

## 目录

目录	1
Linux常用系统参数设置	3
syslog命令	3
systemctl命令	3
crontab命令	4
mount命令	4
fsck命令	5
ulimit命令	6
sysctl命令	7
配置sysctl	8
fstab	9
禁止内核在空闲时频繁打印日志	10
关闭networkmanager服务	10
小技巧	11
自动挂载	11
交换分区UUID	11
swapon -s 确定交换分区	11
swapoff /dev/sda7 禁用交换分区	11
mkswap -U random /dev/sda7 用新 UUID 重新创建交换分区	11
swapon /dev/sda7 激活交换分区	11
路径名有空格	11
外部设备	11
普通用户读写	12
Windows常用系统参数设置	12
Windows系统如何挂载磁盘分区	12
CentOS yum源的配置	14
将金山云yum源切换成系统自带源	14
将系统自带源切换成金山云yum源	14
更改linux yum源方法:	15
Linux取消fstab开机加载	15
Linux取消fstab自动加载	15
Windows防火墙设置	16
允许响应ping (允许ICMP协议)	16
放开DHCP请求端口 (允许67、68端口)	16
放开RDP远程桌面连接端口 (允许3389端口)	16
放开IIS服务器被访问端口 (允许80端口)	16
放开HTTPS服务端口 (允许443端口)	16
445端口为运营商封禁端口, 不可用	16
windows防火墙	17
cmd命令配置	17
实例	17
Linux防火墙配置	17
Linux防火墙配置	17
1. 清除原有规则	18
2. 设定预设规则	18
3. 添加规则	18
著名的端口	19
UNIX 特有的端口	22
注册的端口	23
未注册端口	26



# Linux常用系统参数设置

## syslog命令

syslog是Linux系统默认的日志守护进程。syslog 接受来自系统的各种功能的信息，每个信息都包括重要级。

/etc/syslog.conf 文件通知 syslogd 如何根据设备和信息重要级别来报告信息。可以使用logger命令通过syslogd记录日志。

要向syslog文件/var/log/messages中记录日志信息：

```
logger this is a test log line
输出:
tail -n 1 messagesJan 5 10:07:03 localhost root: this is a test log line
```

记录特定的标记（tag）可以使用：

```
logger -t TAG this is a test log line
输出:
tail -n 1 messagesJan 5 10:37:14 localhost TAG: this is a test log line
```

## systemctl命令

systemctl命令是系统服务管理器指令，它实际上将 service 和 chkconfig 这两个命令组合到一起。

任务	旧指令	新指令
使某服务自动启动	chkconfig --level 3 httpd on	systemctl enable httpd.service
使某服务不自动启动	chkconfig --level 3 httpd off	systemctl disable httpd.service
检查服务状态	service httpd status	systemctl status httpd.service （服务详细信息） systemctl is-active httpd.service （仅显示是否 Active）
显示所有已启动的服务	chkconfig --list	systemctl list-units --type=service
启动某服务	service httpd start	systemctl start httpd.service
停止某服务	service httpd stop	systemctl stop httpd.service
重启某服务	service httpd restart	systemctl restart httpd.service

## 实例

### 1. 启动nfs服务

```
systemctl start nfs-server.service
```

### 2. 设置开机自启动

```
systemctl enable nfs-server.service
```

### 3. 停止开机自启动

```
systemctl disable nfs-server.service
```

### 4. 查看服务当前状态

```
systemctl status nfs-server.service
```

### 5. 重新启动某服务

```
systemctl restart nfs-server.service
```

### 6. 查看所有已启动的服务

```
systemctl list -units --type=service
```

### 开启防火墙22端口

```
iptables -I INPUT -p tcp --dport 22 -j accept
```

关闭SELinux:

修改/etc/selinux/config文件中的SELINUX=" " 为disabled, 然后重启。

彻底关闭防火墙:

```
sudo systemctl status firewalld.service  
sudo systemctl stop firewalld.service  
sudo systemctl disable firewalld.service
```

## crontab命令

**crontab**命令被用来提交和管理用户的需要周期性执行的任务, 当安装完成操作系统后, 默认会安装此服务工具, 并且会自动启动crond进程, crond进程每分钟会定期检查是否有要执行的任务, 如果有要执行的任务, 则自动执行。

### 语法

crontab(选项)(参数)

### 选项

```
-e: 编辑该用户的计时器设置;  
-l: 列出该用户的计时器设置;  
-r: 删除该用户的计时器设置;  
-u<用户名称>: 指定要设定计时器的用户名称。
```

### 参数

```
crontab文件: 指定包含待执行任务的crontab文件。
```

## mount命令

**mount**命令用于加载文件系统到指定的加载点。此命令的最常用于挂载cdrom, 使我们可以访问cdrom中的数据, 因为你将光

盘插入cdrom中，Linux并不会自动挂载，必须使用Linux mount命令来手动完成挂载。

## 语法

mount (选项) (参数)

## 选项

- V: 显示程序版本;
- l: 显示已加载的文件系统列表;
- h: 显示帮助信息并退出;
- v: 冗长模式，输出指令执行的详细信息;
- n: 加载没有写入文件“/etc/mtab”中的文件系统;
- r: 将文件系统加载为只读模式;
- a: 加载文件“/etc/fstab”中描述的所有文件系统。

## 参数

设备文件名: 指定要加载的文件系统对应的设备名

加载点: 指定加载点目录。

## 实例

```
mount -t auto /dev/cdrom /mnt/cdrom
mount: mount point /mnt/cdrom does not exist      /mnt/cdrom目录不存在，需要先创建。
cd /mnt
-bash: cd: /mnt: No such file or directory
mkdir -p /mnt/cdrom      创建/mnt/cdrom目录
ls
bin  dev  home  lib          media  mnt  proc  sbin  srv  tmp
var
boot  etc  initrd  lost+found  misc  opt  root  selinux  sys  usr
mount -t auto /dev/cdrom /mnt/cdrom      挂载cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only      挂载成功
ll /mnt/cdrom      查看cdrom里面内容
total 859
dr-xr-xr-x  4 root root   2048 Sep  4  2005 CentOS
-r--r--r--  2 root root   8859 Mar 19  2005 centosdocs-man.css
-r--r--r--  9 root root  18009 Mar  1  2005 GPL
dr-xr-xr-x  2 root root 241664 May  7 02:32 headers
dr-xr-xr-x  4 root root   2048 May  7 02:23 images
dr-xr-xr-x  2 root root   4096 May  7 02:23 isolinux
dr-xr-xr-x  2 root root  18432 May  2 18:50 NOTES
-r--r--r--  2 root root   5443 May  7 01:49 RELEASE-NOTES-en.html
dr-xr-xr-x  2 root root   2048 May  7 02:34 repodata
-r--r--r--  9 root root   1795 Mar  1  2005 rpm-GPG-KEY
-r--r--r--  2 root root   1795 Mar  1  2005 RPM-GPG-KEY-centos4
-r--r--r--  1 root root 571730 May  7 01:39 yumgroups.xml
```

## f sck 命令

f sck命令被用于检查并且试图修复文件系统中的错误。当文件系统发生错误时，可用f sck指令尝试加以修复。

## 语法

f sck(选项) (参数)

## 选项

```
-a: 自动修复文件系统, 不询问任何问题;  
-A: 依照/etc/fstab配置文件的内容, 检查文件内所列的全部文件系统;  
-N: 不执行指令, 仅列出实际执行会进行的动作;  
-P: 当搭配"-A"参数使用时, 则会同时检查所有的文件系统;  
-r: 采用互动模式, 在执行修复时询问问题, 让用户得以确认并决定处理方式;  
-R: 当搭配"-A"参数使用时, 则会略过/目录的文件系统不予检查;  
-s: 依序执行检查作业, 而非同时执行;  
-t<文件系统类型>: 指定要检查的文件系统类型;  
-T: 执行fsck指令时, 不显示标题信息;  
-V: 显示指令执行过程。
```

## 参数

文件系统: 指定要查看信息的文件系统。

## 实例

linux的文件系统损坏会导致linux不正常关机, 出错的时候如果系统告诉你是哪一块硬盘的分区有问题, 比如是/dev/hda2, 接着用如下的命令去处理它:

```
fsck -y /dev/hda2
```

结束后使用reboot命令重启系统如果不知道时哪个地方出了问题, 可以直接:

```
fsck
```

在随后的多个确认对话框中输入:y

结束后同样使用reboot命令重启系统

## ulimit命令

ulimit命令用来限制系统用户对shell资源的访问, 它是一种简单并且有效的实现资源限制的方式。

作为临时限制, ulimit 可作用于通过使用命令登录的 shell 会话, 在会话终止时便结束限制。对于长期的固定限制, ulimit 命令语句可被添加到由登录 shell 读取的文件中, 作用于特定的 shell 用户。

## 语法

ulimit (选项)

## 选项

```
-a: 显示目前资源限制的设定;  
-c <core文件上限>: 设定core文件的最大值, 单位为区块;  
-d <数据节区大小>: 程序数据节区的最大值, 单位为KB;  
-f <文件大小>: shell所能建立的最大文件, 单位为区块;  
-H: 设定资源的硬性限制, 也就是管理员所设下的限制;  
-m <内存大小>: 指定可使用内存的上限, 单位为KB;  
-n <文件数目>: 指定同一时间最多可开启的文件数;  
-p <缓冲区大小>: 指定管道缓冲区的大小, 单位512字节;  
-s <堆叠大小>: 指定堆叠的上限, 单位为KB;  
-S: 设定资源的弹性限制;  
-t <CPU时间>: 指定CPU使用时间的上限, 单位为秒;  
-u <程序数目>: 用户最多可开启的程序数目;  
-v <虚拟内存大小>: 指定可使用的虚拟内存上限, 单位为KB。
```

## 实例

```
[root@localhost ~]# ulimit -a
core file size          (blocks, -c) 0           #core文件的最大值为100 blocks。
data seg size          (kbytes, -d) unlimited  #进程的数据段可以任意大。
scheduling priority    (-e) 0
file size              (blocks, -f) unlimited  #文件可以任意大。
pending signals        (-i) 98304             #最多有98304个待处理的信号。
max locked memory      (kbytes, -l) 32         #一个任务锁住的物理内存的最大值为32KB。
max memory size        (kbytes, -m) unlimited  #一个任务的常驻物理内存的最大值。
open files             (-n) 1024            #一个任务最多可以同时打开1024的文件。
pipe size              (512 bytes, -p) 8       #管道的最大空间为4096字节。
POSIX message queues   (bytes, -q) 819200     #POSIX的消息队列的最大值为819200字节。
real-time priority     (-r) 0
stack size             (kbytes, -s) 10240     #进程的栈的最大值为10240字节。
cpu time               (seconds, -t) unlimited #进程使用的CPU时间。
max user processes     (-u) 98304            #当前用户同时打开的进程（包括线程）的最大个数为98304。
virtual memory         (kbytes, -v) unlimited  #没有限制进程的最大地址空间。
file locks             (-x) unlimited  #所能锁住的文件的最大个数没有限制。
```

## sysctl命令

**sysctl命令**被用于在内核运行时动态地修改内核的运行参数，可用的内核参数在目录/proc/sys中。它包含一些TCP/ip堆栈和虚拟内存系统的高级选项，用sysctl可以读取设置超过五百个系统变量。

### 语法

sysctl (选项) (参数)

### 选项

- n: 打印值时不打印关键字;
- e: 忽略未知关键字错误;
- N: 仅打印名称;
- w: 当改变sysctl设置时使用此项;
- p: 从配置文件“/etc/sysctl.conf”加载内核参数设置;
- a: 打印当前所有可用的内核参数变量和值;
- A: 以表格方式打印当前所有可用的内核参数变量和值。

### 参数

变量=值: 设置内核参数对应的变量值

### 实例

```
sysctl -a 读一个指定的变量，例如kern.maxproc:
sysctl kern.maxproc kern.maxproc: 1044
要设置一个指定的变量，直接用variable=value这样的语法:
sysctl kern.maxfiles=5000kern.maxfiles: 2088 -> 5000
```

可用sysctl修改系统变量，也可通过编辑sysctl.conf文件来修改系统变量。它用variable=value的形式来设定值。sysctl变量的设置通常是字符串、数字或者布尔型。（布尔型用 1 来表示‘yes’，用 0 来表示‘no’）。

```
sysctl -w kernel.sysrq=0
sysctl -w kernel.core_uses_pid=1
sysctl -w net.ipv4.conf.default.accept_redirects=0
sysctl -w net.ipv4.conf.default.accept_source_route=0
sysctl -w net.ipv4.conf.default.rp_filter=1
sysctl -w net.ipv4.tcp_syncookies=1
sysctl -w net.ipv4.tcp_max_syn_backlog=2048
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_synack_retries=2
sysctl -w net.ipv4.tcp_keepalive_time=3600
sysctl -w net.ipv4.tcp_window_scaling=1
sysctl -w net.ipv4.tcp_sack=1
```

## 配置sysctl

编辑此文件: /etc/sysctl.conf如果该文件为空, 则输入以下内容, 否则请根据情况自己做调整:

```
# Controls source route verification
# Default should work for all interfaces
net.ipv4.conf.default.rp_filter = 1
# net.ipv4.conf.all.rp_filter = 1
# net.ipv4.conf.lo.rp_filter = 1
# net.ipv4.conf.eth0.rp_filter = 1
# Disables IP source routing
# Default should work for all interfaces
net.ipv4.conf.default.accept_source_route = 0
# net.ipv4.conf.all.accept_source_route = 0
# net.ipv4.conf.lo.accept_source_route = 0
# net.ipv4.conf.eth0.accept_source_route = 0
# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0
# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1
# Increase maximum amount of memory allocated to shm
# Only uncomment if needed!
# kernel.shmmax = 67108864
# Disable ICMP Redirect Acceptance
# Default should work for all interfaces
net.ipv4.conf.default.accept_redirects = 0
# net.ipv4.conf.all.accept_redirects = 0
# net.ipv4.conf.lo.accept_redirects = 0
# net.ipv4.conf.eth0.accept_redirects = 0
# enable Log Spoofed Packets, Source Routed Packets, Redirect Packets
# Default should work for all interfaces
net.ipv4.conf.default.log_martians = 1
# net.ipv4.conf.all.log_martians = 1
# net.ipv4.conf.lo.log_martians = 1
# net.ipv4.conf.eth0.log_martians = 1
# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 25
# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1200
# Turn on the tcp_window_scaling
net.ipv4.tcp_window_scaling = 1
# Turn on the tcp_sack
net.ipv4.tcp_sack = 1
# tcp_fack should be on because of sack
net.ipv4.tcp_fack = 1
# Turn on the tcp_timestamps
net.ipv4.tcp_timestamps = 1
```



```
net.ipv4.tcp_timestamps = 1
# Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1
# Enable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1
# Enable bad error message Protection
net.ipv4.icmp_ignore_bogus_error_responses = 1
# make more local ports available
# net.ipv4.ip_local_port_range = 1024 65000
# set TCP Re-Ordering value in kernel to '5'
net.ipv4.tcp_reordering = 5
# Lower syn retry rates
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syn_retries = 3
# Set Max SYN Backlog to '2048'
net.ipv4.tcp_max_syn_backlog = 2048
# Various Settings
net.core.netdev_max_backlog = 1024
# Increase the maximum number of skb-heads to be cached
net.core.hot_list_length = 256
# Increase the tcp-time-wait buckets pool size
net.ipv4.tcp_max_tw_buckets = 360000
# This will increase the amount of memory available for socket input/output queues
net.core.rmem_default = 65535
net.core.rmem_max = 8388608
net.ipv4.tcp_rmem = 4096 87380 8388608
net.core.wmem_default = 65535
net.core.wmem_max = 8388608
net.ipv4.tcp_wmem = 4096 65535 8388608
net.ipv4.tcp_mem = 8388608 8388608 8388608
net.core.optmem_max = 40960
```

## fstab

这个文件描述系统中各种文件系统的信息。文件/etc/fstab包含了静态文件系统信息，定义了存储设备和分区整合到整个系统的方式。mount 命令会读取这个文件，确定设备和分区的挂载选项。

### 文件格式

```
<file system> <dir> <type> <options> <dump> <pass>
```

通过空格或 Tab 分隔。

<file systems> - 要挂载的分区或存储设备。

<dir> - <file systems>的挂载位置。

<type> - 要挂载设备或是分区的文件系统类型，支持许多种不同的文件系统

<options> - 挂载时使用的参数，注意有些mount 参数是特定文件系统才有的。一些比较常用的参数有：

```
auto - 在启动时或键入了 mount -a 命令时自动挂载。
noauto - 只在你的命令下被挂载。
exec - 允许执行此分区的二进制文件。
noexec - 不允许执行此文件系统上的二进制文件。
ro - 以只读模式挂载文件系统。
rw - 以读写模式挂载文件系统。
user - 允许任意用户挂载此文件系统，若无显示定义，隐含启用 noexec, nosuid, nodev 参数。
users - 允许所有 users 组中的用户挂载文件系统。
nouser - 只能被 root 挂载。
owner - 允许设备所有者挂载。
sync - I/O 同步进行。
async - I/O 异步进行。
dev - 解析文件系统上的块特殊设备。
nodev - 不解析文件系统上的块特殊设备。
suid - 允许 suid 操作和设定 sgid 位。
nosuid - 禁止 suid 操作和设定 sgid 位。
noatime - 不更新文件系统上 inode 访问记录，可以提升性能(参见 atime 参数)。
nodiratime - 不更新文件系统上的目录 inode 访问记录，可以提升性能(参见 atime 参数)。
relatime - 实时更新 inode access 记录。只有在记录中的访问时间早于当前访问才会被更新。
flush - vfat 的选项，更频繁的刷新数据，复制对话框或进度条在全部数据都写入后才消失。
defaults - 使用文件系统的默认挂载参数，例如 ext4 的默认参数为:rw, suid, dev, exec, auto, nouser, async.
```

<dump> -dump 工具通过它决定何时作备份。dump 会检查其内容，并用数字来决定是否对这个文件系统进行备份。允许的数字是 0 和 1。0 表示忽略，1 则进行备份。

<pass> -fsck 读取 <pass> 的数值来决定需要检查的文件系统的检查顺序。允许的数字是0, 1, 和2。根目录应当获得最高的优先权 1，其它所有需要被检查的设备设置为 2。0 表示设备不会被 fsck 所检查。

## 实例

使用内核名称标识磁盘：

```
/etc/fstab
# <file system> <dir> <type> <options> <dump> <pass>
tmpfs /tmp tmpfs nodev,nosuid 0 0
/dev/sda1 / ext4 defaults,noatime 0 1
/dev/sda2 none swap defaults 0 0
/dev/sda3 /home ext4 defaults,noatime 0 2
```

## 禁止内核在空闲时频繁打印日志

```
echo 1>/sys/module/rcupdate/parameters/rcu_cpu_stall_suppress
```

这个参数在rc.local文件中，解决云主机空闲时频繁刷日志导致业务卡顿的问题。不加这个参数，隔几秒会打印内核日志。

## 关闭networkmanager服务

centos6.6系统安装x-window后networkmanager服务自动启动会导致云主机重启后无法获取IP地址和DNS地址，可以通过关闭该服务解决此问题。

通过控制台VNC连接主机：

临时关闭：

```
service NetworkManager stop
```

永久关闭:

```
chkconfig NetworkManager off
```

## 小技巧

### 自动挂载

如果 /home 分区较大，可以让不依赖 /home 分区的服务先启动，把下面的参数添加到 /etc/fstab 文件中 /home 项目的参数部分。

```
noauto,x-systemd.automount
```

这样 /home 分区只有需要访问时才会被挂载。内核会缓存所有的文件操作，直到 /home 分区准备完成。这样会使 /home 的文件系统类型被识别为 autofs，mlocate 查询时忽略该目录。

挂载远程文件系统也是同理。另外，可以设置 `x-systemd.device-timeout=#参数`，设置超时时间，以防止网络资源不能访问的时候浪费时间。如果你的加密文件系统需要密钥，则需要添加 `noauto` 参数到 /etc/crypttab 文件中的对应位置。

```
/etc/crypttabdata /dev/md0 /root/key noauto
```

### 交换分区UUID

如果交换分区没有 UUID，可手动加入。

如果使用 `lsblk -f` 命令没有列出交换分区的 UUID 就说明发生了这种情况。为交换分区指定 UUID 的步骤：

## swapon -s 确定交换分区

## swapoff /dev/sda7 禁用交换分区

## mkswap -U random /dev/sda7 用新 UUID 重新创建交换分区

## swapon /dev/sda7 激活交换分区

### 路径名有空格

如果挂载的路径中有空格，可以使用“040”转义字符来表示空格（以三位八进制数来进行表示）

```
/etc/fstab
UUID=47FA-4071 /home/username/Camera040Pictures vfat defaults,noatime 0 2
/dev/sda7 /media/100040GB040(Storage) ext4 defaults,noatime,user 0 0
```

### 外部设备

外部设备在插入时挂载，在未插入时忽略。这需要 `nofail`，启动时设备不存在直接忽略它不报错。

```
/etc/fstab /dev/sdg1 /media/backup jfs defaults,nofail 0 2
```

## 普通用户读写

FAT32必须修改/etc/fstab文件才能取得对 FAT32 分区的写权限。

```
/etc/fstab/dev/sdxY /mnt/some_folder vfat user,rw,umask=000 0 0
```

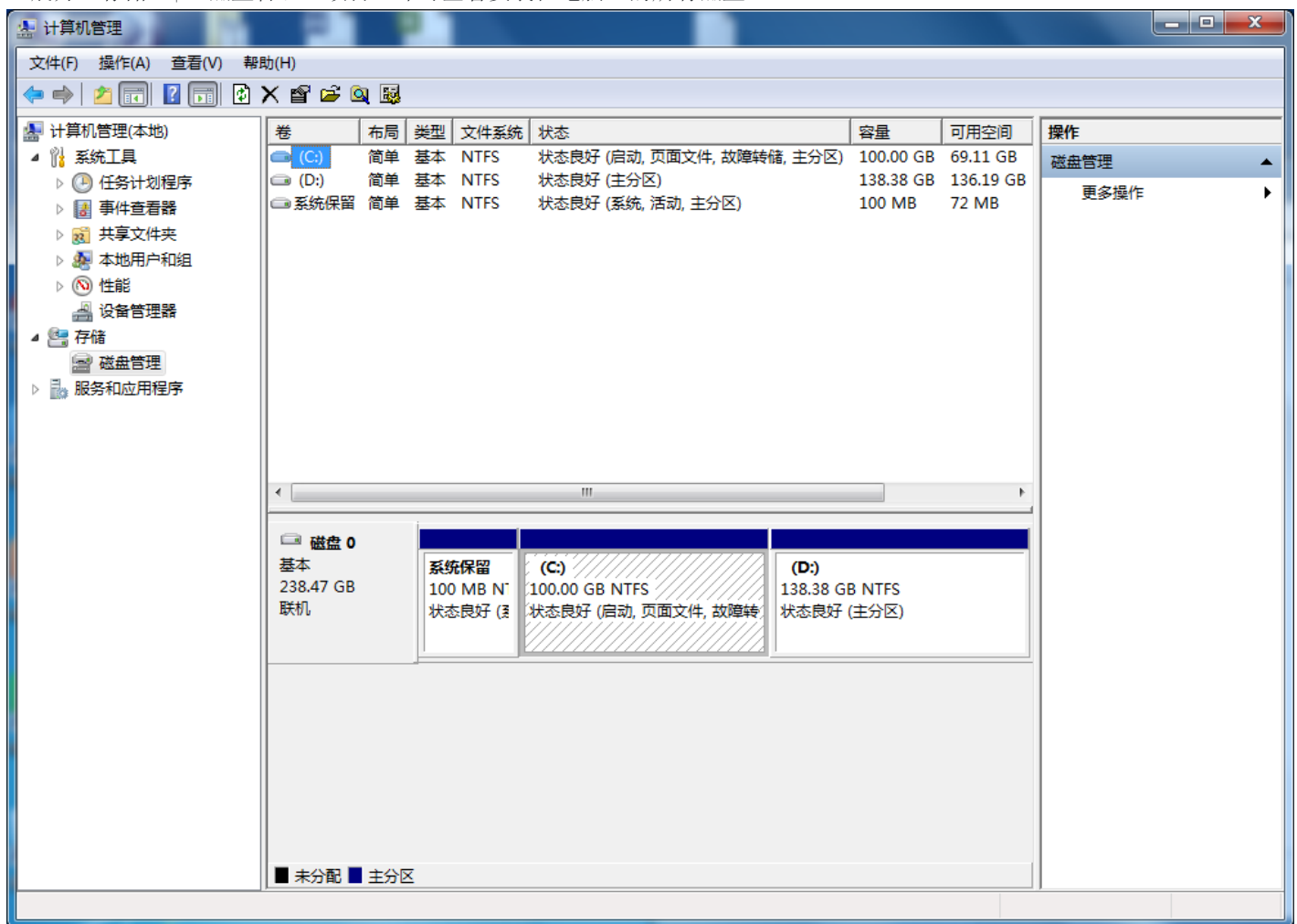
比如你的 FAT32 分区在 '/dev/sda9'，你想将其挂载到 '/mnt/fat32'，那么你需要输入并运行

```
/etc/fstab/dev/sda9 /mnt/fat32 vfat user,rw,umask=111,dmask=000 0 0
```

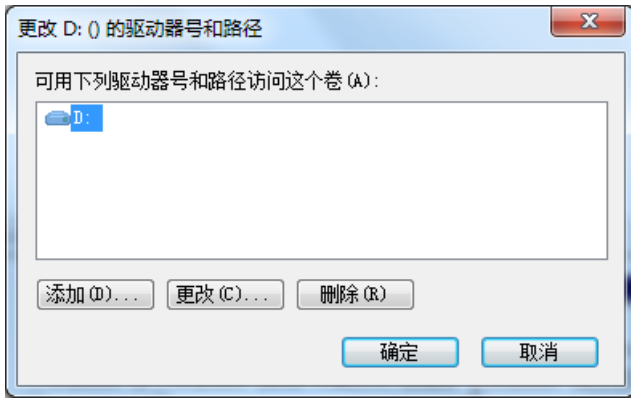
# Windows常用系统参数设置

## Windows系统如何挂载磁盘分区

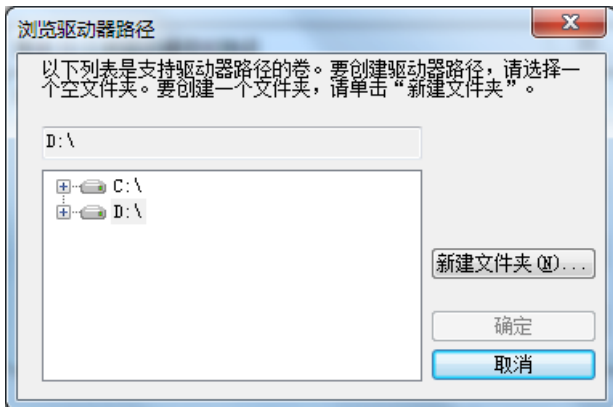
1. 单击开始按钮 按钮，在“计算机”选项上单击右键，选择“管理”选项，打开“计算机管理”窗口。
2. 展开“存储” | “磁盘管理”项目，即可查看安装在电脑上的所有磁盘



3. 在需要挂载的磁盘分区上单击右键，选择“更改驱动器号和路径”选项，然后在弹出的对话框中单击“添加”按钮。



4. 在弹出的对话框中选择要挂载的文件夹，该文件夹必须位于采用NTFS文件系统的硬盘分区中，而且文件夹必须是空的。



5. 点击确定查看

或者通过cmd命令

开始-运行-cmd

进入命令行

输入diskpart,回车。

输入list disk,可以看到硬盘编号,0,1,2,3...

输入select disk x(x是硬盘编号)选中要分区的硬盘。

输入clean清除该硬盘上的所有分区(如果硬盘不是未分区状态)。

输入convert gpt将该硬盘转换成GPT分区表(基本上可以说是永久转换,没法转回MBR,除非再次用clean清除分区表)。

输入create partition primary size=\*\*\* id=yyy创建数据分区,可用于安装操作系统(id=yyy可以不必输入,它代表的是GUID分区类型)。

然后再重复创建多个数据分区。如果整个磁盘创建一个分区的话输入create partition primary

输入list partition查看分区的编号。

输入select partition x(x为分区编号)选中分区。

输入assign letter=D(D代表你要给它分配的盘符,填一个没被占用的盘符)

输入exit退出diskpart分区工具,重新回到命令行。

这时进入我的电脑就能看见多了一个磁盘出来了,格式化就可以用了。

也可以在命令行下面输入以下命令格式化

输入`format D: \ /fs:ntfs \ /q \ /y`格式化这个EFI分区（D为刚才在上面设置的盘符，ntfs是文件格式）。这样一个新盘就挂上去了。

## CentOS yum源的配置

### 将金山云yum源切换成系统自带源

#### 1、切换yum源配置目录

```
cd /etc/yum.repos.d/  
ls
```

```
[root@vm172-31-0-8 yum.repos.d]# cd /etc/yum.repos.d  
[root@vm172-31-0-8 yum.repos.d]# ls  
backup CentOS.repo Epel.repo Rpmforge.repo
```

#### 2、备份金山云yum源名（也许以后还会用到该文件）

```
mv CentOS.repo CentOS_repo
```

```
[root@vm172-31-0-8 yum.repos.d]# mv CentOS.repo CentOS_repo
```

#### 3、将系统自带yum源配置文件拷贝到当前目录来使用

```
cp backup/CentOS-Base.repo .  
ls
```

```
[root@vm172-31-0-8 yum.repos.d]# cp backup/CentOS-Base.repo .  
[root@vm172-31-0-8 yum.repos.d]# ls  
backup CentOS-Base.repo CentOS_repo Epel.repo Rpmforge.repo  
[root@vm172-31-0-8 yum.repos.d]#
```

#### 4、更新

```
yum update
```

### 将系统自带源切换成金山云yum源

#### 1、切换目录

```
cd /etc/yum.repos.d/  
ls
```

```
[root@vm172-31-0-8 yum.repos.d]# cd /etc/yum.repos.d  
[root@vm172-31-0-8 yum.repos.d]# ls  
backup CentOS-Base.repo CentOS_repo Epel.repo Rpmforge.repo
```

#### 2、删除系统自带源yum配置文件，使用金山云的yum配置文件

```
rm -f CentOS-Base.repo  
mv CentOS_repo CentOS.repo
```

```
[root@vm172-31-0-8 yum.repos.d]# rm -f CentOS-Base.repo
[root@vm172-31-0-8 yum.repos.d]# mv CentOS_repo CentOS.repo
[root@vm172-31-0-8 yum.repos.d]# ls
backup CentOS.repo Epel.repo Rpmforge.repo
[root@vm172-31-0-8 yum.repos.d]#
```

### 3、更新

```
yum update
```

## 更改linux yum源方法：

### 1、进入yum配置文件目录

```
cd /etc/yum.repos.d/
```

### 2、备份配置文件

```
mv CentOS-Base.repo CentOS-Base_repo_bak
```

### 3、下载网易的配置（或其他源配置文件）

```
wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
```

下载下来的文件名为CentOS6-Base-163.repo

### 4、重命名（重命名成配置文件中要求的文件名）

```
mv CentOS6-Base-163.repo CentOS-Base.repo
```

### 5、更新

```
yum update
```

## Linux取消fstab开机加载

### Linux取消fstab自动加载

#### 定义

文件/etc/fstab包含了静态文件系统信息，定义了存储设备和分区整合到整个系统的方式。将文件系统挂载点，对应的挂载卷（或者硬盘）和挂载附加信息写入fstab可以实现卷（或者硬盘）的开机自动挂载。

#### 文件示例

```
/etc/fstab
```

```
#
```

```
tmpfs
```

```
/tmp tmpfs nodev,nosuid 0 0
```

```
/dev/vdb / ext4 defaults,noatime 0 1
```

```
/dev/vdc /home ext4 defaults,noatime 0 2
```

### 取消fstab自动加载

通常在以下情况下要取消fstab的自动加载

- (1) 云服务器制作自定义镜像;
- (2) 云服务器文件系统挂载点和对应挂载卷（或者硬盘）对应关系变化;

### 取消fstab自动加载的步骤

- (1) 在Linux系统里，使用文件编辑工具如VIM打开/etc/fstab:

```
vim /etc/fstab
```

如果没有vim需要首先运行yum -y install vim安装。

- (2) 删除对应的fstab自动挂载项： 点击键盘“I”进入编辑，删除对应自动挂载项。
- (3) 保存并关闭/etc/fstab文件： 按“Esc”退出编辑，: wq保存退出。

## Windows防火墙设置

### 允许响应ping（允许ICMP协议）

```
netsh firewall set icmpsetting type=ALL mode=enable //开启ICMP协议（允许你的服务器响应ping）
```

### 放开DHCP请求端口（允许67、68端口）

```
netsh advfirewall firewall set portopening udp 67 enable  
netsh advfirewall firewall set portopening udp 68 enable
```

或者

```
netsh firewall set portopening udp 67 enable  
netsh firewall set portopening udp 68 enable
```

### 放开RDP远程桌面连接端口（允许3389端口）

```
netsh firewall set portopening TCP 3389 ENABLE
```

### 放开IIS服务器被访问端口（允许80端口）

```
netsh firewall set portopening TCP 80 ENABLE
```

### 放开HTTPS服务端口（允许443端口）

```
netsh firewall set portopening TCP 443 ENABLE
```

445端口为运营商封禁端口，不可用



## windows防火墙

防火墙(英文: firewall)是一项协助确保信息安全的设备, 会依照特定的规则, 允许或是限制传输的数据通过。防火墙可以是一台专属的硬件也可以是架设在一般硬件上的一套软件, Windows防火墙是在Windows操作系统中系统自带的软件防火墙。

### cmd命令配置

```
? // 显示命令列表
add // 添加防火墙配置
delete // 删除防火墙配置
dump // 显示一个配置脚本
help // 显示命令列表
reset // 将防火墙配置重置为默认值
set // 设置防火墙配置
show // 显示防火墙配置
add allowedprogram // 添加防火墙允许的程序配置
add portopening // 添加防火墙端口配置
delete allowedprogram // 删除防火墙允许的程序配置
delete portopening // 删除防火墙端口配置
set allowedprogram // 设置防火墙允许的程序配置
set icmpsetting // 设置防火墙 ICMP 配置
set logging // 设置防火墙记录配置
set multicastbroadcastresponse // 设置防火墙多播/广播响应配置
set notifications // 设置防火墙通知配置
set opmode // 设置防火墙操作配置
set portopening // 设置防火墙端口配置
set service // 设置防火墙服务配置
show allowedprogram // 显示防火墙允许的程序配置
show config // 显示防火墙配置。
show currentprofile // 显示当前防火墙配置文件
show icmpsetting // 显示防火墙 ICMP 配置
show logging // 显示防火墙记录配置
show multicastbroadcastresponse // 显示防火墙多播/广播响应配置
show notifications // 显示防火墙操作配置
show opmode // 显示防火墙端口配置
show portopening // 显示防火墙端口配置
show service // 显示防火墙服务配置
show state // 显示当前防火墙状态
```

### 实例

```
netsh firewall show allowedprogram //查看防火墙放行的程序
netsh firewall set portopening TCP 445 ENABLE //打开445端口
netsh firewall set portopening TCP 3389 ENABLE // 打开3389端口
netsh firewall delete allowedprogram C:\A.exe //删除放行程序A.exe
netsh firewall set allowedprogram C:\A.exe A ENABLE //添加程序C盘下的A.exe并放行
netsh firewall add allowedprogram C:\A.exe A ENABLE //添加程序C盘下的A.exe并放行
netsh firewall set icmpsetting type=ALL mode=enable //开启ICMP协议
netsh firewall set icmpsetting type=2 mode=enable //允许出站数据包太大
```

## Linux防火墙配置

### Linux防火墙配置

通过iptables可以为Linux服务器配置有动态的防火墙, 能够指定并记住为发送或接收信息包所建立的连接的状态, 是一套用来设置、维护和检查Linux内核的IP包过滤规则的命令包。 table--Linux的iptables防火墙默认有三种表, Filter、NAT与

Mangle, 当然还有自定义的, 其中Filter即是默认使用的表格, chain一条链, 比如filter有INPUT、OUTPUT、FORWARD三条链。下面以建一个filter表防火墙为例

## 1. 清除原有规则

```
[root@tp ~]# iptables -F      清除预设表filter中的所有规则链的规则
[root@tp ~]# iptables -X      清除预设表filter中使用者自定链中的规则
```

## 2. 设定预设规则

```
[root@tp ~]# iptables -P INPUT DROP
[root@tp ~]# iptables -P OUTPUT ACCEPT
[root@tp ~]# iptables -P FORWARD DROP
```

## 3. 添加规则

远程SSH登录（开启22端口）

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@tp ~]# iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

WEB服务器（开启80端口）

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

邮件服务器（开启25、110端口）

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 110 -j ACCEPT
[root@tp ~]# iptables -A INPUT -p tcp --dport 25 -j ACCEPT
```

FTP服务器（开启21端口）

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 21 -j ACCEPT
```

DNS服务器（开启53端口）

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 53 -j ACCEPT
```

HTTPS(开启443端口)

```
[root@tp ~]# iptables -A INPUT -p tcp --dport 443-j ACCEPT
```

445端口为运营商封禁端口, 不可用

如果你还做了其他的服务器, 需要开启哪个端口, 照写就行了.

上面主要写的都是INPUT链, 凡是不在上面的规则里的, 都DROP

允许ICMP（允许ping）

```
[root@tp ~]# iptables -A OUTPUT -p icmp -j ACCEPT (OUTPUT设置成DROP的话)
[root@tp ~]# iptables -A INPUT -p icmp -j ACCEPT (INPUT设置成DROP的话)
```

允许loopback(不然会导致DNS无法正常关闭等问题)

```
IPTABLES -A INPUT -i lo -p all -j ACCEPT (如果是INPUT DROP)
IPTABLES -A OUTPUT -o lo -p all -j ACCEPT(如果是OUTPUT DROP)
```

## 著名的端口

端口	名称	功能
1	tcpmux	TCP端口服务多路复用
5	rje	远程作业入口
7	echo	Echo服务
9	discard	用于连接测试的空服务
11	systat	用于列举连接了的端口的系统状态
13	daytime	给请求主机发送日期和时间
17	qotd	给连接了的主机发送每日格言
18	msp	消息发送协议
19	chargen	字符生成服务；发送无止境的字符流
20	ftp-data	FTP数据端口
21	ftp	文件传输协议（FTP）端口；有时被文件服务协议（FSP）使用
22	ssh	安全shell (SSH) 服务
23	telnet	Telnet服务
25	smtp	简单邮件传输协议（SMTP）
37	time	时间协议
39	rlp	资源定位协议
42	nameserver	互联网名称服务
43	nicname	WHOIS目录服务
49	tamacas	用于给予TCP/IP验证和访问的终端访问控制器访问控制系统
50	re-mail-ck	远程邮件检查协议
53	domain	域名服务（如BIND）
63	whois++	WHOIS++，被扩展了的WHOIS服务
67	bootps	引导协议（BOOTP）服务；还被动态主机配置协议（DHCP）服务使用
68	bootpc	Bootstrap(BOOTP)客户；还被动态主机配置协议（DHCP）客户使用
69	tftp	小文件传输协议（TFTP）

70	gopher	Gopher互联网文档搜寻与检索
71	netrjs-1	远程作业服务
72	netrjs-2	远程作业服务
73	netrjs-3	远程作业服务
74	netrjs-4	远程作业服务
79	finger	用于用户联系信息的Finger服务
80	http	用于万维网（WWW）服务的超文本传输协议（HTTP）
88	kerberos	Kerberos网络验证系统
95	supdup	Telnet协议扩展
101	hostname	SRI-NIC机器上的主机名服务
102	iso-tsap	ISO开发环境（ISODE）网络应用
105	csnet-ns	邮箱名称服务器；也被CSO名称服务器使用
107	rtelnet	远程Telnet
109	pop2	邮局协议版本2
110	pop3	邮局协议版本3
111	sunrpc	用于远程命令执行过程调用（RPC）协议，被网络文件系统（NFS）使用
113	auth	验证和身份识别协议
115	sftp	安全文件传输协议（SFTP）服务
117	uucp-path	Unix到Unix复制协议（UUCP）路径服务
119	nntp	用于USENET讨论系统的网络新闻传输协议（NNTP）
123	ntp	网络时间协议（NTP）
137	netbios-ns	在红帽企业Linux中被Samba使用的NETBIOS名称服务
138	netbios-dgm	在红帽企业Linux中被Samba使用的NETBIOS名称服务
139	netbios-ssn	在红帽企业Linux中被Samba使用的NETBIOS名称服务
143	imap	互联网消息存取协议（IMAP）
161	snmp	简单网络管理协议（SNMP）
162	snmptrap	SNMP的陷阱
163	cmip-man	通用管理信息协议（CMIP）
164	cmip-agent	通用管理信息协议（CMIP）
174	mailq	MAILQ
177	Xdmcp	X显示器控制管理协议
178	nextstep	NEXTStep窗口服务器
179	bgp	边界网络协议
191	preospero	Clifford Neuman的Prospero服务

194	irc	互联网中继聊天 (IRC)
199	smux	SNMP UNIX多路复用
201	at-rtmp	Apple Talk选路
202	at-nbp	Apple Talk名称绑定
204	at-echo	Apple Talk echo服务
206	at-zis	Apple Talk区块信息
209	qmtmp	快速右键传输协议 (QMTP)
210	z39.50	NISO Z39.50数据库
213	ipx	互连网络分组交换协议 (IPX), 被Novell Netware 环境常用的数据报协议
220	imap3	互联网消息存取协议版本3
245	link	LINK
347	fatserv	Fatmen 服务器
363	rsvp_tunnel	RSVP 隧道
369	rpc2portmap	Coda 文件系统端口映射器
370	codauth2	Coda 文件系统验证服务
372	ulistproc	UNIX Listserv
389	ldap	轻型目录存取协议 (LDAP)
427	svrloc	服务位置协议 (SLP)
434	mobileip-agent	可移互联网协议 (IP) 代理
435	mobileip-mn	可移互联网协议 (IP) 管理器
443	https	安全超文本传输协议 (HTTP)
444	snpp	小型网络分页协议
445 (运营商禁用)	microsoft-ds	通过 TCP/IP 的服务器消息块 (SMB)
464	kpasswd	Kerberos 口令和钥匙改换服务
468	photuris	Photuris 会话钥匙管理协议
487	saft	简单不对称文件传输 (SAFT) 协议
488	gss-http	用于 HTTP 的通用安全服务 (GSS)
496	pim-rp-disc	用于协议独立的多址传播 (PIM) 服务的会合点发现 (RP-DISC)
500	isakmp	互联网安全关联和钥匙管理协议 (ISAKMP)
535	iiop	互联网内部对象请求代理协议 (IIOP)
538	gdomapGNustep	分布式对象映射器 (GDOMAP)
546	dhcpv6-client	动态主机配置协议 (DHCP) 版本6客户
547	dhcpv6-server	动态主机配置协议 (DHCP) 版本6服务
554	rtsp	实时流播协议 (RTSP)

563	nntps	通过安全套接字层的网络新闻传输协议 (NNTPS)
565	whoami	whoami
587	submission	邮件消息提交代理 (MSA)
610	npmp-local	网络外设管理协议 (NPMP) 本地 / 分布式排队系统 (DQS)
611	npmp-gui	网络外设管理协议 (NPMP) GUI / 分布式排队系统 (DQS)
	612	hmmp-indHMMP 指示 / DQS631ipp互联网打印协议 (IPP)
636	ldaps	通过安全套接字层的轻型目录访问协议 (LDAPS)
674	acap	应用程序配置存取协议 (ACAP)
694	ha-cluster	用于带有高可用性的群集的心跳服务
749	kerberos-adm	Kerberos 版本5 (v5) 的 “kadmin” 数据库管理
750	kerberos-iv	Kerberos 版本4 (v4) 服务765webster网络词典
767	phonebook	网络电话簿
873	rsyncrsync	文件传输服务
992	telnets	通过安全套接字层的 Telnet (TelnetS)
993	imaps	通过安全套接字层的互联网消息存取协议 (IMAPS)
994	ircs	通过安全套接字层的互联网中继聊天 (IRCS)
995	pop3s	通过安全套接字层的邮局协议版本3 (POPS3)

## UNIX 特有的端口

端口	名称	功能
512/tcp	exec	用于对远程执行的进程进行验证
512/udp	biff [comsat]	异步邮件客户 (biff) 和服务 (comsat)
513/tcp	login	远程登录 (rlogin)
513/udp	who [whod]	登录的用户列表
514/tcp	shell [cmd]	不必登录的远程 shell (rshell) 和远程复制 (rcp)
514/udp	syslog	UNIX 系统日志服务
515	printer [spooler]	打印机 (lpr) 假脱机
517/udp	talk	远程对话服务和客户

518/udp	ntalk	网络交谈 (ntalk), 远程对话服务和客户519utime [unixtime]UNIX 时间协议 (utime)
520/tcp	efs	扩展文件名服务器 (EFS)
520/udp	router [route, routed]	选路信息协议 (RIP)
521	ripng	用于互联网协议版本6 (IPv6) 的选路信息协议
525	timed [timeserver]	时间守护进程 (timed)
526/tcp	tempo [newdate]	Tempo
530/tcp	courier [rpc]Courier	远程过程调用 (RPC) 协议
531/tcp	conference [chat]	互联网中继聊天
532	netnews	Netnews
533/udp	netwall	用于紧急广播的 Netwall
540/tcp	uucp [uucpd]	Unix 到 Unix 复制服务
543/tcp	klogin	Kerberos 版本5 (v5) 远程登录
544/tcp	kshell	Kerberos 版本5 (v5) 远程 shell
548	afpovertcp	通过传输控制协议 (TCP) 的 Appletalk 文件编制协议 (AFP)
556	remotefs [rfs_server, rfs]	Brunhoff 的远程文件系统 (RFS)

## 注册的端口

端口	名称	功能
1080	socks	SOCKS 网络应用程序代理服务
1236	bvcontrol [rmtcfg]	Garcilis Packeten 远程配置服务器
1300	h323hostcallsc	H. 323 电话会议主机电话安全
1433	ms-sql-s	Microsoft SQL 服务器
1434	ms-sql-m	Microsoft SQL 监视器
1494	icaCitrix	ICA 客户
1512	winsMicrosoft	Windows 互联网名称服务器
1524	ingreslock	Ingres 数据库管理系统 (DBMS) 锁定服务

1525	prospero-np	无特权的 Prospero
1645	datametrics [old-radius]	Datametrics / 从前的 radius 项目
1646	sa-msg-port [oldradacct]	sa-msg-port / 从前的 radacct 项目
1649	kermitKermit	文件传输和管理服务
1701	l2tp [l2f]	第2层隧道服务 (LT2P) / 第2层转发 (L2F)
1718	h323gatedisc	H. 323 电讯守门装置发现机制
1719	h323gatestat	H. 323 电讯守门装置状态
1720	h323hostcall	H. 323 电讯主持电话设置
1758	tftp-mcast	小文件 FTP 组播
1759	mtftp	组播小文件 FTP (MTFTP)
1789	hello	Hello 路由器通信端口
1812	radius	Radius 拨号验证和记帐服务
1813	radius-acct	Radius 记帐
1911	mtp	Starlight 网络多媒体传输协议 (MTP)
1985	hsrp	Cisco 热备用路由器协议
1986	licensedaemon	Cisco 许可管理守护进程
1997	gdp-port	Cisco 网关发现协议 (GDP)
2049	nfs [nfsd]	网络文件系统 (NFS)
2102	zephyr-srv	Zephyr 通知传输和发送服务器
2103	zephyr-clt	Zephyr serv-hm 连接
2104	zephyr-hm	Zephyr 主机管理器
2401	cvspserver	并行版本系统 (CVS) 客户 / 服务器操作
2430/tcp	venus	用于 Coda 文件系统 (codacon 端口) 的 Venus 缓存管理器
2430/udp	venus	用于 Coda 文件系统 (callback/wbc interface 界面) 的 Venus 缓存管理器
2431/tcp	venus-se	Venus 传输控制协议 (TCP) 的副作用
2431/udp	venus-se	Venus 用户数据报协议 (UDP) 的副作用
2432/udp	codasrv	Coda 文件系统服务器端口
2433/tcp	codasrv-se	Coda 文件系统 TCP 副作用



2433/udp	codasrv-se	Coda 文件系统 UDP SFTP 副作用
2600	hpstgmgr [zebrasrv]	HPSTGMGR; Zebra 选路
2601	discp-client [zebra]	discp 客户; Zebra 集成的 shell
2602	discp-server [ripd]	discp 服务器; 选路信息协议守护进程 (ripd)
2603	servicemeter [ripngd]	服务计量; 用于 IPv6 的 RIP 守护进程
2604	nsc-ccs [ospfd]	NSC CCS; 开放式短路径优先守护进程 (ospfd)
2605	nsc-posa	NSC POSA; 边界网络协议守护进程 (bgpd)
2606	netmon [ospf6d]	Dell Netmon; 用于 IPv6 的 OSPF 守护进程 (ospf6d)
2809	corbaloc	公共对象请求代理体系 (CORBA) 命名服务定位器
3130	icpv2	互联网缓存协议版本2 (v2); 被 Squid 代理缓存服务器使用
3306	mysql	MySQL 数据库服务
3346	trnsprntproxy	Trnsprnt 代理
4011	pxe	执行前环境 (PXE) 服务
4321	rwhois	远程 Whois (rwhois) 服务
4444	krb524	Kerberos 版本5 (v5) 到版本4 (v4) 门票转换器
5002	rfe	无射频以太网 (RFE) 音频广播系统
5308	cfengine	配置引擎 (Cfengine)
5999	cvsup [CVSup]	CVSup 文件传输和更新工具
6000	x11 [X]	X 窗口系统服务
7000	afs3-fileserver	Andrew 文件系统 (AFS) 文件服务器
7001	afs3-callback	用于给缓存管理器回电的 AFS 端口
7002	afs3-prserver	AFS 用户和组群数据库
7003	afs3-vlserver	AFS 文件卷位置数据库
7004	afs3-kaserver	AFS Kerberos 验证服务
7005	afs3-volser	AFS 文件卷管理服务器
7006	afs3-errors	AFS 错误解释服务
7007	afs3-bos	AFS 基本监查进程
7008	afs3-update	AFS 服务器到服务器更新器
7009	afs3-rmtsys	AFS 远程缓存管理器服务
9876	sd	会话指引器
10080	amanda	高级 Maryland 自动网络磁盘归档器 (Amanda) 备份服务
11371	pgpkeyserver	良好隐私 (PGP) / GNU 隐私卫士 (GPG) 公钥服务器
11720	h323callsigalt	H. 323 调用信号交替

13720	bprd	Veritas NetBackup 请求守护进程 (bprd)
13721	bpdbm	Veritas NetBackup 数据库管理器 (bpdbm)
13722	bpjava-msvc	Veritas NetBackup Java / Microsoft Visual C++ (MSVC) 协议
13724	vnetd	Veritas 网络工具
13782	bpcd	Veritas NetBackup
13783	vopied	Veritas VOPIED 协议
22273	wnn6 [wnn4]	假名/汉字转换系统
26000	quake	Quake (以及相关的) 多人游戏服务器
26208	wnn6-ds	
33434	traceroute	Traceroute 网络跟踪工具

## 未注册端口

端口	名称	功能
15/tcp	netstat	网络状态 (netstat)
98/tcp	linuxconf	Linuxconf Linux 管理工具
106	poppassd	邮局协议口令改变守护进程 (POPPASSD)
465/tcp	smtps	通过安全套接字层的简单邮件传输协议 (SMTPS)
616/tcp	gii	使用网关的 (选路守护进程) 互动界面
808	omirr [omirrd]	联机镜像 (Omirr) 文件镜像服务
871/tcp	supfileserv	软件升级协议 (SUP) 服务器
901/tcp	swat	Samba 万维网管理工具 (SWAT)
953	rndc	Berkeley 互联网名称域版本9 (BIND 9) 远程名称守护进程配置工具
1127	sufiledbg	软件升级协议 (SUP) 调试
1178/tcp	skkserv	简单假名到汉字 (SKK) 日文输入服务器
1313/tcp	xtel	法国 Minitel 文本信息系统
1529/tcp	support [prmsd, gnatsd]	GNATS 错误跟踪系统
2003/tcp	cfinger	GNU Finger 服务
2150	ninstall	网络安装服务
2988	afbackup	afbackup 客户-服务器备份系统
3128/tcp	squid	Squid 万维网代理缓存
3455	prsvp	RSVP 端口
5432	postgres	PostgreSQL 数据库
4557/tcp	fax	FAX 传输服务 (旧服务)

4559/tcp	hylafax	HylaFAX 客户-服务器协议 (新服务)
5232	sgi-dgl	SGI 分布式图形库
5354	noclog	NOCOL 网络操作中心记录守护进程 (noclogd)
5355	hostmon	NOCOL 网络操作中心主机监视
5680/tcp	canna	Canna 日文字符输入界面
6010/tcp	x11-ssh-offset	安全 Shell (SSH) X11 转发偏移
6667	ircd	互联网中继聊天守护进程 (ircd)
7100/tcp	xf86	X 字体服务器 (XFS)
7666/tcp	tircproxy	Tircproxy IRC 代理服务
8008	http-alt	超文本传输协议 (HTTP) 的另一选择
8080	webcache	万维网 (WWW) 缓存服务
8081	tproxy	透明代理
9100/tcp	jetdirect [laserjet, hplj]	Hewlett-Packard (HP) JetDirect 网络打印服务
9359	mandelspawn [mandelbrot]	用于 X 窗口系统的并行 Mandelbrot 生成程序
10081	kamanda	使用 Kerberos 的 Amanda 备份服务
10082/tcp	amandaidx	Amanda 备份服务
10083/tcp	amidxtape	Amanda 备份服务
20011	isdnlog	综合业务数字网 (ISDN) 登录系统
20012	vboxd	ISDN 音箱守护进程 (vboxd)
22305/tcp	wnn4_Kr	kWnn 韩文输入系统
22289/tcp	wnn4_Cn	cWnn 中文输入系统
22321/tcp	wnn4_Tw	tWnn 中文输入系统 (台湾)
24554	binkp	Binkley TCP/IP Fidonet 邮寄程序守护进程
27374	asp	地址搜索协议
60177	tfido	Ifmail FidoNet 兼容邮寄服务
60179	fido	FidoNet 电子邮件和新闻网络