命令行

2016-3-14

作者：朱林峰

目录

命令行

1 shell

1. 清理屏幕

clear

reset

printf "\033c"

2. 定位命令的可执行文件

type command：报告命令的类型或是位置

which command：和 type 一样

whereis command：报告命令可执行文件的位置

3. 命令的摘要式说明

whatis command

4. 给命令定义别名

alias l='ls -l'

5. 定时重复执行命令

watch -n sec command

6. 以守护进程方式启动程序

./a.out &

7. 重复上一条命令

!!

8. 无记录执行命令

<space>command

9. 给定时间执行命令

poweroff | at 08:10

10. 连接两条命令

&&

11. 显示命令使用记录

history

12. 执行某条历史命令

!number

13. 切换到上一次的路径

cd -

14. 显示当前路径

pwd

15. tee：重定向输出到指定文件并同时在终端显示

2 系统管理

2.1 查看系统基本属性

1. **who：显示系统当前登录用户**

- 示例用法

```
zlf@ubuntu:~$ who -H
NAME     LINE         TIME             COMMENT
zlf      tty1         2016-03-14 15:05
zlf      pts/0        2016-03-14 15:07 (10.12.76.5)
```

终端显示当前有两个用户登录系统，分别通过 tty 和 pts。

2. id：显示当前实际有效的用户 ID 和组 ID

- 示例用法

```
zlf@ubuntu:~$ id
uid=1000(zlf) gid=1000(zlf) groups=1000(zlf),4(adm),24(cdrom),27(sudo),30(dip),
46(plugdev),114(sambashare),120(lpadmin)
```

3. 显示主机名

```
hostname - show or set the system's host name
domainname - show or set the system's NIS/YP domain name
ypdomainname - show or set the system's NIS/YP domain name
nisdomainname - show or set the system's NIS/YP domain name
dnsdomainname - show the system's DNS domain name
hostnamectl - may be used to query and change the system hostname and related
settings.
```

4. uname：查看系统版本信息

```
zlf@ubuntu:~$ uname -a
Linux ubuntu 3.19.0-15-generic #15-Ubuntu SMP Thu Apr 16 23:32:37 UTC 2015
x86_64 x86_64 x86_64 GNU/Linux

内核名：Linux
发行版：Ubuntu
内核版本：3.19.0-15-generic
系统版本：#15-Ubuntu SMP Thu Apr 16 23:32:37 UTC 2015
操作系统：GNU/Linux
```

5. lsb_release：查看系统版本

```
zlf@ubuntu:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 15.04
Release: 15.04
Codename: vivid
```

6. last：打印系统登录日志

7. env：查看、设置环境变量，并让一个程序在此被临时修改过的环境上执行

- 查看

```
zlf@ubuntu:~/project/c$ env
XDG_SESSION_ID=5
TERM=xterm-256color
SHELL=/bin/bash
SSH_CLIENT=10.12.76.5 52052 22
SSH_TTY=/dev/pts/2
USER=zlf
...
```

- 设置&执行

```
zlf@ubuntu:~/project/c$ env USER=ZLF ./a.out
hello world
```

1. locale：查看或设置系统语言环境

2.2 时间

1. date：显示当前时间

2. cal：显示日历

- 显示本月日历

```
zlf@ubuntu:~$ cal
     March 2016
```

```
Su Mo Tu We Th Fr Sa
       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
```

- 显示某年的日历

```
cal 2016
```

3. when：显示当前日期，并可以编辑日历

4. calendar：备忘录

2.3 系统监控

1. ps：静态显示系统当前的进程

- 默认用法：显示当前有效用户下的由当前终端激活的进程

```
zlf@ubuntu:~/document$ ps
  PID TTY          TIME CMD
 1452 pts/0    00:00:00 bash
13456 pts/0    00:00:00 ps
```

- 显示系统所有进程

```
ps -e
ps -A
ps -ax
```

- 树形结构显示进程

```
ps -ejH
ps -axjf
```

- 显示线程有关信息

```
ps -eLf
ps -axms
```

2. top：动态实时显示系统进程

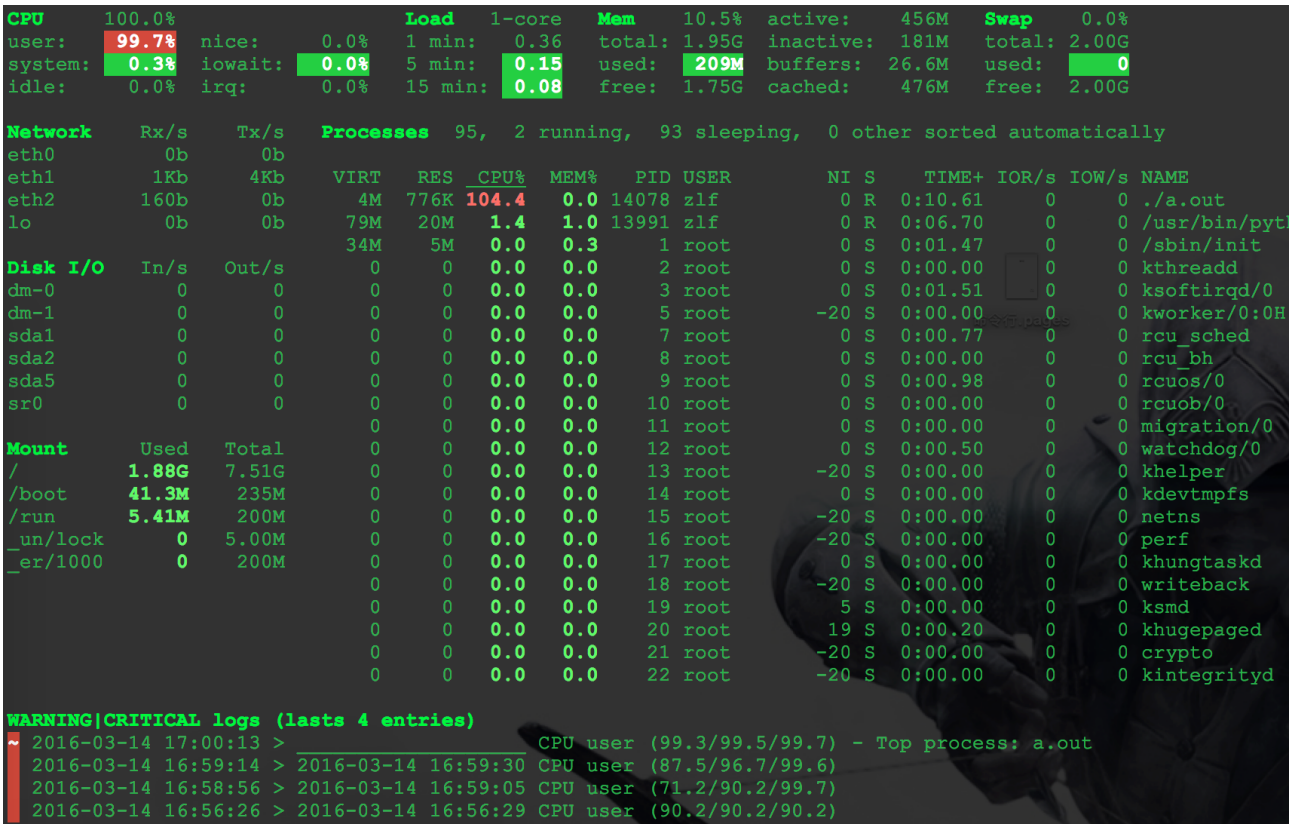
```
按键盘 b:打开或关闭当前活跃进程的高亮显示
按键盘 x:打开或关闭按 CPU 一栏的高亮显示
shift+> 或 shift+< :改变排序方式
按键盘 h:进入帮助视图
按键盘 m:打开或关闭内存信息显示
```

3. pstree：以树形结构静态显示系统进程

4. pgrep：是 ps | grep 的综合，根据模式查找符合条件的进程

5. glances：实时显示当前 CPU、内存、网络、磁盘状态，适应终端屏幕大小，自动高亮预警

```
a – 对进程自动排序
c – 按 CPU 百分比对进程排序
m – 按内存百分比对进程排序
p – 按进程名字母顺序对进程排序
i – 按读写频率（I/O）对进程排序
d – 显示/隐藏磁盘 I/O 统计信息
f – 显示/隐藏文件系统统计信息
n – 显示/隐藏网络接口统计信息
s – 显示/隐藏传感器统计信息
y – 显示/隐藏硬盘温度统计信息
l – 显示/隐藏日志（log）
b – 切换网络 I/O 单位（Bytes/bits）
w – 删除警告日志
x – 删除警告和严重日志
1 – 切换全局 CPU 使用情况和每个 CPU 的使用情况
```




```
h — 显示/隐藏这个帮助画面
t — 以组合形式浏览网络 I/O
u — 以累计形式浏览网络 I/O
q — 退出（‘ESC‘ 和 ‘Ctrl&C‘ 也可以）
```

6. free：查看系统内存使用状况

7. nmon dstat saidar ccze

8. vmstat：系统资源状态的监控工具，报告整个系统的 CPU，内存，I/O 的状态

```
zlf@ubuntu:~$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 2  0      0 1343780  29804 487124    0    0    28    66   72  125  1  0 99  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   76  121  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   61   96  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   60  101  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   54   93  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   55   90  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   43   70  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   50   80  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   43   74  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   57   91  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   48   84  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   53   77  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   52   87  0  0 100  0  0
 0  0      0 1343764  29804 487124    0    0     0     0   59   85  0  0 100  0  0
```

```
r     处于 run 状态的进程个数
b     表示阻塞的进程
swpd  虚拟内存已使用的大小，如果大于0，表示你的机器物理内存不足了
free  空闲的物理内存的大小
buff  缓冲区
cache 高速缓存
si    每秒从磁盘读入虚拟内存的大小，如果这个值大于0，表示物理内存不够用或者内存泄露了
so    每秒虚拟内存写入磁盘的大小，如果这个值大于0，同上。
```

```
bi    块设备每秒接收的块数量，这里的块设备是指系统上所有的磁盘和其他块设备，默认块大小是1024byte
bo    块设备每秒发送的块数量，例如我们读取文件，bo就要大于0。bi和bo一般都要接近0，不然就是IO过于频繁，需要调整。
in    每秒CPU的中断次数，包括时间中断
cs    每秒上下文切换次数，例如我们调用系统函数，就要进行上下文切换，线程的切换，也要进程上下文切换，这个值要越小越好，太大了，要考虑调低线程或者进程的数目,例如在apache和nginx这种web服务器中，我们一般做性能测试时会进行几千并发甚至几万并发的测试，选择web服务器的进程可以由进程或者线程的峰值一直下调，压测，直到cs到一个比较小的值，这个进程和线程数就是比较合适的值了。系统调用也是，每次调用系统函数，我们的代码就会进入内核空间，导致上下文切换，这个是很耗资源，也要尽量避免频繁调用系统函数。上下文切换次数过多表示你的CPU大部分浪费在上下文切换，导致CPU干正经事的时间少了，CPU没有充分利用，是不可取的。
us    用户CPU时间
sy    系统CPU时间，如果太高，表示系统调用时间长，例如是IO操作频繁。
id    空闲 CPU时间，一般来说，id + us + sy = 100
wt    等待IO CPU时间。
```

9. iostat：监控系统设备的IO负载情况

10. sysdig：系统故障分析排查工具，可以捕捉过滤系统调用和其他系统事件

11. sysctl：查看或动态调整内核参数，可以显著提高系统性能

- 查看

```
读取所有变量
sysctl -a

读一个指定的变量，例如 kern.maxproc
sysctl kern.maxproc
>kern.maxproc: 1044
```

- 设置

```
设置一个指定的变量,直接用 variable=value 这样的语法
sysctl kern.maxfiles=5000
```

12. dmesg：查看或控制内核的 ring buffer，主要是关于启动的一些日志信息。

ring buffer：在 Linux 中, 所有的系统信息(包内核信息)都会传送到 ring buffer 中。而内核产生的信息由 printk() 打印出来。系统启动时所看到的信息都是由该函数打印到屏幕中。printk() 打出的信息往往以 <0><2>… 这的数字表明消息的重要级别。高于一定的优先级别会打印到屏幕上， 否则只会保留在系统的缓冲区中 (ring buffer)。

13. ulimit -a：系统限制

```
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 7913
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 7913
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

14. /var/log：此目录包含系统各类日志

- /var/log/messages — 包括整体系统信息，其中也包含系统启动期间的日志。此外，mail，cron，daemon，kern和auth等内容也记录在var/log/messages日志中。
- /var/log/dmesg — 包含内核缓冲信息（kernel ring buffer）。在系统启动时，会在屏幕上显示许多与硬件有关的信息。可以用dmesg查看它们。
- /var/log/auth.log — 包含系统授权信息，包括用户登录和使用的权限机制等。
- /var/log/boot.log — 包含系统启动时的日志。
- /var/log/daemon.log — 包含各种系统后台守护进程日志信息。
- /var/log/dpkg.log — 包括安装或dpkg命令清除软件包的日志。
- /var/log/kern.log — 包含内核产生的日志，有助于在定制内核时解决问题。
- /var/log/lastlog — 记录所有用户的最近信息。这不是一个ASCII文件，因此需要用lastlog命令查看内容。
- /var/log/maillog /var/log/mail.log — 包含来着系统运行电子邮件服务器的日志信息。例如，sendmail日志信息就全部送到这个文件中。
- /var/log/user.log — 记录所有等级用户信息的日志。
- /var/log/Xorg.x.log — 来自X的日志信息。
- /var/log/alternatives.log — 更新替代信息都记录在这个文件中。
- /var/log/btmp — 记录所有失败登录信息。使用last命令可以查看btmp文件。例如，"last -f /var/log/btmp | more"。
- /var/log/cups — 涉及所有打印信息的日志。
- /var/log/anaconda.log — 在安装Linux时，所有安装信息都储存在这个文件中。
- /var/log/yum.log — 包含使用yum安装的软件包信息。
- /var/log/cron — 每当cron进程开始一个工作时，就会将相关信息记录在这个文件中。
- /var/log/secure — 包含验证和授权方面信息。例如，sshd会将所有信息记录（其中包括失败登录）在这里。
- /var/log/wtmp或/var/log/utmp — 包含登录信息。使用wtmp可以找出谁正在登陆进入系统，谁使用命令显示这个文件或信息等。
- /var/log/faillog — 包含用户登录失败信息。此外，错误登录命令也会记录在本文件中。
- /var/log/httpd/或/var/log/apache2 — 包含服务器access_log和error_log信息。
- /var/log/lighttpd/ — 包含light HTTPD的access_log和error_log。
- /var/log/mail/ — 这个子目录包含邮件服务器的额外日志。
- /var/log/prelink/ — 包含.so文件被prelink修改的信息。
- /var/log/audit/ — 包含被 Linux audit daemon储存的信息。
- /var/log/samba/ — 包含由samba存储的信息。
- /var/log/sa/ — 包含每日由sysstat软件包收集的sar文件。
- /var/log/sssd/ — 用于守护进程安全服务。

除了手动存档和清除这些日志文件以外，还可以使用logrotate在文件达到一定大小后自动删除。可以尝试用vi，tail，grep和less等命令查看这些日志文件。

3 程序管理

3.1 程序控制

1. kill：给指定进程发送信号，也可以直接杀死进程

- 列出所有信号

```
zlf@ubuntu:~/project/c$ 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
```

- 给进程发送信号

```
kill -14 123 254
给 pid = 123, pid = 254 的进程发送 SIGALRM 信号
```

- 杀死进程

```
kill 123
杀死 pid = 123  的进程
kill -9 -1
杀死所有能杀死的进程
```

2. pkill：kill | grep 的综合，根据指定的模式找到对应的进程，并向它们发送信号

3.2 程序监控

1. /proc 文件系统

Linux系统上的 /proc 目录是一种文件系统，即 proc 文件系统。与其它常见的文件系统不同的是，/proc 是一种伪文件系统（也即虚拟文件系统），存储的是当前内核运行状态的一系列特殊文件，用户可以通过这些文件查看有关系统硬件及当前正在运行进程的信息，甚至可以通过更改其中某些文件来改变内核的运行状态。

基于 /proc 文件系统如上所述的特殊性，其内的文件也常被称作虚拟文件，并具有一些独特的特点。例如，其中有些文件虽然使用查看命令查看时会返回大量信息，但文件本身的大小却会显示为0字节。此外，这些特殊文件中大多数文件的时间及日期属性通常为当前系统时间和日期，这跟它们随时会被刷新（存储于RAM中）有关。

为了查看及使用上的方便，这些文件通常会按照相关性进行分类存储于不同的目录甚至子目录中，如 /proc/scsi目录中存储的就是当前系统上所有 SCSI 设备的相关信息，/proc/N 中存储的则是系统当前正在运行的进程的相关信息，其中 N 为正在运行的进程（可以想象得到，在某进程结束后其相关目录则会消失）。

大多数虚拟文件可以使用文件查看命令如 cat、more 或者 less 进行查看，有些文件信息表述的内容可以一目了然，但也有文件的信息却不怎么具有可读性。不过，这些可读性较差的文件在使用一些命令如 apm、free、lspci 或 top 查看时却可以有着不错的表现。

- proc 根目录

```
1      1535   239   7    9            kallsyms     slabinfo
10     156    24    710  acpi         kcore        softirqs
11     157    242   712  asound       keys         stat
1124   1585   25    717  buddyinfo    key-users    swaps
1125   1598   253   734  bus          kmsg         sys
1127   16     26    742  cgroups      kpagecount   sysrq-trigger
12     16283  27    75   cmdline      kpageflags   sysvipc
125    16363  3     753  consoles     loadavg      thread-self
128    16514  31    76   cpuinfo      locks        timer_list
129    16516  32    763  crypto       mdstat       timer_stats
13     16518  33    781  devices      meminfo      tty
130    16525  344   788  diskstats    misc         uptime
13534  16823  372   8    dma          modules      version
13624  17     45    820  driver       mounts       version_signature
13625  18     46    846  execdomains  mtrr         vmallocinfo
139    19     47    875  fb           net          vmstat
14     2      48    876  filesystems  pagetypeinfo zoneinfo
140    20     49    877  fs           partitions
14175  21     5     888  interrupts   sched_debug
142    210    50    889  iomem        schedstat
144    22     55    890  ioports      scsi
15     23     5898  891  irq          self
```

根目录下，以数字编号命令的蓝色目录，对应于各个进程，目录名即进程号。

- devices

Character devices		Block devices	
1 mem	89 i2c	1 ramdisk	128 sd
4 /dev/vc/0	99 ppdev	259 blkext	129 sd
4 tty	108 ppp	7 loop	130 sd
4 ttyS	116 alsa	8 sd	131 sd
5 /dev/tty	128 ptm	9 md	132 sd
5 /dev/console	136 pts	11 sr	133 sd
5 /dev/ptmx	180 usb	65 sd	134 sd
5 ttyprintk	189 usb_device	66 sd	135 sd
7 vcs	226 drm	67 sd	252 device-mapper
10 misc	251 hidraw	68 sd 69 sd	253 virtblk
13 input	252 bsg	70 sd	254 mdp
21 sg	253 watchdog	71 sd	
29 fb	254 rtc		

- meminfo

```
MemTotal:        2049104 kB    //总共的物理内存,2G
MemFree:         1756060 kB    //空闲的物理内存
MemAvailable:    1781220 kB    //可用的物理内存
Buffers:           14816 kB    //用于块设备的缓冲大小
Cached:           121192 kB    //用于文件的缓冲大小
SwapCached:            0 kB    //已经被交换出的内存，存放在swapfile中
Active:           182848 kB    //最近经常被使用的内存，除非非常必要否则不会被移作他用
Inactive:          63324 kB    //最近不经常被使用的内存，非常用可能被用于其他途径
Active(anon):     110772 kB    //
Inactive(anon):     4740 kB
Active(file):      72076 kB
Inactive(file):    58584 kB
Unevictable:           0 kB
Mlocked:               0 kB    //
SwapTotal:       2097148 kB    //交换空间总和，2G
SwapFree:        2097148 kB    //交换空间剩余量
Dirty:                44 kB    //等待被写回到磁盘的内存大小
Writeback:             0 kB    //正在被写回到磁盘的内存大小
AnonPages:        110164 kB
Mapped:            45164 kB    //
Shmem:              5352 kB
Slab:              25280 kB    //内核数据结构缓存
SReclaimable:      14244 kB
SUnreclaim:        11036 kB
KernelStack:        1712 kB
PageTables:         8340 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     3121700 kB
Committed_AS:     284320 kB
```

```
VmallocTotal:   34359738367 kB
VmallocUsed:       11108 kB
VmallocChunk:   34359724028 kB
HardwareCorrupted:     0 kB
AnonHugePages:     65536 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:       51136 kB
DirectMap2M:     2045952 kB
```

- 进程目录

```
attr            cpuset   limits      net             projid_map  stat
autogroup       cwd      loginuid    ns              root        statm
auxv            environ  map_files   numa_maps       sched       status
cgroup          exe      maps        oom_adj         schedstat   syscall
clear_refs      fd       mem         oom_score       sessionid   task
cmdline         fdinfo   mountinfo   oom_score_adj   setgroups   timers
comm            gid_map  mounts      pagemap         smaps       uid_map
coredump_filter io       mountstats  personality     stack       wchan
```

cmdline文件是启动进程的命令行，内容如下：

```
zlf@ubuntu:/proc$ more 888/cmdline
/usr/sbin/smbd
```

可见进程号为 888 的进程是 smbd 服务。

environ 文件记录的是当前进程的环境变量信息。

limits 文件是当前进程的限制值，内容如下：

```
Limit                     Soft Limit           Hard Limit           Units
Max cpu time              unlimited            unlimited            seconds
Max file size             unlimited            unlimited            bytes
Max data size             unlimited            unlimited            bytes
Max stack size            8388608              unlimited            bytes
Max core file size        0                    unlimited            bytes
Max resident set          unlimited            unlimited            bytes
Max processes             7913                 7913                 processes
Max open files            1024                 65536                files
Max locked memory         65536                65536                bytes
Max address space         unlimited            unlimited            bytes
Max file locks            unlimited            unlimited            locks
Max pending signals       7913                 7913                 signals
Max msgqueue size         819200               819200               bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited            us
```

exe 是当前进程可执行文件的符号链接，执行它可启动当前进程的一个拷贝。

fd 目录包含当前进程打开的每一个文件的文件描述符（file descriptor），这些文件描述符是指向实际文件的一个符号链接：

```
zlf@ubuntu:/proc/888$ sudo ls ./fd
0  1  10  11  12  13  14  15  16  17  18  19  2  20  21  22  23  24  25  26  27
28  29  3  30 31  32 33  34 35  4  5  6  7 8  9
```

maps 文件是当前进程关联到的每个可执行文件和库文件在内存中的映射区域及其访问权限所组成的列表，部分内容如下：

物理地址	映射文件

f9afef89000-7f9afeff0000 rw-s 00000000 fc:00 278451	/var/lib/samba/private/passdb.tdb
7f9afeff0000-7f9aff057000 rw-s 00000000 fc:00 278994	/var/lib/samba/account_policy.tdb
7f9aff057000-7f9aff0be000 rw-s 00000000 00:11 14025	/run/samba/gencache_notrans.tdb
7f9aff0be000-7f9aff0ca000 r-xp 00000000 fc:00 799	/lib/x86_64-linux-gnu/libnss_files-2.21.so
7f9aff0ca000-7f9aff2c9000 ---p 0000c000 fc:00 799	/lib/x86_64-linux-gnu/libnss_files-2.21.so
7f9aff2c9000-7f9aff2ca000 r--p 0000b000 fc:00 799	/lib/x86_64-linux-gnu/libnss_files-2.21.so
7f9aff2ca000-7f9aff2cb000 rw-p 0000c000 fc:00 799	/lib/x86_64-linux-gnu/libnss_files-2.21.so

2. strace：捕捉进程的系统调用和信号，是诊断和调试的得力工具

3. ltrace：捕捉进程的动态链接库函数的调用、系统调用和信号

4. pmap：打印进程的虚拟内存到执行文件的映射表 memstat ipcs

5. time：运行一个程序，并统计其 CPU 时间开销

```
zlf@ubuntu:/var/log/apt$ time ls
history.log  term.log

real 0m0.002s
user 0m0.000s
sys 0m0.000s
```

6. timeout：限定程序运行时长。

3.3 程序文件

1. ldd：查看可执行文件的共享库依赖

```
zlf@ubuntu:~/project/c$ ldd -v a.out
    linux-vdso.so.1 =>  (0x00007ffe88585000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fa622671000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fa622a47000)

    Version information:
    ./a.out:
    libc.so.6 (GLIBC_2.2.5) => /lib/x86_64-linux-gnu/libc.so.6
    /lib/x86_64-linux-gnu/libc.so.6:
    ld-linux-x86-64.so.2 (GLIBC_2.3) => /lib64/ld-linux-x86-64.so.2
    ld-linux-x86-64.so.2 (GLIBC_PRIVATE) => /lib64/ld-linux-x86-64.so.2
```

2. objdump： 查看目标文件&可执行文件

```
-a 列举.a文件中所有的目标文件
-b <bfdname> 指定BFD名
-C 对于c++符号名进行反修饰
-g 显示调试信息
-d 对包含机器指令的段进行反汇编
-D 对所有段进行反汇编
-f 显示目标文件文件头
-h 显示段表
-l 显示符号信息
-p 显示专有头部信息，具体内容取决于文件格式
-r 显示重定位信息
-R 显示动态链接重定位信息
-s 显示文件所有内容
-S 显示源代码和反汇编代码（包含-d参数）
-W 显示文件中包含有DWARF调试信息格式的段
-t 显示文件中的符号表
```

```
-T 显示动态链接符号表
-x 显示文件的所有文件头
```

3. size：

4 文件管理

1. df：报告文件系统磁盘容量

```
-a 全部文件系统列表
-h 方便阅读方式显示
-H 等于“-h”，但是计算式，1K=1000，而不是1K=1024
-i 显示inode信息
-k 区块为1024字节
-l 只显示本地文件系统
-m 区块为1048576字节
--no-sync 忽略 sync 命令
-P 输出格式为POSIX
--sync 在取得磁盘信息前，先执行sync命令
-T 文件系统类型
```

2. file：显示文件类型

file命令对文件的检查分为文件系统、魔法幻数检查和语言检查3个过程。

- 显示单个文件类型

```
zlf@ubuntu:~/document$ file sources.list
sources.list: ASCII text
```

- 显示多个文件类型

```
zlf@ubuntu:~/document$ file *
epoll_create.txt: Microsoft Document Imaging Format
epoll_ctl.txt:    Microsoft Document Imaging Format
epoll.txt:        Microsoft Document Imaging Format
epoll_wait.txt:   Microsoft Document Imaging Format
sources.list:     ASCII text
```

3. du：报告文件所占用的磁盘空间大小

- 显示某文件占用空间

```
zlf@ubuntu:~/document$ du sources.list
4 sources.list
```

- 显示当前目录所有文件占用空间

```
zlf@ubuntu:~/document$ du -ah
16K ./epoll.txt
4.0K ./sources.list
8.0K ./epoll_wait.txt
4.0K ./epoll_create.txt
8.0K ./epoll_ctl.txt
44K .
```

- 显示占用的 iNode 个数

```
zlf@ubuntu:~/document$ du -a --inodes
1 ./epoll.txt
1 ./sources.list
1 ./epoll_wait.txt
1 ./epoll_create.txt
1 ./epoll_ctl.txt
6 .
```

4. findmnt：快速查看挂载位置和选项

5. ls：文件列表

```
-a -A ： 列出所有文件，包括隐藏文件
```

```
-d     : 显示目录，而非文件
-i     : 显示文件的 iNode 号
-l     : 显示详细信息
-r     : 排序反转
-R     : 递归显示子目录
-s     : 显示块大小
-S     : 按文件大小排序
-t     : 以修改时间排序，最近的排最前
-1     : 每行显示一个文件
```

6. tree：以树形结构列出文件，选项和 ls 类似

7. stat：查看文件详细信息

- 无参默认用法

```
zlf@ubuntu:~/document$ stat sources.list
  File: ‘sources.list’
  Size: 2582       Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d Inode: 147365      Links: 1
Access: (0777/-rwxrwxrwx)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2016-03-20 12:19:52.449451548 +0800
Modify: 2016-03-14 19:06:40.667052403 +0800
Change: 2016-03-20 09:43:55.254379318 +0800
 Birth: -
```

```
Access : 访问时间
Modify : 修改时间
Change : 属性修改时间
```

8. chmod：修改文件权限

文件或目录的访问权限分为：只读、只写、可执行三种

三种不同类型的用户可对文件或目录进行访问：文件所有者、同组用户、其他用户

因此每一文件或目录的访问权限都有三组，每组用三位表示。

权限位		
权限	数字表示	解释
r	4	读权限
w	2	写权限
x	1	执行权限
-	0	删除权限

权限范围	
权限范围	解释
u	目录或者文件的当前的用户
g	目录或者文件的当前的群组
o	当前用户或群组之外的用户或者群组
a	所有的用户及群组

权限操作方式	
操作方式	解释

+	添加权限
-	删除权限
=	设置权限

- chmod <权限范围><操作方式><权限设置> filename

```
sudo chmod a+rwx source.list
```

9. rename：批量改名

10. cp：复制文件

```
-l     : 创建硬链接文件
-r -R  : 复制目录
-s     : 创建符号链接
-u     : 更新，只有在源文件比目标文件新才copy
```

11. mv：移动或重命名文件

12. rm：删除文件

13. unlink：删除文件

14. touch：创建文件/更改文件的时间戳

15. mkdir：创建目录

- mkdir：创建新目录
- mkdir -m rwx：创建指定模式的目录
- mkdir -p src/main：递归创建目录

16. truncate：将文件缩短或扩展至指定大小

如果指定文件不存在则创建。
如果指定文件超出指定大小则超出的数据将丢失。
如果指定文件小于指定大小则用0 补足。

```
-c, --no-create        不创建文件
-o, --io-blocks        将SIZE 视为IO 块数而不使用字节数
-r, --reference=文件    使用此文件的大小
-s, --size=大小         使用此大小
```

5 文件转换

1. od：按指定进制格式打印文件

格式选项	解释
c	以 ASCII 字符形式打印
d	以无符号十进制形式打印
f	以浮点形式打印
o	以八进制形式打印
x	以十六进制形式打印

2. hd / hexdump：按指定进制格式打印文件

3. split：拆分文件

格式选项	解释

-b	以指定字节为单位划分文件
-C	以行划分文件，每行不超过指定长度
-l	以指定行数划分文件
-n	将文件划分为多少分

- 划分文件，默认后缀 x

```
zlf@ubuntu:~/document/test$ split -n 10 a.txt
zlf@ubuntu:~/document/test$ ls
a.txt  xaa  xab  xac  xad  xae  xaf  xag  xah  xai  xaj
```

- 划分文件，指定后缀

```
zlf@ubuntu:~/document/test$ split -n 10 a.txt b
zlf@ubuntu:~/document/test$ ls
a.txt  baa  bab  bac  bad  bae  baf  bag  bah  bai  baj
```

- 划分文件，指定用数字后缀

```
zlf@ubuntu:~/document/test$ split -n 10 -d  a.txt
zlf@ubuntu:~/document/test$ ls
a.txt  x00  x01  x02  x03  x04  x05  x06  x07  x08  x09
```

4. csplit：根据模式拆分文件

```
csplit a.txt /\n/ {*} -f split -b “%02d.txt"

// a.txt : 拆分 a.txt
// /\n/  : 正则表达式，按照换行符拆分
// {*}   : 表示按照模式一直拆分到文件结尾，此处也可以指定数字，表示匹配多少次
// -f split : 指定输出文件名的前缀为 split
// -b “%02d.txt” : 指定输出文件名的后缀
```

```
zlf@ubuntu:~/document/test$ ls
a.txt        split02.txt  split05.txt  split08.txt  split11.txt  split14.txt
split17.txt  split20.txt  split23.txt  split26.txt  split29.txt  split32.txt
split00.txt  split03.txt  split06.txt  split09.txt  split12.txt  split15.txt
split18.txt  split21.txt  split24.txt  split27.txt  split30.txt  split33.txt
split01.txt  split04.txt  split07.txt  split10.txt  split13.txt  split16.txt
split19.txt  split22.txt  split25.txt  split28.txt  split31.txt  split34.txt
```

5. cat：一次性显示文件内容，或将多份文件合并显示

```
cat file1 file2 ...
```

6. iconv：转换文件编码(charsets)

```
iconv -f ASCII -t UTF-32 a.txt -o b.txt
```

转换前

```
zlf@ubuntu:~/document/test$ file a.txt && ls -l
a.txt: ASCII text
total 4
-rwxrwxr-x 1 zlf zlf 2582 Mar 20 18:46 a.txt
```

转换后

```
zlf@ubuntu:~/document/test$ file b.txt && ls -l
b.txt: Unicode text, UTF-32, little-endian
total 16
-rwxrwxr-x 1 zlf zlf  2582 Mar 20 18:46 a.txt
-rw-rw-r-- 1 zlf zlf 10332 Mar 20 19:52 b.txt

// 10332/2582 = 4
```

- iconv -l： 显示有哪些编码格式

7. tar：归档压缩文件

- 归档文件

```
tar -cvf a.tar *.txt

// -c : 新建归档
// -v : 显示操作过程
// -f : 指定归档文件
```

- 显示归档文件内容

```
tar -tvf a.tar

// -t : 显示归档文件内容

less a.tar
```

- 使用 gz 压缩文件

```
tar -zcvf a.tar.gz *.txt
```

- 解压文件

```
tar -xvf a.tar.gz
```

- 解压部分文件

```
tar -xvf a.tar.gz a.txt

//单独解压 a.txt
```

- 向归档文件追加文件

```
tar -rf a.tar b.txt

// -r : 向归档文件追加文件
// 向 a.tar 归档文件追加 b.txt
```

- 向归档文件删除文件

```
tar -f a.tar --delete a.txt
```

8. zip：压缩文件/ unzip：解压缩文件

```
-d    : 从压缩文件内删除指定的文件
-g    : 将文件压缩后附加在既有的压缩文件之后，而非另行建立新的压缩文件
-m    : 将文件压缩并加入压缩文件后，删除原始文件，即把文件移到压缩文件
-u    : 更换较新的文件到压缩文件内
```

- 压缩文件

```
zip a.zip *.txt

//将目录所有 txt 文件压缩为 a.zip
```

- 递归压缩文件

```
zip -r a.zip *.txt
```

- 查看压缩文件内容

```
unzip -l a.zip
unzip -v a.zip
less a.zip
```

- 解压缩文件

```
unzip a.zip
```

9. tr：转换或删除字符

- 将集合 1 的字符替换为集合 2 的字符

```
tr a-z A-Z < a.txt

//将 a.txt 中所有小写字符替换为对应的大写字符
```

- 将集合 1 的补集字符替换为集合 2 的字符

```
tr -c a-zA-Z " " < a.txt
```

```
//将 a.txt 中所有不为  a-zA-Z0-9 的字符替换为空格
```

- 删除出现在集合 1 中的字符

```
tr -d a-zA-Z0-9 < a.txt

//删除 a.txt 中所有为 a-zA-Z0-9 的字符
```

- 去掉重复字符

```
tr -s / < a.txt

//将 a.txt 中所有重复出现的 / 字符去掉，如将 “//” 替换为 “/”
```

10. rev：以字符为单位反转文件的每一行

11. pr：将文本文件转换为可打印的文件

为文件进行分页，每页添加标题和页码等。

12. fold：为文件限定列宽，默认为 80 字符

13. col：文本过滤器，过滤一些无法显示的控制字符

6 信息检索

1. more：逐页显示文件内容

2. less：逐页显示文件内容，比 more 更强大

less 不会一下子加载整个文件，可以前后翻页，进行搜索。

- 命令行选项

```
-b <缓冲区大小> 设置缓冲区的大小
-e  当文件显示结束后，自动离开
-f  强迫打开特殊文件，例如外围设备代号、目录和二进制文件
-g  只标志最后搜索的关键词
-i  忽略搜索时的大小写
-m  显示类似more命令的百分比
-N  显示每行的行号
-o <文件名> 将less 输出的内容在指定文件中保存起来
-Q  不使用警告音
-s  显示连续空行为一行
-S  行过长时间将超出部分舍弃
-x <数字> 将“tab”键显示为规定的数字空格
```

- 控制键

```
/字符串 ：向下搜索“字符串”的功能
?字符串 ：向上搜索“字符串”的功能
n       ：重复前一个搜索（与 / 或 ? 有关）
N       ：反向重复前一个搜索（与 / 或 ? 有关）
b       ：向后翻一页
d       ：向后翻半页
h       ：显示帮助界面
Q       ：退出less 命令
u       ：向前滚动半页
y       ：向前滚动一行
空格键  ：滚动一行
回车键  ：滚动一页
[pagedown]：  向下翻动一页
[pageup]：    向上翻动一页
```

3. nl：显示文件并添加行号

4. head：显示文件头

```
-c --bytes : 显示文件前多少字节数
-n --lines : 显示文件前多少行数
```

- 显示文件前 5 行

```
head -n 5 a.txt
```

- 显示文件直至倒数 5 行

```
head -n -5 a.txt
```

5. tail：显示文件尾

```
-f –flow : 循环读取，动态显示，适合查看日志文件
```

6. wc：统计文件行数、字数、字节数

```
-c --bytes : 统计字节数
-m --chars : 统计字符数
-l --lines : 统计行数
-L --max-line-length : 最长行的长度
-w --words : 统计字数
```

7. strings：打印文件中所有的字符串

8. cut：从文件每行中选取数据

```
cut -c  file    //以字符为单位分割
cut -bn file    //以字节为单位分割，并取消分割多字节字符
cut -df file    //自定义分割符，默认为制表符
```

文件样例如下：

```
/proc/iomem

00000000-00000fff : reserved
00001000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000c0000-000c7fff : Video ROM
000e2000-000ef3ff : Adapter ROM
000f0000-000fffff : reserved
  000f0000-000fffff : System ROM
00100000-7ffeffff : System RAM
  01000000-017ce961 : Kernel code
  017ce962-01d3417f : Kernel data
```

- 截取每行第 8 个字符

```
cut -c 8 a.txt
0
0
0
0
0
0
0
0
0
9
```

- 截取每行前 8 个字符

```
cut -c -8 a.txt
00000000
00001000
0009fc00
000c0000
000e2000
000f0000
  000f00
00100000
  010000
```

```
  017ce9
```

- 截取第 8 个字符及其后面的所有字符

```
cut -c 8- a.txt
0-00000fff : reserved
0-0009fbff : System RAM
0-0009ffff : reserved
0-000c7fff : Video ROM
0-000ef3ff : Adapter ROM
0-000fffff : reserved
000-000fffff : System ROM
0-7ffeffff : System RAM
000-017ce961 : Kernel code
962-01d3417f : Kernel data
```

- 截取指定字符

```
cut -c 1,5-8,12- a.txt
00000 reserved
01000 System RAM
0fc00 reserved
00000 Video ROM
02000 Adapter ROM
00000 reserved
 0f00 : System ROM
00000 System RAM
 0000 : Kernel code
 7ce9 : Kernel data
```

- 自定义分割模式截取

```
cut -d : -f 2 a.txt
//以 : 为分割依据，选取第二段内容
 reserved
 System RAM
 reserved
 Video ROM
 Adapter ROM
 reserved
 System ROM
 System RAM
 Kernel code
 Kernel data
```

9. colrm：过滤掉输入文本中指定的列

```
cat a.txt | colrm 5
//删掉每行中第五及其以后的列

cat a.txt | colrm 3 5
//删掉每行的 3 4 5 列
```

10. locate：根据文件名查找文件

locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的数据库，之后当寻找时就只需查询这个数据库，而不必实际深入档案系统之中了。

locate 命令在搜寻数据库时比由整个由硬盘资料来搜寻资料来得快，但较差劲的是 locate 所找到的档案若是最近才建立或刚更名的，可能会找不到，在内定值中，updatedb 每天会跑一次，可以由修改 crontab来更新设定值。(etc/crontab)

11. tac：将文件反序输出，默认以行为单位

12. sort：排序文本行

```
-b    : 忽略每行前面开始出的空格字符
-c    : 检查文件是否已经按照顺序排序
-d    : 字典序
-f    : 忽略大小写字母
```

```
-i     : 只考虑可打印字符
-M     : 将前面3个字母依照月份的缩写进行排序
-n     : 依照数值的大小排序
-o<输出文件>      : 将排序后的结果存入指定的文件
-r     : 以相反的顺序来排序
-t<分隔字符>      : 指定排序时所用的栏位分隔字符
-k     : —选择以哪个区间进行排序
-R     : 随机排序
-m     : 合并已经排序好的文件
-u     : 消除重复行
```

13. uniq：报告或消除相邻重复行

14. comm：逐行比较两个排好序的文件

15. paste：将文件的行进行合并

```
paste -s a.txt
//把 a.txt 的所有行合并为一行

paste a.txt b.txt
//将 a.txt 和 b.txt 对应的行合并为一行
```

16. join：将两个文件中，指定栏位内容相同的行连接起来，类似于数据库的卡氏积

17. look：查找文件中以指定字符串开头的行

18. spell & ispell：输出文件中拼写有误的单词

19. diffstat：根据diff的比较结果，显示统计结果

```
diff a.txt b.txt | diffstat
```

20. grep：使用正则表达式搜索文本，打印匹配行

```
-n     : 显示行号
-c     : 统计匹配的行数
-f     : 指定规则文件，文件的每一行视为一条规则
-v     : 求补集，显示非匹配行之外的所有行
-R     : 递归遍历目录下所有文件
```

```
grep -R 'keyword'   .
//以当前目录为根目录，递归检索所有文件内容
```

21. find：根据文件名模式搜索文件

```
find . -name “*.txt”

//在当前目录下搜索所有后缀为 .txt 的文件
```

awk sed

7 网络管理

1. interfaces：指导如何配置 interfaces 文件(/etc/network/interfaces)

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 10.12.79.171
netmask 255.255.252.0

auto eth2
```

```
iface eth2 inet static
address 192.168.1.101
netmask 255.255.255.0
```

2. /proc/net：此目录中的文件包含了系统网络的各种状态报告

```
/proc/net/dev     -- device information
/proc/net/raw     -- raw socket information
/proc/net/tcp     -- TCP socket information
/proc/net/udp     -- UDP socket information
/proc/net/igmp    -- IGMP multicast information
/proc/net/unix    -- Unix domain socket information
/proc/net/ipx     -- IPX socket information
/proc/net/ax25    -- AX25 socket information
/proc/net/appletalk -- DDP (appletalk) socket information
/proc/net/nr      -- NET/ROM socket information
/proc/net/route   -- IP routing information
/proc/net/ax25_route -- AX25 routing information
/proc/net/imx_route -- IPX routing information
/proc/net/nr_nodes -- NET/ROM nodelist
/proc/net/nr_neigh -- NET/ROM neighbours
/proc/net/ip_masquerade -- masqueraded connections
/proc/net/snmp -- statistics
```

3. netstat：是网络监测的瑞士军刀

显示与 IP、TCP、UDP 和 ICMP 协议相关的统计数据，一般用于检验本机各端口的网络连接情况。netstat 是在内核中访问网络及相关信息的程序，它能提供 TCP 连接，TCP 和 UDP 监听，进程内存管理的相关报告。如果你的计算机有时候接收到的数据报导致出错数据或故障，你不必感到奇怪，TCP/IP 可以容许这些类型的错误，并能够自动重发数据报。但如果累计的出错情况数目占到所接收的IP数据报相当大的百分比，或者它的数目正迅速增加，那么你就应该使用 netstat 查一查为什么会出现这些情况了。

- netstat 无参数默认用法

```
zlf@ubuntu:/var/log/apt$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0     36 10.12.79.171:ssh        10.12.79.115:49157      ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    15140    /run/user..
unix  2      [ ]         DGRAM                    9560     /run/sys..
unix  2      [ ]         DGRAM                    8800     /run/systemd/notify
unix  7      [ ]         DGRAM                    8824     /run/syste..
unix  8      [ ]         DGRAM                    8832     /run/syste..
unix  2      [ ]         DGRAM                    9170     /run/syste..
unix  2      [ ]         DGRAM                    9824
unix  3      [ ]         STREAM     CONNECTED     15114    /run/syste..
```

- 列出所有套接字

```
zlf@ubuntu:/var/log/apt$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:microsoft-ds          *:*                     LISTEN
tcp        0      0 10.12.79.171:ssh        10.12.79.115:49157      ESTABLISHED
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
tcp6       0      0 [::]:http-alt           [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 localhost:8005          [::]:*                  LISTEN
```

- 列出所有 UDP 套接字

```
zlf@ubuntu:/var/log/apt$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

```
udp        0      0 *:bootpc                *:*
udp        0      0 192.168.1.25:netbios-ns *:*
udp        0      0 192.168.1.10:netbios-ns *:*
udp        0      0 10.0.2.255:netbios-ns   *:*
udp        0      0 10.0.2.15:netbios-ns    *:*
udp        0      0 10.12.79.255:netbios-ns *:*
udp        0      0 10.12.79.171:netbios-ns *:*
udp        0      0 *:netbios-ns            *:*
udp        0      0 192.168.1.2:netbios-dgm *:*
udp        0      0 192.168.1.1:netbios-dgm *:*
udp        0      0 10.0.2.255:netbios-dgm  *:*
udp        0      0 10.0.2.15:netbios-dgm   *:*
udp        0      0 10.12.79.25:netbios-dgm *:*
udp        0      0 10.12.79.17:netbios-dgm *:*
udp        0      0 *:netbios-dgm           *:*
udp        0      0 *:13975                 *:*
udp6       0      0 [::]:60314              [::]:*
```

- 列出所有 TCP 套接字

```
zlf@ubuntu:/var/log/apt$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:microsoft-ds          *:*                     LISTEN
tcp        0      0 10.12.79.171:ssh        10.12.79.115:49157      ESTABLISHED
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
tcp6       0      0 [::]:http-alt           [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 localhost:8005          [::]:*                  LISTEN
```

- 列出所有的监听套接字

```
zlf@ubuntu:/var/log/apt$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:microsoft-ds          *:*                     LISTEN
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
tcp6       0      0 [::]:http-alt           [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 localhost:8005          [::]:*                  LISTEN
```

- 列出 UDP 监听套接字：略
- 列出 TCP 监听套接字：略
- 列出 UNIX 监听套接字：略
- 显示路由信息

```
zlf@ubuntu:/var/log/apt$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         10.0.2.2        0.0.0.0         UG        0 0          0 eth0
10.0.2.0        *               255.255.255.0   U         0 0          0 eth0
10.12.76.0      *               255.255.252.0   U         0 0          0 eth1
192.168.1.0     *               255.255.255.0   U         0 0          0 eth2
```

- netstat -s：显示各种协议的统计信息
- netstat -i：显示接口信息
- 查看连接某服务端口最多的的IP地址

```
zlf@ubuntu:~$ netstat -nat | grep "192.168.1.15:22" |awk '{print $5}'|awk -F:
'{print $1}'|sort|uniq -c|sort -nr|head -20
18 221.136.168.36
```

```
3 154.74.45.242
2 78.173.31.236
2 62.183.207.98
2 192.168.1.14
2 182.48.111.215
2 124.193.219.34
2 119.145.41.2
2 114.255.41.30
1 75.102.11.99
```

- TCP 连接状态统计

```
zlf@ubuntu:~$ netstat -nat |awk '{print $6}'|sort|uniq -c
143 ESTABLISHED
1 FIN_WAIT1
1 Foreign
1 LAST_ACK
36 LISTEN
6 SYN_SENT
113 TIME_WAIT
1 established)
```

4. ifconfig：查看或配置网络接口信息(/etc/network/interfaces)

- 启动或关闭指定网卡

```
ifconfig eth0 up
ifconfig eth0 down
```

- 为网卡配置 IP 地址

```
ifconfig eth0 192.168.120.56
ifconfig eth0 192.168.120.56 netmask 255.255.255.0
ifconfig eth0 192.168.120.56 netmask 255.255.255.0 broadcast 192.168.120.255
```

- 修改网卡 MAC 地址

```
ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE
```

- 指定网卡启用/关闭 ARP 协议

```
ifconfig eth0 arp
ifconfig eth0 -arp
```

- 设置最大传输单元

```
ifconfig eth0 mtu 1500
```

5. ping：向指定主机发送 ICMP 请求报文，探测网络是否可达

- 多参数使用

```
ping -i 3 -s 1024 -t 255  www.baidu.com

//-i:指定时间间隔
//-s:指定包大小
//-t:指定 ttl 大小
```

6. mtr：动态路由追踪

- mtr 无参用法

```
mtr www.baidu.com
```

```
                             My traceroute  [v0.85]
ubuntu (0.0.0.0)                                  Sat Mar 19 13:59:42 2016
Resolver: Received error response 2. (server failure)er of fields   quit
                                  Packets               Pings
 Host                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.0.2.2                     0.0%     7    0.3   0.3   0.3   0.5   0.0
 2. ???
 3. 192.168.255.166              0.0%     7    7.4  16.3   7.4  31.4   8.7
 4. 192.168.255.210             83.3%     7   15.2  15.2  15.2  15.2   0.0
 5. 202.114.1.186               83.3%     7    6.6   6.6   6.6   6.6   0.0
 6. ???
```

字段	解释
Loss%	丢包率
Snt	sent包的数量
Last	最后一个包的延时
Avg	所有包的平均延时
Best	延时最小的包
Wrst	延时最大的包
StDev	标准偏差

- 用法举例

```
mtr -c 1 -n –report

//-n：不解析主机
//-c：发送多少个数据包
//–report：给出报告，而不是动态显示
```

7. arp：管理系统 ARP 缓存(/proc/net/arp)

- 查看系统 ARP 表项

```
zlf@ubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.78.222             ether   0c:b3:19:8d:4b:52   C                     eth1
10.12.76.46              ether   a0:86:c6:9c:e6:9d   C                     eth1
10.0.2.2                 ether   52:54:00:12:35:02   C                     eth0
10.12.77.209             ether   a4:5e:60:e3:c0:43   C                     eth1
```

- 删除一个表项

```
arp -d 10.0.2.2
```

删除后

```
zlf@ubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.78.222             ether   0c:b3:19:8d:4b:52   C                     eth1
10.12.76.46              ether   a0:86:c6:9c:e6:9d   C                     eth1
10.0.2.2                         (incomplete)                              eth0
10.12.77.209             ether   a4:5e:60:e3:c0:43   C                     eth1
```

- 配置一个表项

```
arp -s 10.0.2.2 52:54:00:12:35:02
```

配置后

```
zlf@ubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.12.79.120                     (incomplete)                              eth1
10.12.78.222             ether   0c:b3:19:8d:4b:52   C                     eth1
10.12.79.113                     (incomplete)                              eth1
10.12.76.46              ether   a0:86:c6:9c:e6:9d   C                     eth1
10.0.2.2                 ether   52:54:00:12:35:02   CM                    eth0
10.12.77.209             ether   a4:5e:60:e3:c0:43   C                     eth1
```

8. route：查看或修改 IP 路由表(/proc/net/route)

- 查看 IP 路由表

```
zlf@ubuntu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
10.0.2.0        *               255.255.255.0   U     0      0        0 eth0
10.12.76.0      *               255.255.252.0   U     0      0        0 eth1
```

```
192.168.1.0     *               255.255.255.0   U     0      0        0 eth2
```

9. **iptables：IP 信息包过滤系统，也就是防火墙（待续）**

10. **host：域名解析工具（hosts）**

- 域名解析

```
zlf@ubuntu:/proc/net$ host www.baidu.com
www.baidu.com is an alias for www.a.shifen.com
www.a.shifen.com has address 220.181.112.76
www.a.shifen.com has address 220.181.111.111
```

rdig ss iftop netogs ngrep slurm nslookup iptables ifup named hostm

7 项目管理

cloc git

8 常用工具

gcc gdb objdump make 数学计算 ssh watch maven jq openssl pandoc diff patch idconfig readelf pstack ngrep indent tempfile expr test protoize nm unprotoize gcov as info mesg md5sum sum bc cksum fgrep bzcmp expand unexpand ispell slocate cmp colrm ln arch echo groff nroff as ld gprof c++filt nm readelf strip seq GNU indent GNU cflow nm ldd snoop swap pagesize dbx