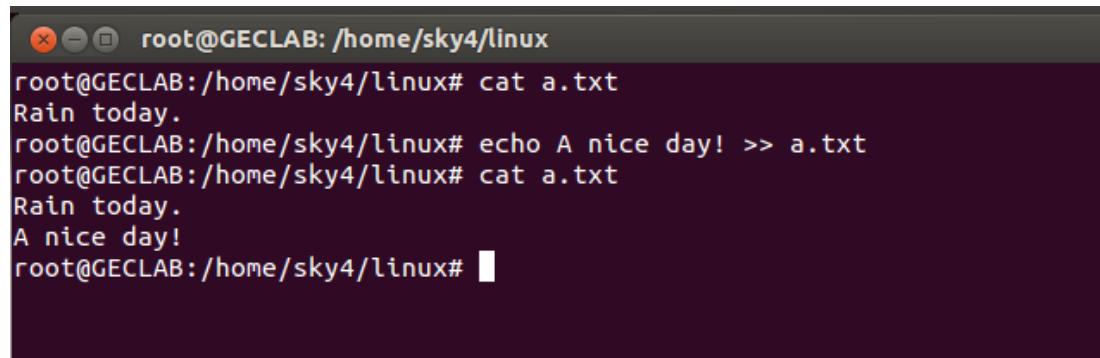
A screenshot of a terminal window titled "root@GECLAB: /home/sky4/linux". The terminal shows the following command sequence:

```
root@GECLAB:/home/sky4/linux# echo Rain today.  
Rain today.  
root@GECLAB:/home/sky4/linux# echo Rain today. > a.txt  
root@GECLAB:/home/sky4/linux# cat a.txt  
Rain today.  
root@GECLAB:/home/sky4/linux#
```

The terminal window has a dark background and light-colored text. It includes standard window controls (close, minimize, maximize) in the top-left corner.

追加

上面的输出重定向，我们可以发现：新的内容会完全覆盖原来的内容，但是追加就不会了。



The screenshot shows a terminal window with a dark background and white text. The title bar says "root@GECLAB: /home/sky4/linux". The terminal contains the following session:

```
root@GECLAB:/home/sky4/linux# cat a.txt
Rain today.
root@GECLAB:/home/sky4/linux# echo A nice day! >> a.txt
root@GECLAB:/home/sky4/linux# cat a.txt
Rain today.
A nice day!
root@GECLAB:/home/sky4/linux#
```

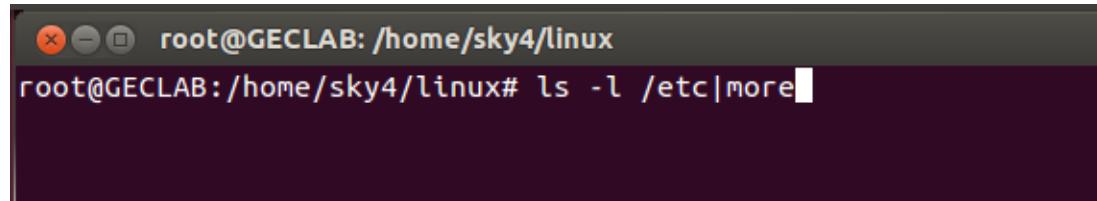
This demonstrates that the command `>>` appends new content to the end of the file, while `cat` shows both the original and appended content.

管道符

管道符 - |

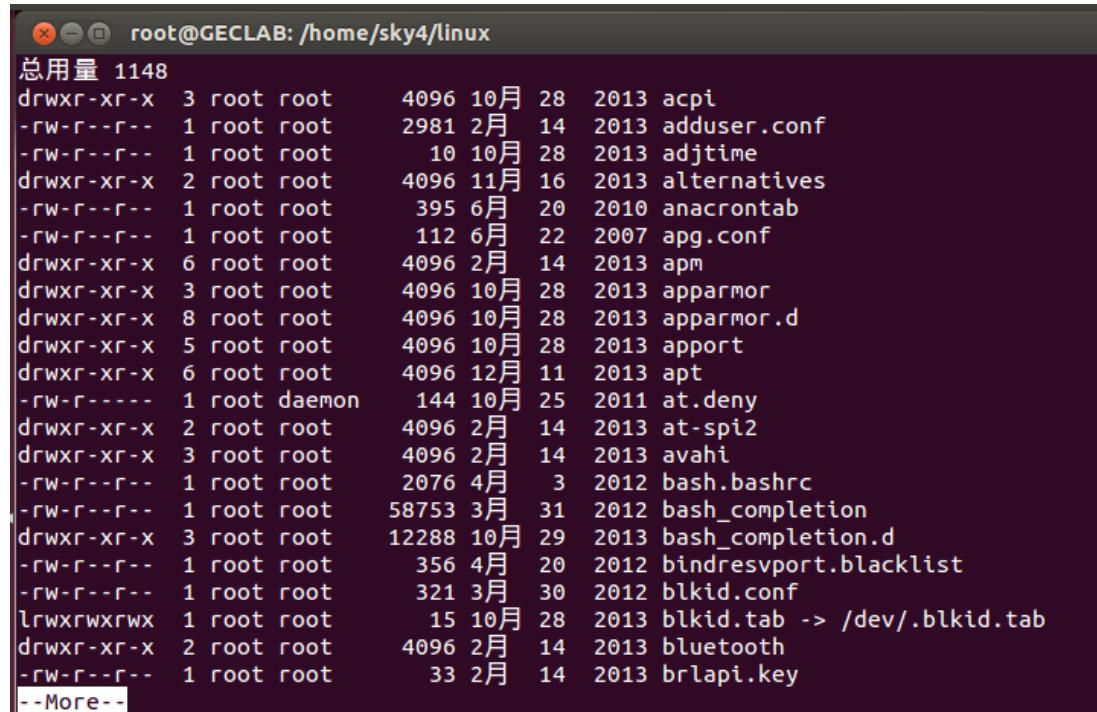
将一个命令的输出作为下一个命令的输入

举例： ls -l /etc|more



```
root@GECLAB: /home/sky4/linux
root@GECLAB: /home/sky4/linux# ls -l /etc|more
```

命令的运行结果：

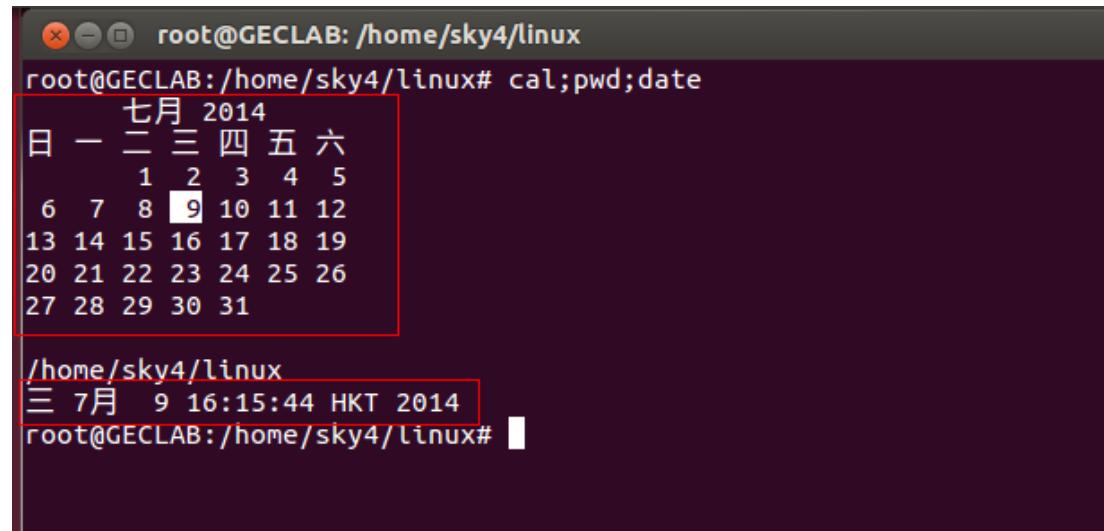


```
总用量 1148
drwxr-xr-x  3 root root    4096 10月 28 2013 acpi
-rw-r--r--  1 root root    2981 2月 14 2013 adduser.conf
-rw-r--r--  1 root root      10 10月 28 2013 adjtime
drwxr-xr-x  2 root root    4096 11月 16 2013 alternatives
-rw-r--r--  1 root root     395 6月 20 2010 anacrontab
-rw-r--r--  1 root root    112 6月 22 2007 apg.conf
drwxr-xr-x  6 root root    4096 2月 14 2013 aptm
drwxr-xr-x  3 root root    4096 10月 28 2013 apparmor
drwxr-xr-x  8 root root    4096 10月 28 2013 apparmor.d
drwxr-xr-x  5 root root    4096 10月 28 2013 apport
drwxr-xr-x  6 root root    4096 12月 11 2013 apt
-rw-r-----  1 root daemon   144 10月 25 2011 at.deny
drwxr-xr-x  2 root root    4096 2月 14 2013 at-spi2
drwxr-xr-x  3 root root    4096 2月 14 2013 avahi
-rw-r--r--  1 root root    2076 4月  3 2012 bash.bashrc
-rw-r--r--  1 root root   58753 3月 31 2012 bash_completion
drwxr-xr-x  3 root root   12288 10月 29 2013 bash_completion.d
-rw-r--r--  1 root root     356 4月 20 2012 bindresvport.blacklist
-rw-r--r--  1 root root     321 3月 30 2012 blkid.conf
lrwxrwxrwx  1 root root      15 10月 28 2013 blkid.tab -> /dev/.blkid.tab
drwxr-xr-x  2 root root    4096 2月 14 2013 bluetooth
-rw-r--r--  1 root root     33 2月 14 2013 brlapi.key
--More--
```

命令分隔符 (;)

作用：在同一命令中分隔多条命令，其实这个符号在前面的例子里面我们已经使用过很多次了。

例子： cal; pwd; date



The screenshot shows a terminal window with the following content:

```
root@GECLAB: /home/sky4/linux
root@GECLAB: /home/sky4/linux# cal;pwd;date
                七月 2014
日 一 二 三 四 五 六
      1  2  3  4  5
      6  7  8  9  10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

/home/sky4/linux
三 7月 9 16:15:44 HKT 2014
root@GECLAB:/home/sky4/linux#
```

The output of the command `cal` is highlighted with a red box, showing the calendar for July 2014. The date '9' is also highlighted with a red box. The output of `pwd` and the current date from `date` are also shown.

续行符 (\)

继续在下一行输入命令
举例: cat /etc/passwd \
| sort > ~/userinfo
命令的运行结果:

```
root@GECLAB: /home/sky4/linux
root@GECLAB:/home/sky4/linux# cat /etc/passwd \
> | sort > ~/userinfo
root@GECLAB:/home/sky4/linux# more ~/userinfo
```

现在显示的结果都是经过排序的。

```
root@GECLAB: /home/sky4/linux
avahi-autoipd:x:106:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:107:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
backup:x:34:34:backup:/var/backups:/bin/sh
bin:x:2:2:bin:/bin:/sh
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
games:x:5:60:games:/usr/games:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/bin/false
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
messagebus:x:102:105::/var/run/dbus:/bin/false
news:x:9:9:news:/var/spool/news:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
pulse:x:110:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
root:x:0:0:root:/root:/bin/bash
--More-- (67%)
```