

Linux Command基本应用

张 强



熟练掌握Linux Command是使用Linux系统进行开发所需要的基本技能，Linux命令交互方式完胜于视窗交互方式，但由于命令交互下每个命令的功能又过于强大，而且参数众多，往往导致很多人知道这些命令，但是使用时却无所适从。

培训目的:

通过整理开发中常用命令的使用用法，本文档简略介绍了这些命令的常用使用技能和注意事项。同时，也可将该文档作为Linux Command速查手册，便于在需要时使用。



mount

示例:

- `mount -t nfs -o intr,nolock,rsize=1024,wsiz=1024 10.3.4.100:/home/zhangqiang/ /mnt/nfs`
- `mount -t vfat /dev/sda1 /mnt/hd`
- `mount`

参数说明:

`mount -o` 后面的用来指定文件系统的参数，可以通过`man nfs`查看。

- `intr` linux2.6.25后已无实际意义。
- `nolock` 无文件锁。
- `rsiz` 一次 READ 操作最大能读取的数据字节数。
- `wsiz` 一次 WRITE操作最大能写入的数据字节数。

注意事项:

- Mount目录(示例中/mnt/nfs)必须为空目录，否则会导致文件系统混乱。



tar

示例:

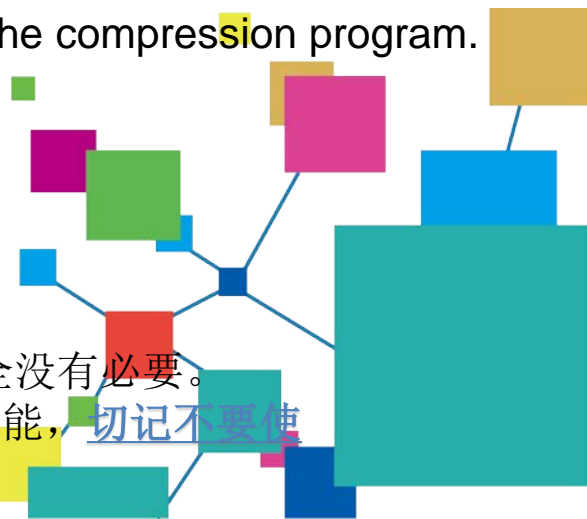
- `tar -zcf test.tar.gz test`
- `tar -zxvf test.tar.gz`

参数说明:

- `-j, --bzip2` filter the archive through bzip2
- `--lzma` filter the archive through lzma
- `--no-auto-compress` do not use archive suffix to determine the compression program.
- `-x, --extract, --get` extract files from an archive
- `-c, --create` create a new archive
- `-f, --file=ARCHIVE` use archive file or device ARCHIVE
- `-z, --gzip, --gunzip, --ungzip` filter the archive through gzip

注意事项:

- 发现很多同事在打包和解包的过程中喜欢使用 `-v` 参数，其实完全没有必要。
- 虽然 winrar 等 windows 使用的解压工具也可以完成打包和解包功能，切记不要使用。除非你能保证包中无链接文件等 linux 特有文件。



find

示例:

➤ `find . -name rcS*`

```
zhangqiang@chnetserver:~/target$ find ./ -name rcS*  
./etc/rc.d/rcS.d  
./etc/default/rcS  
./etc/init.d/rcSBB  
./etc/init.d/rcS  
./usr/share/man/man5/rcS.5.gz
```

参数说明:

无

注意事项:

- 注意通配符的使用。
- `find`命令一般用于查找指定目录下的文件或者目录。



grep

示例:

➤ `grep -r PB_DS* .`

```
zhangqiang@chtnetserver:~/target/usr/include/c++/4.2.4/ext/pb_ds/detail/basic_tree_policy$ grep -r PB_DS* ./
./basic_tree_policy_base.hpp:#ifndef PB_DS_TREE_LIKE_POLICY_BASE_HPP
./basic_tree_policy_base.hpp:#define PB_DS_TREE_LIKE_POLICY_BASE_HPP
./basic_tree_policy_base.hpp:#define PB_DS_CLASS_C_DEC
./basic_tree_policy_base.hpp:    return (const_cast<PB_DS_CLASS_C_DEC*>(this)->end());
./basic_tree_policy_base.hpp:#undef PB_DS_CLASS_C_DEC
./basic_tree_policy_base.hpp:#endif // #ifndef PB_DS_TREE_LIKE_POLICY_BASE_HPP
./traits.hpp:#ifndef PB_DS_NODE_AND_IT_TRAITS_HPP
./traits.hpp:#define PB_DS_NODE_AND_IT_TRAITS_HPP
./traits.hpp:#endif // #ifndef PB_DS_NODE_AND_IT_TRAITS_HPP
./null_node_metadata.hpp:#ifndef PB_DS_NULL_NODE_METADATA_HPP
./null_node_metadata.hpp:#define PB_DS_NULL_NODE_METADATA_HPP
```

参数说明:

无

注意事项:

- 注意通配符的使用。
- `grep`一般用于查找指定目录下文件中的内容，在文件量很大的情况下，效率很高，特别适用于查找工程中某些宏定义。



whereis which

示例:

- whereis ls
- which ls

```
zhangqiang@chnetserver:~$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
zhangqiang@chnetserver:~$ which ls
/bin/ls
```

参数说明:

无

注意事项:

- whereis 是查询命令所在的目录以及帮助文档所在目录。
- which 是查询命令所在的目录。



chmod

示例:

➤ `chmod -R 755 target`

参数说明:

无

关于权限

在unix系统中,文件的属性在ls中可以看到是用10个字符定义的。
如"drwxrwxrwx",现在我们来详细解释一下其中每一位的意思。

第一位是表示文件类型,我所知道的有:

d--文件夹

b--块类型

l--快捷方式

第二到第四位表示文件拥有者对文件的访问权限

第五到第七位表示和文件拥有者同组的用户对文件的访问权限

第八到第十位表示其他用户对文件的访问权限

以第二到第四位为例:

r表示有读(read)的权限

w表示有写(write)的权限

x表示有运行(execute)的权限。

请特别注意:有些根文件系统文件必须用超级权限才能操作。

注意事项:

- 很多同事喜欢用777修改目标的访问权限,其实这是很不好的做法。建议按照需求设定相应值,万不得已的情况下才设置777。
- 在开发过程中,很多时候会发现文件和目录不能访问,优先考虑的原因一定是权限问题。



chown

示例:

- `chown -R zhangqiang main`
- `chown -R zhangqiang.software main`

```
-rwxr-xr-x 1 zhangqiang software 35597346 2014-04-18 11:48 main
```

参数说明:

无

注意事项:

- `chown`一般用于修改文件属主,第二个示例同时修改属主和文件所属群组。
- `chown` 可以记为change owner。



chgrp

示例:

➤ chgrp -R software target

参数说明:

无

注意事项:

- chgrp用户修改目标文件所属群组。
- 可以记为 change group。



cd

示例:

- cd or cd ~
- cd -

参数说明:

- 无

注意事项:

- 示例1为进入用户主目录，cd -进入上次所在目录。





示例:

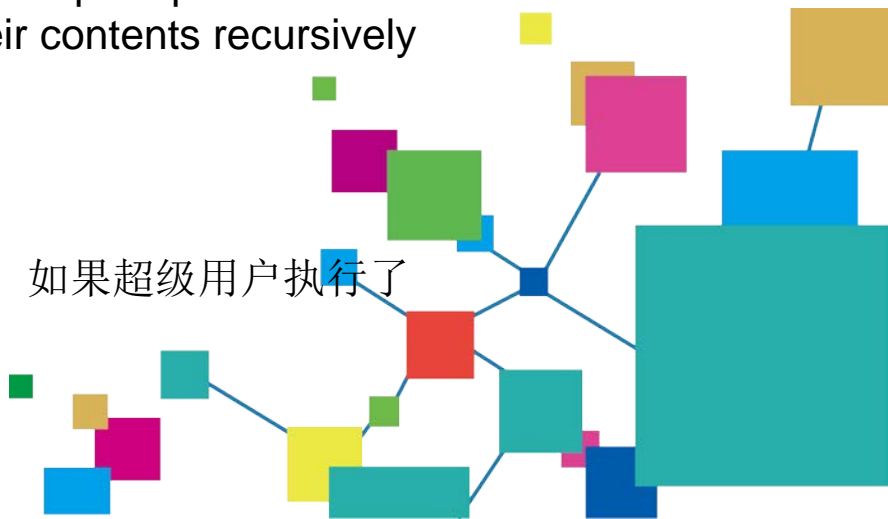
- `rm -rf target`

参数说明:

- `-f, --force` ignore nonexistent files, never prompt
- `-r, -R, --recursive` remove directories and their contents recursively

注意事项:

- `rm`使用时一定要小心，特别是超级用户使用时，如果超级用户执行了 `sudo rm -rf /`，可以想象后果是什么。



cp

示例:

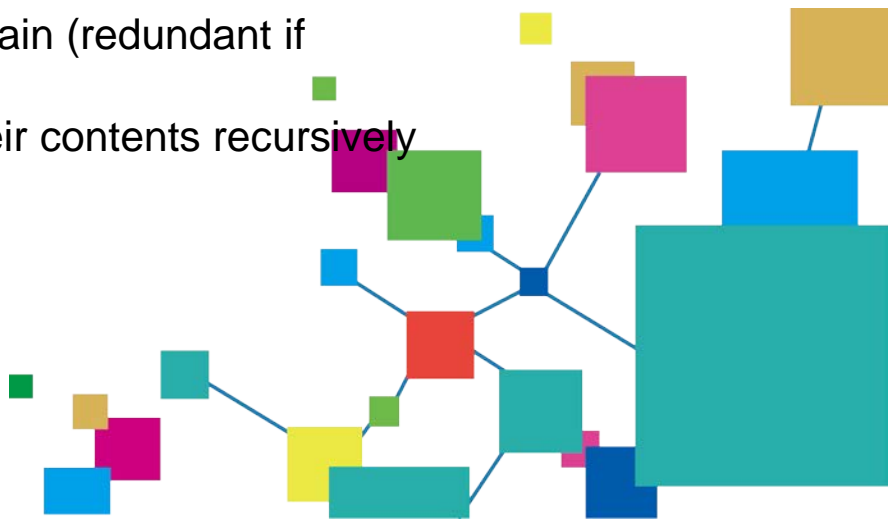
- `cp -rf target target_bak`

参数说明:

- `-f, --force` if an existing destination file cannot be opened, remove it and try again (redundant if the `-n` option is used)
- `-r, -R, --recursive` remove directories and their contents recursively

注意事项:

- 无



mv

示例:

➤ mv target target_bak

参数说明:

➤ 无。

注意事项:

➤ 移动或重命名目标，可记为move。



uname

示例:

➤ `uname -a`

```
zhangqiang@chnetserver:~$ uname -a  
Linux chnetserver 2.6.32-33-server #70-Ubuntu SMP Thu Jul 7 22:28:30 UTC 2011 x86_64 GNU/Linux
```

参数说明:

- `-a, --all` print all information, in the following order, except omit `-p` and `-i` if unknown.
- 查看linux版本信息

注意事项:

- 无



uptime

示例:

➤ uptime

```
zhangqiang@chnetserver:~$ uptime  
16:20:01 up 14 days, 22:36, 3 users, load average: 0.45, 0.26, 0.09
```

参数说明:

➤ 查看系统启动时间

注意事项:

➤ 无



ps

示例:

➤ ps -A

参数说明:

➤ -A all processes

```
zhangqiang@chnetserver:~$ sudo ps -A
  PID TTY          TIME CMD
    1 ?            00:00:03 init
    2 ?            00:00:00 kthreadd
    3 ?            00:00:00 migration/0
    4 ?            00:00:00 ksoftirqd/0
    5 ?            00:00:00 watchdog/0
    6 ?            00:00:00 migration/1
    7 ?            00:00:00 ksoftirqd/1
    8 ?            00:00:00 watchdog/1
    9 ?            00:00:00 migration/2
   10 ?            00:00:02 ksoftirqd/2
   11 ?            00:00:00 watchdog/2
```

进程的5种状态

D 不可中断 uninterruptible sleep (usually IO)

R 运行 runnable (on run queue)

S 中断 sleeping

T 停止 traced or stopped

Z 僵死 a defunct ("zombie") process

注意事项:

➤ 无



top

示例:

➤ top

参数说明:

➤ 无

注意事项:

➤ top是动态监控指令。

zhangqiang@chnetserver:~\$ top

```
top - 16:21:28 up 14 days, 22:37, 3 users, load average: 0.20, 0.22, 0.09
Tasks: 381 total, 1 running, 380 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.1%us, 0.0%sy, 0.0%ni, 99.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 16452244k total, 15356392k used, 1095852k free, 8419664k buffers
Swap: 34866200k total, 0k used, 34866200k free, 3154744k cached
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8320	zhangqia	20	0	19484	1520	948	R	2	0.0	0:00.02	top
1	root	20	0	23840	1972	1196	S	0	0.0	0:03.23	init
2	root	20	0	0	0	0	S	0	0.0	0:00.01	kthreadd
3	root	RT	0	0	0	0	S	0	0.0	0:00.01	migration/0
4	root	20	0	0	0	0	S	0	0.0	0:00.43	ksoftirqd/0
5	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/0
6	root	RT	0	0	0	0	S	0	0.0	0:00.01	migration/1
7	root	20	0	0	0	0	S	0	0.0	0:00.57	ksoftirqd/1
8	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/1
9	root	RT	0	0	0	0	S	0	0.0	0:00.04	migration/2
10	root	20	0	0	0	0	S	0	0.0	0:02.18	ksoftirqd/2
11	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/2
12	root	RT	0	0	0	0	S	0	0.0	0:00.04	migration/3
13	root	20	0	0	0	0	S	0	0.0	0:02.26	ksoftirqd/3
14	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/3
15	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/4
16	root	20	0	0	0	0	S	0	0.0	0:00.07	ksoftirqd/4
17	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/4
18	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/5
19	root	20	0	0	0	0	S	0	0.0	0:00.14	ksoftirqd/5
20	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/5
21	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/6

关于top信息

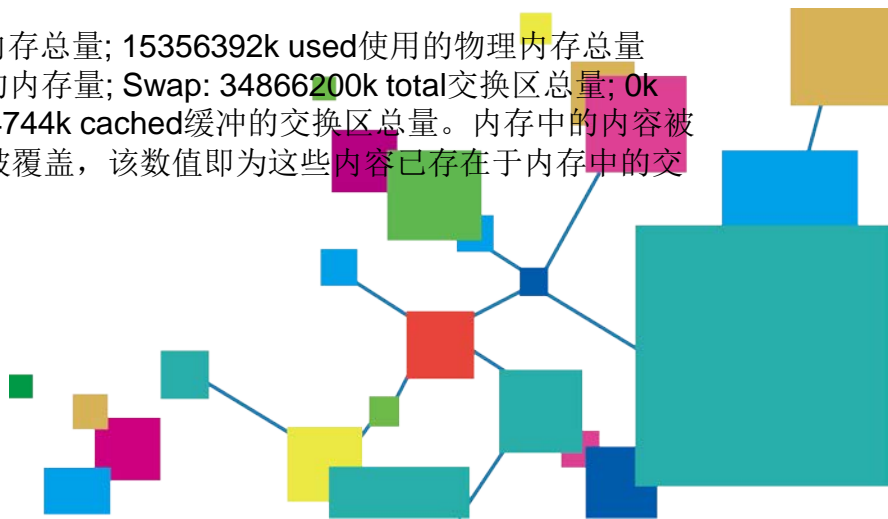
统计信息区前五行为系统整体的统计信息。

第一行是任务队列信息，同 **uptime** 命令的执行结果。其内容如下：**01:06:48** 当前时间；**up 1:22**系统运行时间，格式为时:分；**1 user** 当前登录用户数；**load average: 0.06, 0.60, 0.48** 系统负载，即任务队列的平均长度。三个数值分别为 1分钟、5分钟、15分钟前到现在的平均值。

第二、三行为进程和CPU的信息。当有多个CPU时，这些内容可能会超过两行。内容如下：**Tasks: 29 total** 进程总数；**1 running**正在运行的进程数；**28 sleeping** 睡眠的进程数；**0 stopped** 停止的进程数；**0 zombie**僵尸进程数；**Cpu(s): 0.3% us** 用户空间占用CPU百分比；**1.0% sy** 内核空间占用CPU百分比；**0.0% ni** 用户进程空间内改变过优先级的进程占用CPU百分比；**98.7% id** 空闲CPU百分比；**0.0% wa** 等待输入输出的CPU时间百分比；**0.0% hi**0.0% si

最后两行为内存信息。内容如下：**Mem: 16452244k total**物理内存总量；**15356392k used**使用的物理内存总量；**1095852k free**空闲内存总量；**8419664k buffers**用作内核缓存的内存量；**Swap: 34866200k total**交换区总量；**0k used**使用的交换区总量；**34866200k free** 空闲交换区总量；**3154744k cached**缓冲的交换区总量。内存中的内容被换出到交换区，而后再被换入到内存，但使用过的交换区尚未被覆盖，该数值即为这些内容已存在于内存中的交换区的大小。相应的内存再次被换出时可不必再对交换区写入。

进程信息区统计信息区域的下方显示了各个进程的详细信息。



free

示例:

➤ free

```
zhangqiang@chnetserver:~$ free
              total        used        free      shared    buffers     cached
Mem:          16452244    15350352     1101892           0      8419960     3149064
-/+ buffers/cache:    3781328    12670916
Swap:         34866200           0     34866200
zhangqiang@chnetserver:~$
```

参数说明:

➤ 无

注意事项:

➤ 如果根文件系统没有**free**命令, 可以使用**cat /proc/meminfo**, 有兴趣的可以参考**proc**文件系统。



df

示例:

➤ df -h

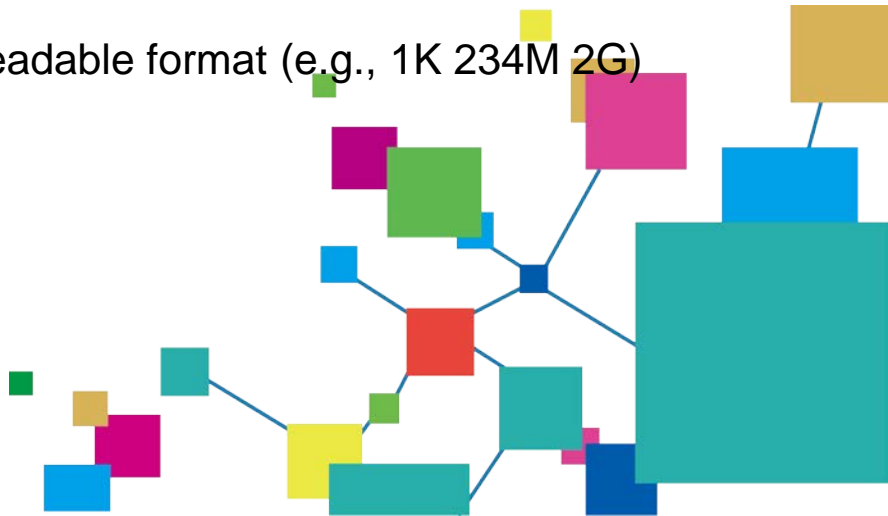
```
zhangqiang@chnetserver:~/target$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        56G   6.7G   46G   13% /
none             7.9G  212K   7.9G    1% /dev
none             7.9G    0   7.9G    0% /dev/shm
none             7.9G   8.3M   7.9G    1% /var/run
none             7.9G    0   7.9G    0% /var/lock
none             7.9G    0   7.9G    0% /lib/init/rw
/dev/sda2       143G   92G   44G   68% /opt
/dev/sda4       1.6T   1.1T  466G   70% /home
```

参数说明:

➤ -h, --human-readable print sizes in human readable format (e.g., 1K 234M 2G)

注意事项:

➤ 无



du

示例:

- `du -h --max-depth=1`

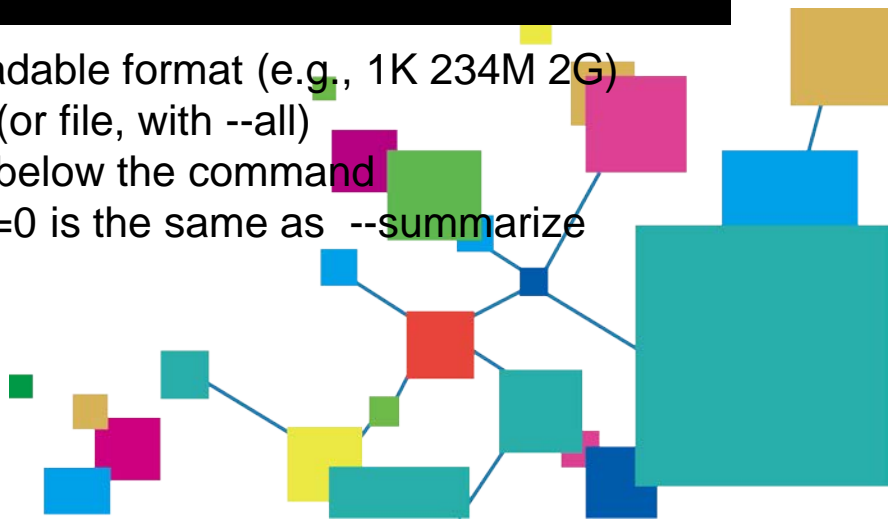
```
zhangqiang@chnetserver:~/target$ du -h --max-depth=1
2.6M    ./etc
128M    ./boot
4.0K    ./sys
9.2M    ./sbin
2.6G    ./root
4.0K    ./tmp
292K    ./app
4.0K    ./media
7.7M    ./lib
3.9M    ./var
4.0K    ./proc
4.0K    ./opt
5.5M    ./bin
12K     ./mnt
28K     ./home
4.0K    ./misc
68K     ./dev
931M    ./usr
1.3M    ./ch_flash_data
8.0K    ./include
3.6G    -
```

参数说明:

- `-h, --human-readable` print sizes in human readable format (e.g., 1K 234M 2G)
- `--max-depth=N` print the total for a directory (or file, with `--all`) only if it is N or fewer levels below the command line argument; `--max-depth=0` is the same as `--summarize`

注意事项:

- 无



fdisk

示例:

- `sudo fdisk -l`
- `sudo fdisk /dev/sdb`

参数说明:

- 无

注意事项:

- 使用fdisk必须用超级权限。
- 对磁盘进行分区时(示例2), 使用d删除分区, 使用n分区, 注意:选择p(primary partition)(1-4)主分区即sdb1~4),w写入分区信息。

```
zhangqiang@chnetserver:~/MStar_7C51G/porting_0404_test/project/mstar$ sudo fdisk -l

WARNING: GPT (GUID Partition Table) detected on '/dev/sda'! The util fdisk doesn't s

Disk /dev/sda: 2000.4 GB, 2000398934016 bytes
255 heads, 63 sectors/track, 243201 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x99ff9a00

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1                1      243202    1953514583+   ee    GPT

Disk /dev/sdb: 2000.4 GB, 2000398934016 bytes
255 heads, 63 sectors/track, 243201 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table
```

mkfs.***

示例:

- sudo mkfs.ext4 /dev/sdb1
- sudo mkfs.ntfs /dev/sdb1
- sudo mkfs.vfat /dev/sdb1

参数说明:

- 无

关于自动mount

编辑/etc/fstab文件插入相应内容:

```
fs_spec fs_file fs_type fs_options fs_dump fs_pass
/dev/hda1 / ext2 defaults 1 1
```

注意事项:

- 格式化磁盘。

```
zhangqiang@chnetserver:~/MStar_7C51G/porting_0404_test/project/mstar$ vi /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' 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>
proc /proc proc nodev,noexec,nosuid 0 0
# / was on /dev/sda1 during installation
UUID=b07af806-355c-4881-9f53-50b7f2070efc / ext4 errors=remount-ro 0 1
# /home was on /dev/sda4 during installation
#UUID=b07af806-355c-4881-9f53-50b7f2070efc /home ext4 defaults 0 2
#UUID=b07af806-355c-4881-9f53-50b7f2070efc /home ext4 defaults,usrquota,grpquota 0 2
# /opt was on /dev/sda2 during installation
UUID=3a23193a-2892-480e-9c68-d52d8eb6272a /opt ext4 defaults 0 2
# swap was on /dev/sda3 during installation
UUID=31d4cdec-f462-4ae3-9001-04a1fbf3918e none swap sw 0 0
# /home1 was on /dev/sdb for future use
#UUID=955066e8-ae5d-4b82-93ba-151f6cdc9d6f /home1 ext4 defaults 0 2
```


blkid

示例:

- `sudo blkid`

```
zhangqiang@chnetserver:~/MStar_7C51G/porting_0404_test/project/mstar$ sudo blkid
[sudo] password for zhangqiang:
/dev/sda1: UUID="b07af806-355c-4881-9f53-50b7f2070efc" TYPE="ext4"
/dev/sda2: UUID="3a23193a-2892-480e-9c68-d52d8eb6272a" TYPE="ext4"
/dev/sda3: UUID="31d4cdec-f462-4ae3-9001-04a1fbf3918e" TYPE="swap"
/dev/sda4: UUID="bf44255d-5c3c-427f-ab74-bcd03cb723d3" TYPE="ext4"
/dev/sdb: UUID="955066e8-ae5d-4b82-93ba-151f6cdc9d6f" TYPE="ext4"
```

参数说明:

- 无

注意事项:

- 查看卷标签。 可用于编辑/etc/fstab的fs_spec字段。使磁盘不受设备名的影响。





示例:

➤ ln -s main main.so.1

```
zhangqiang@chnetserver:~/MStar_7C51G/porting_0404_test/project/mstar$ ln -s main main.so.1
zhangqiang@chnetserver:~/MStar_7C51G/porting_0404_test/project/mstar$ ls -l main*
-rwxr-xr-x 1 zhangqiang software 35597587 2014-04-21 18:19 main
-rwxr-xr-x 1 zhangqiang software      1493 2014-04-04 17:30 main.c
-rw-r--r-- 1 zhangqiang software 6287228 2014-04-21 18:19 main.map
lrwxrwxrwx 1 zhangqiang software      4 2014-04-22 09:12 main.so.1 -> main
```

参数说明:

➤ -s, --symbolic make symbolic links instead of hard links

注意事项:

➤ 注意源文件和软链接文件的顺序。



ifconfig udhcpc

示例:

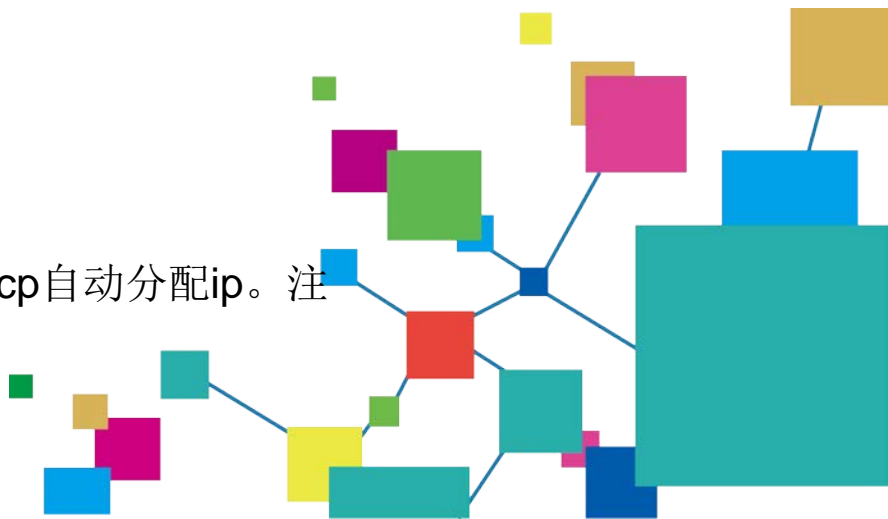
- ifconfig eth1 -addr=10.3.45.195
- ifconfig eth0 hw ether 00:14:49:00:00:F0
- Ifconfig -a
- udhpc -i eth1

参数说明:

- 无

注意事项:

- 示例分别为设置ip,mac,显示所有网口状态, dhcp自动分配ip。注意多网口盒子设备号的使用(eth0,eth1...).



WIFI STA Command

示例:

- ifconfig wlan0 up
- iwlist wlan0 scan

iwconfig 设置 WEP 密码:

- iwconfig wlan0 essid "leo"
- iwconfig wlan0 channel 6
- iwconfig wlan0 key s:12345

wpa_supplicant 设置 WEP, WAP, WAP2 密码

- ifconfig wlan0 down
- wpa_supplicant -Dwext -iwlan0 -c/root/wpa_supplicant.conf &
- wpa_cli -iwlan0 disable_network all
- wpa_cli -iwlan0 remove_network all
- ifconfig wlan0 down
- ifconfig wlan0 up



WIFI STA Command

示例:

WEP(若为NONE,则除掉wep_key0项的设置):

- wpa_cli -iwlan0 add_network(这个的值作为下面ID的代号)
- wpa_cli -iwlan0 set_network 0 ssid ""leo""
- wpa_cli -iwlan0 set_network 0 key_mgmt NONE
- wpa_cli -iwlan0 set_network 0 scan_ssid 1
- wpa_cli -iwlan0 set_network 0 wep_key0 ""12345""
- wpa_cli -iwlan0 enable_network 0
- wpa_cli -iwlan0 status

WPA, WPA2:

- wpa_cli -iwlan0 add_network(这个的值作为下面ID的代号)
- wpa_cli -iwlan0 set_network 1 ssid ""leo""
- wpa_cli -iwlan0 set_network 1 scan_ssid 1
- wpa_cli -iwlan0 set_network 1 proto 'WPA RSN'
- wpa_cli -iwlan0 set_network 1 key_mgmt WPA-PSK
- wpa_cli -iwlan0 set_network 1 pairwise 'CCMP TKIP'
- wpa_cli -iwlan0 set_network 1 group 'CCMP TKIP'
- wpa_cli -iwlan0 set_network 1 psk ""123456789a""
- wpa_cli -iwlan0 enable_network 1
- wpa_cli -iwlan0 status

- wgetid wlan0
- udhcpc -i wlan0



WIFI AP Command

示例:

- insmod *.ko
- ifconfig wlan0 up
- hostapd /root/hostapd.conf &

wlan0 AP做DHCP服务器:

- ifconfig wlan0 192.168.0.1
- udhcpd /etc/udhcpd.conf

NetBridge Open:

//建立网桥

- brctl addbr br0

//添加端口到网桥

- brctl addif br0 eth0
- brctl addif br0 wlan0

//打开混杂模式

- ifconfig eth0 0.0.0.0 promisc
- ifconfig wlan0 0.0.0.0 promisc

//启动网桥

- ifconfig br0 up

NetBridge Close:

//关闭网桥

- ifconfig br0 down

//去掉端口

- brctl delif br0 eth0
- brctl delif br0 wlan0

//删除网桥

- brctl delbr br0



谢谢

