

目录

- 1. Linux启动过程
- 2. Linux 文件系统结构
- 3. Linux 的目录结构是怎样的？
- 4. Linux IO
- 5. 文件管理命令
 - cat 命令
 - chmod 命令
 - chown 命令
 - cp&find
 - more 命令
 - less 命令
 - locate 命令
 - mv&rm
 - tail 命令
 - touch 命令
 - whereis 命令
 - which 命令
- 文档编辑命令
 - grep 命令
 - awk 命令
 - sed 命令
- 网络通讯命令
 - netstat 命令
- 系统管理命令
 - iostat 命令
 - top 命令
 - ps 命令
- 磁盘命令
 - fdisk 命令
 - df 命令

1. Linux启动过程

Linux 开机启动过程？

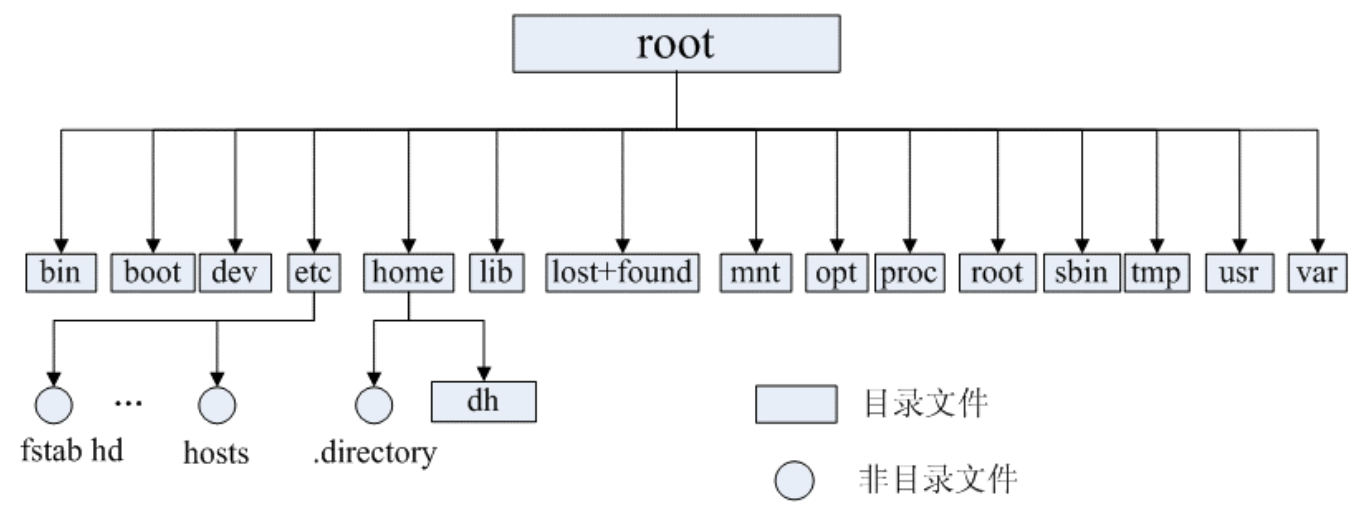
1. 主机加电自检，加载 BIOS 硬件信息。
2. 读取 MBR 的引导文件(GRUB、LILO)。
3. 引导 Linux 内核。
4. 运行第一个进程 init (进程号永远为 1)。
5. 进入相应的运行级别。
6. 运行终端，输入用户名和密码。

2. Linux 文件系统结构

在 Linux 操作系统中，所有被操作系统管理的资源，例如网络接口卡、磁盘驱动器、打印机、输入输出设备、普通文件或是目录都被看作是一个文件。

也就是说在 Linux 系统中有一个重要的概念：**一切都是文件**。其实这是 Unix 哲学的一个体现，而 Linux 是重写 Unix 而来，所以这个概念也就传承了下来。在 Unix 系统中，把一切资源都看作是文件，包括硬件设备。UNIX 系统把每个硬件都看成是一个文件，通常称为设备文件，这样用户就可以用读写文件的方式实现对硬件的访问。

3. Linux 的目录结构是怎样的？



常见目录说明：

- /bin：存放二进制可执行文件(ls,cat,mkdir等)，常用命令一般都在这里；
- /etc：存放系统管理和配置文件；
- /home：存放所有用户文件的根目录，是用户主目录的基点，比如用户user的主目录就是/home/user，可以用~user表示；
- /usr：用于存放系统应用程序；
- /opt：额外安装的可选应用程序包所放置的位置。一般情况下，我们可以把tomcat等都安装到这里；
- /proc：虚拟文件系统目录，是系统内存的映射。可直接访问这个目录来获取系统信息；
- /root：超级用户（系统管理员）的主目录（特权阶级o）；
- /sbin: 存放二进制可执行文件，只有root才能访问。这里存放的是系统管理员使用的系统级别的管理命令和程序。如ifconfig等；
- /dev：用于存放设备文件；
- /mnt：系统管理员安装临时文件系统的安装点，系统提供这个目录是让用户临时挂载其他的文件系统；
- /boot：存放用于系统引导时使用的各种文件；
- /lib：存放着和系统运行相关的库文件；
- /tmp：用于存放各种临时文件，是公用的临时文件存储点；

/var：用于存放运行时需要改变数据的文件，也是某些大文件的溢出区，比方说各种服务的日志文件（系统启动日志等。）等；

/lost+found：这个目录平时是空的，系统非正常关机而留下“无家可归”的文件（windows下叫什么.chk）就在这里。

4. Linux IO

在了解不同的IO之前先了解：同步与异步，阻塞与非阻塞的区别 同步，一个任务的完成之前不能做其他操作，必须等待（等于在打电话） 异步，一个任务的完成之前，可以进行其他操作（等于在聊QQ） 阻塞，是相对于CPU来说的，挂起当前线程，不能做其他操作只能等待 非阻塞，无须挂起当前线程，可以去执行其他操作

- 什么是BIO

BIO：同步并阻塞，服务器实现一个连接一个线程，即客户端有连接请求时服务器端就需要启动一个线程进行处理，没处理完之前此线程不能做其他操作（如果是单线程的情况下，我传输的文件很大呢？），当然可以通过线程池机制改善。BIO方式适用于连接数目比较小且固定的架构，这种方式对服务器资源要求比较高，并发局限于应用中，JDK1.4以前的唯一选择，但程序直观简单易理解。

- 什么是NIO

NIO:同步非阻塞，服务器实现一个连接一个线程，即客户端发送的连接请求都会注册到多路复用器上，多路复用器轮询到连接有I/O请求时才启动一个线程进行处理。NIO方式适用于连接数目多且连接比较短（轻操作）的架构，比如聊天服务器，并发局限于应用中，编程比较复杂，JDK1.4之后开始支持。

- 什么是AIO

AIO：异步非阻塞，服务器实现模式为一个有效请求一个线程，客户端的I/O请求都是由操作系统先完成了再通知服务器应用去启动线程进行处理，AIO方式使用于连接数目多且连接比较长（重操作）的架构，比如相册服务器，充分调用操作系统参与并发操作，编程比较复杂，JDK1.7之后开始支持。

AIO属于NIO包中的类实现，其实IO主要分为BIO和NIO，AIO只是附加品，解决IO不能异步的实现。

在以前很少有Linux系统支持AIO，Windows的IOCP就是该AIO模型。但是现在的服务器一般都是支持AIO操作。

5. 文件管理命令

cat 命令

cat 命令用于连接文件并打印到标准输出设备上。

功能：

1. 一次显示整个文件:

```
cat filename
```

2. 从键盘创建一个文件:

```
cat > filename //只能创建新文件，不能编辑已有文件。
```

3. 将几个文件合并为一个文件:

```
// -b 对非空输出行号  
// -n 输出所有行号  
cat file1 file2 > file
```

chmod 命令

每一文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写和执行权限；与属主同组的用户的读、写和执行权限；系统中其他用户的读、写和执行权限。可使用 `ls -l test.txt` 查找。

以文件 `log2012.log` 为例：

```
-rw-r--r-- 1 root root 296K 11-13 06:03 log2012.log
```

第一列共有 10 个位置，第一个字符指定了文件类型。在通常意义上，一个目录也是一个文件。如果第一个字符是横线，表示是一个非目录的文件。如果是 `d`，表示是一个目录。从第二个字符开始到第十个 9 个字符，3 个字符一组，分别表示了 3 组用户对文件或者目录的权限。权限字符用横线代表空许可，`r` 代表只读，`w` 代表写，`x` 代表可执行。

常用参数：

```
-c 当发生改变时，报告处理信息  
-R 处理指定目录以及其子目录下所有文件
```

权限范围：

```
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 : 除了目录或者文件的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
```

权限代号：

```
r : 读权限，用数字4表示  
w : 写权限，用数字2表示  
x : 执行权限，用数字1表示  
- : 删除权限，用数字0表示  
s : 特殊权限
```

举例：

1. 增加文件 t.log 所有用户可执行权限

```
chmod a+x t.log
```

2. 撤销原来所有的权限，然后使拥有者具有可读权限,并输出处理信息

```
chmod u=r t.log -c
```

3. 给 file 的属主分配读、写、执行(7)的权限，给file的所在组分配读、执行(5)的权限，给其他用户分配执行(1)的权限

```
chmod 751 t.log -c ( 或者 : chmod u=rwx,g=rx,o=x t.log -c)
```

4. 将 test 目录及其子目录所有文件添加可读权限

```
chmod u+r,g+r,o+r -R test/ -c
```

chown 命令

chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户 ID；组可以是组名或者组 ID；文件是以空格分开的要改变权限的文件列表，支持通配符。

常用参数：

```
-c 当发生改变时，报告处理信息  
-R 处理指定目录以及其子目录下所有文件
```

实例：

1. 改变拥有者和群组 并显示改变信息

```
chown -c mail:mail log2012.log
```

2. 改变文件群组

```
chown -c :mail t.log
```

3. 改变文件夹及子文件目录属主及属组为 mail

```
chown -cR mail: test/
```

cp&find

1. 复制 a.txt 到 test 目录下，保持原文件时间，如果原文件存在提示是否覆盖。

```
cp -ai a.txt test
```

2. 为 a.txt 建议一个链接（快捷方式）

```
cp -s a.txt link_a.txt
```

3. 查找 48 小时内修改过的文件

```
find -atime -2
```

4. 在当前目录查找以 .log 结尾的文件。 . 代表当前目录

```
find ./ -name '*.log'
```

5. 查找 /opt 目录下 权限为 777 的文件

```
find /opt -perm 777
```

6. 查找大于 1K 的文件

```
find -size +1000c
```

more 命令

功能类似于 cat, more 会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下页显示，按 b 键就会往回（back）一页显示。

1. 显示文件中从第3行起的内容

```
more +3 text.txt
```

2. 在所列文件目录详细信息，借助管道使每次显示 5 行

```
ls -l | more -5
```

less 命令

`less` 与 `more` 类似，但使用 `less` 可以随意浏览文件，而 `more` 仅能向前移动，却不能向后移动，而且 `less` 在查看之前不会加载整个文件。

1. `ps` 查看进程信息并通过 `less` 分页显示

```
ps -aux | less -N
```

2. 查看多个文件

```
less 1.log 2.log
```

locate 命令

`locate` 通过搜寻系统内建文档数据库达到快速找到档案，数据库由 `updatedb` 程序来更新，`updatedb` 是由 `cron daemon` 周期性调用的。默认情况下 `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`)。

`locate` 与 `find` 命令相似，可以使用如 `*`、`?` 等进行正则匹配查找

常用参数：

```
-l num ( 要显示的行数 )  
-f    将特定的档案系统排除在外，如将 proc 排除在外  
-r    使用正则运算式做为寻找条件
```

实例：

1. 查找和 `pwd` 相关的所有文件(文件名中包含 `pwd`)

```
locate pwd
```

2. 搜索 etc 目录下所有以 sh 开头的文件

```
locate /etc/sh
```

3. 查找 /var 目录下，以 reason 结尾的文件

```
locate -r '^/var.*reason$' ( 其中.表示一个字符，*表示任务多个；.*表示任意多个字符 )
```

mv&rm

tail 命令

用于显示指定文件末尾内容，不指定文件时，作为输入信息进行处理。常用查看日志文件。

常用参数：

```
-f 循环读取 ( 常用于查看递增的日志文件 )  
-n<行数> 显示行数 ( 从后向前 )
```

例子：循环读取逐渐增加的文件内容

```
ping 127.0.0.1 > ping.log &
```

后台运行：可使用 jobs -l 查看，也可使用 fg 将其移到前台运行。

```
tail -f ping.log
```

touch 命令

Linux touch命令用于修改文件或者目录的时间属性，包括存取时间和更改时间。若文件不存在，系统会建立一个新的文件。

ls -l 可以显示档案的时间记录。

1. 使用指令"touch"修改文件"testfile"的时间属性为当前系统时间，输入如下命令：

```
touch testfile           //修改文件的时间属性
```

2. 使用ls命令查看testfile文件的属性:


```
ls -l testfile          //查看文件的时间属性
//原来文件的修改时间为16:09
-rw-r--r-- 1 hdd hdd 55 2011-08-22 16:09 testfile
```

3. 执行指令"touch"修改文件属性以后，并再次查看该文件的时间属性，如下所示：

```
touch testfile          //修改文件时间属性为当前系统时间
$ ls -l testfile        //查看文件的时间属性
//修改后文件的时间属性为当前系统时间
-rw-r--r-- 1 hdd hdd 55 2011-08-22 19:53 testfile
```

whereis 命令

whereis 命令只能用于程序名的搜索，而且只搜索二进制文件（参数-b）、man说明文件（参数-m）和源代码文件（参数-s）。如果省略参数，则返回所有信息。whereis 及 locate 都是基于系统内建的数据库进行搜索，因此效率很高，而find则是遍历硬盘查找文件。

1. 查找 locate 程序相关文件

```
whereis locate
```

2. 查找 locate 的源码文件

```
whereis -s locate
```

3. 查找 locate 的帮助文件

```
whereis -m locate
```

which 命令

在 linux 要查找某个文件，但不知道放在哪里了，可以使用下面的一些命令来搜索：

which	查看可执行文件的位置。
whereis	查看文件的位置。
locate	配合数据库查看文件位置。
find	实际搜寻硬盘查询文件名称。

文档编辑命令

grep 命令

强大的文本搜索命令，grep(Global Regular Expression Print) 全局正则表达式搜索。

grep 的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到标准输出，不影响原文件内容。

常用参数：

```
-A n --after-context 显示匹配字符后n行  
-B n --before-context 显示匹配字符前n行  
-C n --context 显示匹配字符前后n行  
-c --count 计算符合样式的列数  
-i 忽略大小写  
-l 只列出文件内容符合指定的样式的文件名称  
-f 从文件中读取关键词  
-n 显示匹配内容的所在文件中行数  
-R 递归查找文件夹
```

1. 查找指定进程

```
ps -ef | grep svn
```

2. 查找指定进程个数

```
ps -ef | grep svn -c
```

3. 从文件中读取关键词

```
cat test1.txt | grep -f key.log
```

4. 从文件夹中递归查找以grep开头的行，并只列出文件

```
grep -lR '^grep' /tmp
```

5. 查找非x开关的行内容

```
grep '^[^x]' test.txt
```

6. 显示包含 ed 或者 at 字符的内容行

```
grep -E 'ed|at' test.txt
```

awk 命令

简单来说awk就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

sed 命令

Linux sed命令是利用script来处理文本文件。sed可依照script的指令，来处理、编辑文本文件。Sed主要用来自动编辑一个或多个文件；简化对文件的反复操作；编写转换程序等。

常用参数：

```
-e< script> 或-expression=< script> 以选项中指定的script来处理输入的文本文件。  
-f< script文件>或-file=< script文件> 以选项中指定的script文件来处理输入的文本文件。  
-h或-help 显示帮助。  
-n或-quiet或-silent 仅显示script处理后的结果。  
-V或-version 显示版本信息。
```

动作说明：

```
a : 新增， a 的后面可以接字符串，而这些字符串会在新的一行出现(目前的下一行)  
c : 取代， c 的后面可以接字符串，这些字符串可以取代 n1,n2 之间的行！  
d : 删除，因为是删除啊，所以 d 后面通常不接任何咚咚；  
i : 插入， i 的后面可以接字符串，而这些字符串会在新的一行出现(目前的上一行)；  
p : 打印，亦即将某个选择的数据印出。通常 p 会与参数 sed -n 一起运行  
s : 取代，可以直接进行取代的工作哩！通常这个 s 的动作可以搭配正规表示法！例如  
1,20s/old/new/g
```

例子：在testfile文件的第四行后添加一行，并将结果输出到标准输出，在命令行提示符下输入如下命令：

```
sed -e 4a\newLine testfile
```

首先查看testfile中的内容如下：

```
$ cat testfile #查看testfile 中的内容  
HELLO LINUX!  
Linux is a free unix-type opterating system.  
This is a linux testfile!  
Linux test
```

使用sed命令后，输出结果如下:

```
$ sed -e 4a\newline testfile #使用sed 在第四行后添加新字符串
HELLO LINUX! #testfile文件原有的内容
Linux is a free unix-type operating system.
This is a linux testfile!
Linux test
newline
```

awk、sed、grep对比 grep 更适合单纯的查找或匹配文本 sed 更适合编辑匹配到的文本 awk 更适合格式化文本，对文本进行较复杂格式处理

网络通讯命令

netstat 命令

参数说明：

- a或-all 显示所有连线中的Socket。
- A<网络类型>或-<网络类型> 列出该网络类型连线中的相关地址。
- c或-continuous 持续列出网络状态。
- C或-cache 显示路由器配置的快取信息。
- e或-extend 显示网络其他相关信息。
- F或-fib 显示FIB。
- g或-groups 显示多重广播功能群组组员名单。
- h或-help 在线帮助。
- i或-interfaces 显示网络界面信息表单。
- l或-listening 显示监控中的服务器的Socket。
- M或-masquerade 显示伪装的网络连线。
- n或-numeric 直接使用IP地址，而不通过域名服务器。
- N或-netlink或-symbolic 显示网络硬件外围设备的符号连接名称。
- o或-timers 显示计时器。
- p或-programs 显示正在使用Socket的程序识别码和程序名称。
- r或-route 显示Routing Table。
- s或-statistic 显示网络工作信息统计表。
- t或-tcp 显示TCP传输协议的连线状况。
- u或-udp 显示UDP传输协议的连线状况。
- v或-verbose 显示指令执行过程。
- V或-version 显示版本信息。
- w或-raw 显示RAW传输协议的连线状况。
- x或-unix 此参数的效果和指定"-A unix"参数相同。
- ip或-inet 此参数的效果和指定"-A inet"参数相同。

1. 如何查看系统都开启了哪些端口？

```
netstat -lnp
```

2. 如何查看网络连接状况？

```
netstat -an
```

3. 如何统计系统当前进程连接数？

```
netstat -an | grep ESTABLISHED | wc -l
```

4. 用 netstat 命令配合其他命令，按照源 IP 统计所有到 80 端口的 ESTABLISHED 状态链接的个数？

```
netstat -an|grep ESTABLISHED
```

系统管理命令

iostat 命令

iostat命令被用于监视系统输入输出设备和CPU的使用情况。它的特点是汇报磁盘活动统计情况，同时也会汇报出CPU使用情况。同vmstat一样，iostat也有一个弱点，就是它不能对某个进程进行深入分析，仅对系统的整体情况进行分析。

常用参数：

- c：仅显示CPU使用情况；
- d：仅显示设备利用率；
- k：显示状态以千字节每秒为单位，而不使用块每秒；
- m：显示状态以兆字节每秒为单位；
- p：仅显示块设备和所有被使用的其他分区的状态；
- t：显示每个报告产生时的时间；
- v：显示版本号并退出；
- x：显示扩展状态。

1. 用iostat -x /dev/sda1来观看磁盘I/O的详细情况：

```
iostat -x /dev/sda1
```

top 命令

显示当前系统正在执行的进程的相关信息，包括进程 ID、内存占用率、CPU 占用率等。

常用参数：

- c 显示完整的进程命令
- s 保密模式

- p <进程号> 指定进程显示
- n <次数>循环显示次数

例子：

```
top - 14:06:23 up 70 days, 16:44, 2 users, load average: 1.25, 1.32, 1.35
Tasks: 206 total, 1 running, 205 sleeping, 0 stopped, 0 zombie
Cpu(s): 5.9%us, 3.4%sy, 0.0%ni, 90.4%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 32949016k total, 14411180k used, 18537836k free, 169884k buffers
Swap: 32764556k total, 0k used, 32764556k free, 3612636k cached
PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
28894 root        22   0 1501m 405m 10m  S 52.2  1.3   2534:16 java
```

- 第一行，任务队列信息，同 uptime 命令的执行结果。
- 第二行，Tasks — 任务（进程），具体信息说明如下：

系统现在共有206个进程，其中处于运行中的有1个，205个在休眠（sleep），stoped状态的有0个，zombie状态（僵尸）的有0个。

- 第三行，cpu状态信息，具体属性说明如下：

```
5.9%us - 用户空间占用CPU的百分比。
3.4% sy - 内核空间占用CPU的百分比。
0.0% ni - 改变过优先级的进程占用CPU的百分比
90.4% id - 空闲CPU百分比
0.0% wa - IO等待占用CPU的百分比
0.0% hi - 硬中断（Hardware IRQ）占用CPU的百分比
0.2% si - 软中断（Software Interrupts）占用CPU的百分比
```

- 第四行，内存状态，具体信息如下：

```
32949016k total - 物理内存总量（32GB）
14411180k used - 使用中的内存总量（14GB）
18537836k free - 空闲内存总量（18GB）
169884k buffers - 缓存的内存量（169M）
```

- 第五行，swap交换分区信息，具体信息说明如下：

```
32764556k total - 交换区总量（32GB）
0k used - 使用的交换区总量（0K）
32764556k free - 空闲交换区总量（32GB）
3612636k cached - 缓冲的交换区总量（3.6GB）
```

- 第六行，空行。

- 第七行以下：各进程（任务）的状态监控，项目列信息说明如下：

```
PID - 进程id
USER - 进程所有者
PR - 进程优先级
NI - nice值。负值表示高优先级，正值表示低优先级
VIRT - 进程使用的虚拟内存总量，单位kb。VIRT=SWAP+RES
RES - 进程使用的、未被换出的物理内存大小，单位kb。RES=CODE+DATA
SHR - 共享内存大小，单位kb
S - 进程状态。D=不可中断的睡眠状态 R=运行 S=睡眠 T=跟踪/停止 Z=僵尸进程
%CPU - 上次更新到现在的CPU时间占用百分比
%MEM - 进程使用的物理内存百分比
TIME+ - 进程使用的CPU时间总计，单位1/100秒
COMMAND - 进程名称（命令名/命令行）
```

ps 命令

ps(process status)，用来查看当前运行的进程状态，一次性查看，如果需要动态连续结果使用 top

linux上进程有5种状态:

运行(正在运行或在运行队列中等待)

中断(休眠中, 受阻, 在等待某个条件的形成或接受到信号)

不可中断(收到信号不唤醒和不可运行, 进程必须等待直到有中断发生)

僵死(进程已终止, 但进程描述符存在, 直到父进程调用wait4()系统调用后释放)

停止(进程收到SIGSTOP, SIGSTP, SIGTIN, SIGTOU信号后停止运行运行)

ps 工具标识进程的5种状态码:

```
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
```

常用参数：

```
-A 显示所有进程
a 显示所有进程
-a 显示同一终端下所有进程
c 显示进程真实名称
e 显示环境变量
f 显示进程间的关系
r 显示当前终端运行的进程
-aux 显示所有包含其它使用的进程
```

实例：

1. 显示当前所有进程环境变量及进程间关系

```
ps -ef
```

2. 显示当前所有进程

```
ps -A
```

3. 与grep联用查找某进程

```
ps -aux | grep apache
```

4. 找出与 cron 与 syslog 这两个服务有关的 PID 号码

```
ps aux | grep '(cron|syslog)'
```

磁盘命令

fdisk 命令

fdisk命令用于观察硬盘实体使用情况，也可对硬盘分区。它采用传统的问答式界面，而非类似DOS fdisk的cfdisk互动式操作界面，因此在使用上较为不便，但功能却丝毫不打折扣。

常用参数：

```
-b<分区大小>：指定每个分区的大小；  
-l：列出指定的外围设备的分区表状况；  
-s<分区编号>：将指定的分区大小输出到标准输出上，单位为区块；  
-u：搭配"-l"参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址；  
-v：显示版本信息。
```

直接：

```
fdisk /dev/sdb
```

df 命令

df命令用于显示磁盘分区上的可使用的磁盘空间。默认显示单位为KB。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

```
-a或--all：包含全部的文件系统；
--block-size=<区块大小>：以指定的区块大小来显示区块数目；
-h或--human-readable：以可读性较高的方式来显示信息；
-H或--si：与-h参数相同，但在计算时是以1000 Bytes为换算单位而非1024 Bytes；
-i或--inodes：显示inode的信息；
-k或--kilobytes：指定区块大小为1024字节；
-l或--local：仅显示本地端的文件系统；
-m或--megabytes：指定区块大小为1048576字节；
--no-sync：在取得磁盘使用信息前，不要执行sync指令，此为预设值；
-P或--portability：使用POSIX的输出格式；
--sync：在取得磁盘使用信息前，先执行sync指令；
-t<文件系统类型>或--type=<文件系统类型>：仅显示指定文件系统类型的磁盘信息；
-T或--print-type：显示文件系统的类型；
-x<文件系统类型>或--exclude-type=<文件系统类型>：不要显示指定文件系统类型的磁盘信息；
--help：显示帮助；
--version：显示版本信息。
```

直接：

```
df
```