1. fdisk命令 对硬盘及分区的操作

fdisk –l 主分区 扩展分区 逻辑分区

df –h 或 df //查看当前文件系统分区及挂载

1. mkfs.ext4 /dev/sdb //格式化,sdb为新增加的硬盘

mount /dev/sdb /media //挂载

1. lsblk命令可查看块设备和文件系统信息:
2. -a, --all 显示所有设备
3. -b, --bytes 以bytes方式显示设备大小
4. -d, --nodeps 不显示 slaves 或 holders
5. -D, --discard print discard capabilities
6. -e, --exclude <list> 排除设备 (default: RAM disks)
7. -f, --fs 显示文件系统信息
8. -h, --help 显示帮助信息
9. -i, --ascii use ascii characters only
10. -m, --perms 显示权限信息
11. -l, --list 使用列表格式显示
12. -n, --noheadings 不显示标题
13. -o, --output <list> 输出列
14. -P, --pairs 使用key="value"格式显示
15. -r, --raw 使用原始格式显示
16. -t, --topology 显示拓扑结构信息
17. 查看内核版本命令: #cat /proc/version #uname -a
18. dd

使用dd这个linux命令可以创建一定大小文件。

linux创建文件命令：dd命令

把指定的输入文件拷贝到指定的输出文件中，并且在拷贝的过程中可以进行格式转换。语法：

dd 〔选项〕

if =输入文件(或设备名称)。

of =输出文件(或设备名称)。

ibs = bytes 一次读取bytes字节，即读入缓冲区的字节数。

skip = blocks 跳过读入缓冲区开头的ibs\*blocks块。

obs = bytes 一次写入bytes字节，即写 入缓冲区的字节数。

bs = bytes 同时设置读/写缓冲区的字节数(等于设置obs和obs)。

cbs = bytes 一次转换bytes字节。

count = blocks 只拷贝输入的blocks块。

conv = ASCII 把EBCDIC码转换为ASCII码。

conv = ebcdic 把ASCII码转换为EBCDIC码。

conv = ibm 把ASCII码转换为alternate EBCDIC码。

conv = blick 把变动位转换成固定字符。

conv = ublock 把固定位转换成变动位

conv = ucase 把字母由小写变为大写。

conv = lcase 把字母由大写变为小写。

conv = notrunc 不截短输出文件。

conv = swab 交换每一对输入字节。

conv = noerror 出错时不停止处理。

conv = sync 把每个输入记录的大小都调到ibs的大小(用ibs填充)。

fdformat命令

低级格式化软盘。

实例:

将/dev/hdx全盘数据备份到指定路径的image文件：dd if=/dev/hdx of=/path/to/image

创建一个100M的空文件

dd if=/dev/zero of=hello.txt bs=100M count=1

1. linux下查看动态库和静态库

静态库用ar -t YourFile

动态库用 nm -D YourFile

ar命令可以用来创建、修改库，也可以从库中提出单个模块。

<http://blog.csdn.net/syh63053767/article/details/14128373>

1. Linux动态库文件搜索路径

1)、环境变量LD\_LIBRARY\_PATH 指定的动态库搜索路径

2）、/etc/ld.so.cache中所缓存的动态库路径，这个可以通过先修改配置文件/etc/ld.so.conf中指定的动态库搜索路径，然后执行ldconfig命令来改变。

3）、默认的动态库搜索路径/lib

4）、默认的动态库搜索路径/usr/lib

5）、ELF可执行文件中动态段中DT\_RPATH所指定的路径。即在编译目标代码时, 对gcc加入链接参数“-Wl,-rpath”指定动态库搜索路径，eg：gcc -Wl,-rpath,/home/arc/test,-rpath,/lib/,-rpath,/usr/lib/,-rpath,/usr/local/lib test.c

嵌入式中2）和5）应用较多。

1. **关于环境变量命令介绍**

1.echo 显示某个环境变量值 echo $PATH

2.export 设置一个新的环境变量 export HELLO="hello" (可以无引号)  
3.env 显示所有环境变量  
4.set 显示本地定义的shell变量  
5.unset 清除环境变量 unset HELLO  
6.readonly 设置只读环境变量 readonly HELLO  
**常见的环境变量**  
PATH：决定了shell将到哪些目录中寻找命令或程序  
HOME：当前用户主目录  
MAIL：是指当前用户的邮件存放目录。  
SHELL：是指当前用户用的是哪种Shell。  
HISTSIZE：是指保存历史命令记录的条数  
LOGNAME：是指当前用户的登录名。

HOSTNAME：是指主机的名称，许多应用程序如果要用到主机名的话，通常是从这个环

境变量中来取得的。

LANG/LANGUGE：是和语言相关的环境变量，使用多种语言的用户可以修改此环境变量。  
PS1：是基本提示符，对于root用户是#，对于普通用户是$。  
PS2：是附属提示符，默认是“>”。可以通过修改此环境变量来修改当前的命令符，比如下列命令会将提示符修改成字符串“Hello,My NewPrompt :) ”。  
# PS1=" Hello,My NewPrompt :) "

1. file命令

file命令用来探测给定文件的类型。  
-f<名称文件>：指定名称文件，其内容有一个或多个文件名称时，让file依序辨识这些文件，格式为每列一个文件名称；

1. gpasswd命令 功能：管理组

用法：gpasswd[-a user][-d user][-A user,...][-M user,...][-r][-R]groupname

-a：添加用户到组

-d：从组删除用户

-A：指定管理员

-M：指定组成员和-A的用途差不多

-r：删除密码

1. -R：限制用户登入组，只有组中的成员才可以用newgrp加入该组
2. strings 命令

在对象文件或二进制文件中查找可打印的字符串。

退出状态：0 表示命令成功运行。>0 表示出错。

如：strings /lib/x86\_64-linux-gnu/libc.so.6 |grep GLIBC\_

strings /lib/i386-linux-gnu/libc.so.6 |grep GLIBC\_

1. objdump命令 用查看目标文件或者可执行的目标文件的构成的gcc工具

-p 查看库文件版本等信息 如：objdump -p libmininas.so

-a 显示档案库的成员信息,类似ls -l将lib\*.a的信息列出。

-T 显示文件的动态符号表入口，仅仅对动态目标文件有意义，比如某些

共享库。它显示的信息类似于 nm -D|--dynamic 显示的信息。

如：objdump -T app |grep GLIBC\_2.14

1. linux查看硬件信息命令
2. 查看硬盘和分区分布 lsblk
3. 查看硬盘和分区的详细信息 fdisk –l
4. 查看网卡硬件信息 lspci | grep -I ‘eth’
5. 查看系统的所有网络接品 ifconfig –a
6. 查看某个网络接口的详细信息，例如eth0的详细参数和指标 ethtool eth0
7. 查看内存情况 free -m
8. 查看内存详细使用 cat /proc/meminfo
9. 查看cpu的统计信息 lscpu
10. 查看/proc/cpuinfo，查看每个cpu信息，如每个cpu的型号，主频等 cat /proc/cpuinfo
11. 系统的glibc版本太低

1)、到http://www.gnu.org/software/libc/下载最新版本，我这里下载了glibc-2.14.tar.xz 这个版本，解压到任意目录准备编译

2)、在glibc源码目录建立构建目录，并cd进入构建目录

$ mkdir build $ cd build

3）、运行configure配置，make && sudo make install

$ ../configure --prefix=/opt/glibc-2.14

$ make -j4  //启用4个cpu去编译。提高编译速度

$ sudo make install

4）临时修改环境变量

export LD\_LIBRARY\_PATH=/opt/glibc-2.14/lib:$LD\_LIBRARY\_PATH

configure --prefix=/ 指定安装路径, 不指定prefix，则可执行文件默认放在/usr /local/bin，库文件默认放在/usr/local/lib，配置文件默认放在/usr/local/etc。

1. 清除linux缓存命令

查看内存情况 # more /proc/meminfo

1)、#sync

2)、#echo 3 > /proc/sys/vm/drop\_caches

To free pagecache:

# echo 1 > /proc/sys/vm/drop\_caches

To free dentries and inodes:

# echo 2 > /proc/sys/vm/drop\_caches

To free pagecache, dentries and inodes:

echo 3 > /proc/sys/vm/drop\_caches

1. Linux系统挂载硬件（root）

1）、lsblk

2）、mkfs.ext4 /dev/sdb

3）、mount /dev /sdb /mnt/mininas

1. df命令

linux中df命令的功能是用来检查linux服务器的文件系统的磁盘空间占用情况。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。

df [-ahikHTm] [目录或文件名]

选项与参数：

-a ：列出所有的文件系统，包括系统特有的 /proc 等文件系统；

-k ：以 KBytes 的容量显示各文件系统；

-m ：以 MBytes 的容量显示各文件系统；

-h ：以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；

-H ：以 M=1000K 取代 M=1024K 的进位方式；

-T ：显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出；

-i ：不用硬盘容量，而以 inode 的数量来显示

例：

1）、df –h //将容量结果以易读的容量格式显示出来

2）、df –aT //将系统内的所有特殊文件格式及名称都列出来

3）、df -h /etc //将 /etc 底下的可用的磁盘容量以易读的容量格式显示

1. du命令

inux du命令也是查看使用空间的，但是与df命令不同的是Linux du命令是对文件和目录磁盘使用的空间的查看

du [-ahskm] 文件或目录名称

选项与参数：

-a ：列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。

-h ：以人们较易读的容量格式 (G/M) 显示；

-s ：列出总量而已，而不列出每个各别的目录占用容量；

-S ：不包括子目录下的总计，与 -s 有点差别。

-k ：以 KBytes 列出容量显示；

-m ：以 MBytes 列出容量显示

1）、du –sm /\* //检查根目录底下每个目录所占用的容量

2）、du –a //将文件的容量也列出来

1. su 命令

功能说明：切换用户身份。

语　　法：su [-flmp][--help][--version][-][-c <指令>][-s <shell>][用户帐号]

补充说明：su可让用户暂时变更登入的身份。变更时须输入所要变更的用户帐号与密码。

参　　数：

　-c<指令>或--command=<指令> 　执行完指定的指令后，即恢复原来的身份。

　-f或--fast 　适用于csh与tsch，使shell不用去读取启动文件。

　-l或--login 　改变身份时，也同时变更工作目录，以及HOME,SHELL,USER,LOGNAME。此外，也会变更PATH变量。

　-m,-p或--preserve-environment 　变更身份时，不要变更环境变量。

　-s<shell>或--shell=<shell> 　指定要执行的shell。

　--help 　显示帮助。

　--version 　显示版本信息。

su命令和su -命令最大的本质区别就是：前者只是切换了root身份，但Shell环境仍然是普通用户的Shell；而后者连用户和Shell环境一起切换成root身份了。

　[用户帐号] 指定要变更的用户。若不指定此参数，则预设变更为root。

如：

　　su -c ls root 变更帐号为 root 并在执行 ls 指令后退出变回原使用者。

　　su root -f 变更帐号为 root 并传入 -f 参数给新执行的 shell。

　　su - test 变更帐号为 test 并改变工作目录至 test 的家目录（home dir）。

1. 动态库libxxxx.so

1、在Makefile中添加以下代码

so:

gcc -shared -fPIC -o libuserinfo.so ./src/client\_info.c ./src/win\_mysql.c ./src/cJSON.c -I ./header -lmysqlclient –lm

执行make so

2、将生成的libxxxx.so移到相应的目录，如./usr/local/lib中

3、ldconfig让动态链接库为系统所共享

1. Linux动态库文件搜索路径

1)、ELF可执行文件中动态段中DT\_RPATH所指定的路径。即在编译目标代码时, 对gcc加入链接参数“-Wl,-rpath”指定动态库搜索路径，eg：gcc -Wl,-rpath,/home/arc/test,-rpath,/lib/,-rpath,/usr/lib/,-rpath,/usr/local/lib test.c

2)、环境变量LD\_LIBRARY\_PATH 指定的动态库搜索路径

3)、/etc/ld.so.cache中所缓存的动态库路径，这个可以通过先修改配置文件/etc/ld.so.conf中指定的动态库搜索路径，然后执行ldconfig命令来改变。

4)、默认的动态库搜索路径/lib

5)、默认的动态库搜索路径/usr/lib

1. version `GLIBC\_2.14' not found 解决方法

将 glibc 2.14 版本memcpy降到glibc 2.2.5. 在源代码写入以下代码

\_\_asm\_\_(".symver memcpy,memcpy@GLIBC\_2.2.5");

1. 修改时区

　　cp -f /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

1. 更新文件 find -name "\*.c"|xargs touch

find /dir -type f|xargs touch

1. ./deal: /lib/x86\_64-linux-gnu/libc.so.6: version `GLIBC\_2.14' not found (required by /usr/local/lib/libuserinfo.so)问题：

nm libalgrms.so | grep GLIBC\_2.14 //查看当前引用了GLIBC\_2.14中哪些函数

在.c文件中，#include后面加下面一句话：

\_\_asm\_\_(".symver memcpy,memcpy@GLIBC\_2.13");

1. top命令是Linux下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于Windows的任务管理器。
2. 配置DNS

文件/etc/resolv.conf配置DNS客户，它包含了主机的域名搜索顺序和DNS服务器的地址，每一行应包含一个关键字和一个或多个的由空格隔开的参数。

domain MyPBX

search MyPBX

nameserver 192.168.5.1

nameserver 114.114.114.114

nameserver 114.114.115.115

1. 指定linux地址，如：192.168.5.131：指定固定ip

cd /etc/network 修改目录下的interface

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

auto eth0

iface eth0 inet static

address 192.168.5.70

netmask 255.255.255.0

gateway 192.168.5.1

1. ln -s a b 中的 a 就是源文件，b是链接文件名,其作用是当进入b目录，实际上是链接进入了a目录

删除软链接：

rm -rf b 注意不是rm -rf b/

1. chown [-R] 账号名称 文件或目录 //更改文件拥有者

chown yangzongde testfile //修改文件拥有者为 yangzongde

chgrp [-R] 用户组名称 dirname/filename ... //更改文件所属用户组

chgrp yangzongde testfile //修改拥有者组为 yangzongde

1. xauth命令用于显示和编辑被用于连接X服务器的认证信息。

1）语法：xauth(选项)(参数)  
2）选项：-f：不使用默认的认证文件，而使用指定的认证文件；

-q：安静模式，不打印未请求的状态信息；

-v：详细模式，打印指定的各种操作信息；

-i：忽略认证文件锁定；

-b：执行任何操作，终端认证文件锁定。

3）参数 add：添加认证条目到认证文件中；

extract：将指定的设备内容加入到指定的密码文件中；

info：显示授权文件相关信息；

exit：退出交互模式； list：列出给定的显示设备的内容；

merge：合并多个授权文件内容；

extract：将指定设备内容写入指定的授权文件；

nextrct：将指定设备内容写入指定的授权文件；

nmerge：合并多个授权文件内容；

remove：删除指定显示设备的授权条目；

source：从指定文件读取包含xauth的内容指令。

1. linux 文件系统下/var/cache/apt/archives 主要放置什么类型的文件？

主要apt-get命令下载的deb软件包；另有一个文件lock，随apt-get启动而建立，随apt-get退出而删除，用以保证同一时间只有一个apt-get例程；

1. ubuntu下修改默认sh为bash

先用命令ls -l /bin/sh 看看 结果是： /bin/sh -> dash

如果要修改默认的sh，可以采用命令

sudo dpkg-reconfigure dash 然后选择【否】

成功后再执行ls -l /bin/sh 看看 结果是： /bin/sh -> bash

1. cal命令可以用来显示公历（阳历）日历。

-1 显示一个月的月历

-3 显示系统前一个月，当前月，下一个月的月历

-s 显示星期天为一个星期的第一天，默认的格式

-m 显示星期一为一个星期的第一天

-j 显示在当年中的第几天（一年日期按天算，从1月1号算起，默认显示当前月在一年中的天数）

-y 显示当前年份的日历

cal 9 2015 //显示指定月份的日历

cal -y 2013 //显示2013年日历

cal –j //显示自1月1日的天数

cal –m //星期一显示在第一列

1. head和tail用来显示开头或结尾某个数量的文字区块，head 用来显示档案的开头至标准输出中，而 tail 想当然尔就是看档案的结尾。

head 用来显示档案的开头至标准输出中，默认head命令打印其相应文件的开头10行。

-q 隐藏文件名

-v 显示文件名

-c<字节> 显示字节数

-n<行数> 显示的行数

head -n 5 log2014.log //显示文件的前5行

head -c 20 log2014.log //显示文件前20个字节

head -c -32 log2014.log //文件的除了最后32个字节以外的内容

head -n -6 log2014.log //输出文件除了最后n行的全部内容

tail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。

-f 循环读取

-q 不显示处理信息

-v 显示详细的处理信息

-c<数目> 显示的字节数

-n<行数> 显示行数

--pid=PID 与-f合用,表示在进程ID,PID死掉之后结束.

-q, --quiet, --silent 从不输出给出文件名的首部

-s, --sleep-interval=S 与-f合用,表示在每次反复的间隔休眠S秒

tail -n 5 log2014.log //显示文件末尾内容

tail -f test.log //循环查看文件内容

tail -n +5 log2014.log //从第5行开始显示文件

1. route命令是用于操作基于内核ip路由表，它的主要作用是创建一个静态路由让指定一个主机或者一个网络通过一个网络接口，如eth0。当使用"add"或者"del"参数时，路由表被修改，如果没有参数，则显示路由表当前的内容。

route [-f] [-p] [Command [Destination] [mask Netmask] [Gateway] [metric Metric]] [if Interface]]

-c 显示更多信息

-n 不解析名字

-v 显示详细的处理信息

-F 显示发送信息

-C 显示路由缓存

-f 清除所有网关入口的路由表。

-p 与 add 命令一起使用时使路由具有永久性。

add:添加一条新路由。

del:删除一条路由。

-net:目标地址是一个网络。

-host:目标地址是一个主机。

netmask:当添加一个网络路由时，需要使用网络掩码。

gw:路由数据包通过网关。注意，你指定的网关必须能够达到。

metric：设置路由跳数。

Command 指定您想运行的命令 (Add/Change/Delete/Print)。

Destination 指定该路由的网络目标。

mask Netmask 指定与网络目标相关的网络掩码（也被称作子网掩码）。

Gateway 指定网络目标定义的地址集和子网掩码可以到达的前进或下一跃点 IP 地址。

metric Metric 为路由指定一个整数成本值标（从 1 至 9999），当在路由表(与转发的据包目标地址最匹配)的多个路由中进行选择时可以使用。

if Interface 为可以访问目标的接口指定接口索引。若要获得一个接口列表和它们相应的接口索引，使用 route print 命令的显示功能。可以使用十进制或十六进制值进行接口索引。

1. sed主要是以行为单位进行处理，可以进行替换、删除、新增、选取等特定工作。

sed [-nefri] ‘command’ 输入文本

常用选项：

-n∶使用安静(silent)模式。在一般 sed 的用法中，所有来自 STDIN的资料一般都会被列出到萤幕上。但如果加上 -n 参数后，则只有经过sed 特殊处理的那一行(或者动作)才会被列出来。

-e∶直接在指令列模式上进行 sed 的动作编辑；

-f∶直接将 sed 的动作写在一个档案内，-f filename则可以执行 filename 内的sed 动作；

-r∶sed 的动作支援的是延伸型正规表示法的语法。(预设是基础正规表示法语法)

-i∶直接修改读取的档案内容，而不是由萤幕输出。

常用命令：

a ∶新增， a 的后面可以接字串，而这些字串会在新的一行出现(目前的下一行)～

c ∶取代， c 的后面可以接字串，这些字串可以取代 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 的后面可以接字串，而这些字串会在新的一行出现(目前的上一行)；

p∶打印，亦即将某个选择的资料印出。通常 p 会与参数 sed -n 一起运作～

s ∶取代，可以直接进行取代的工作。

sed '1d' log2014.log //删除第一行

sed '$d' log2014.log //删除最后一行

sed '1,2d' log2014.log //删除第一行到第二行

sed '2,$d' log2014.log //删除第二行到最后一行

sed -n '1p' log2014.log //显示第一行

sed -n '$p' log2014.log //显示最后一行

sed -n '2,$p' log2014.log //显示第二行到最后一行

sed -n '/ruby/p' log2014.log    //查询包括关键字ruby所在所有行

sed -n '/\$/p' log2014.log //查询包括关键字$所在所有行，使用反斜线\屏蔽特殊含义

sed '1a drink tea' log2014.log //第一行后增加字符串"drink tea"

sed '1,3a drink tea' log2014.log //第一行到第三行后增加字符串"drink tea"

sed '1a drink tea\nor coffee' log2014.log   //第一行后增加多行，使用换行符\n

sed '1c Hi' log2014.log //第一行代替为Hi

sed '1,2c Hi' log2014.log //第一行到第二行代替为Hi

格式：sed 's/要替换的字符串/新的字符串/g'  //替换一行中的某部分

sed -n '/ruby/p' log2014.log | sed 's/ruby/bird/g'    //替换ruby为bird

sed -n '/ruby/p' log2014.log | sed 's/ruby//g'        //删除ruby

sed -i '$a bye' log2014.log //在文件log2014.log中最后一行直接输入"bye"

1. cut 命令从文件的每一行剪切字节、字符和字段并将这些字节、字符和字段写至标准输出。

-b ：以字节为单位进行分割。这些字节位置将忽略多字节字符边界,除非也指定了-n标志。

-c ：以字符为单位进行分割。

-d ：自定义分隔符，默认为制表符。

-f ：与-d一起使用，指定显示哪个区域。

-n ：取消分割多字节字符。仅和 -b 标志一起使用。如果字符的最后一个字节落在由 -b 标志的 List 参数指示的<br />范围之内，该字符将被写出；否则，该字符将被排除。

cut -d : -f 1 //用-d来设置间隔符为冒号，然后用-f来设置我要取的是第一个域

cut -d : -f 1,3-5 //用-d来设置间隔符为冒号，用-f来设置我要取的是第1和第3到5域

1. awk工作流程是这样的：读入有'\n'换行符分割的一条记录，然后将记录按指定的域分隔符划分域，填充域，$0则表示所有域,$1表示第一个域,$n表示第n个域。

$ awk '/^(no|so)/' log2014.log-----打印所有以模式no或so开头的行。

$ awk '/^[ns]/{print $1}' log2014.log-----如果记录以n或s开头，就打印这个记录。

$ awk '$1 ~/[0-9][0-9]$/(print $1}' log2014.log-----如果第一个域以两个数字结束就打印这个记录。

$ awk '$1 == 100 || $2 < 50' log2014.log-----如果第一个或等于100或者第二个域小于50，则打印该行。

$ awk '$1 != 10' log2014.log-----如果第一个域不等于10就打印该行。

$ awk '/test/{print $1 + 10}' log2014.log-----如果记录包含正则表达式test，则第一个域加10并打印出来。

$ awk '{print ($1 > 5 ? "ok "$1: "error"$1)}' log2014.log-----如果第一个域大于5则打印问号后面的表达式值，否则打印冒号后面的表达式值。

$ awk '/^root/,/^mysql/' log2014.log-----打印以正则表达式root开头的记录到以正则表达式mysql开头的记录范围内的所有记录。如果找到一个新的正则表达式root开头的记 录，则继续打印直到下一个以正则表达式mysql开头的记录为止，或到文件末尾。

例：awk ‘$1 == “inet” {print $2}’

1. uname可显示电脑以及操作系统的相关信息

-a或--all 　//显示全部的信息。

-m或--machine 　//显示电脑类型。

-n或-nodename 　//显示在网络上的主机名称。

-r或--release 　 //显示操作系统的发行编号。

-s或--sysname 　 //显示操作系统名称。

-p  //处理器类型和名称（我这里显示 “unknown”）

-v 　 //显示操作系统的版本

1. arch 命令显示机器的处理器架构
2. strip命令通过除去绑定程序和符号调试程序使用的信息，减少扩展公共对象文件格式（XCOFF）的对象文件的大小。

strip 命令从 XCOFF 对象文件中有选择地除去行号信息、重定位信息、调试段、typchk 段、注释段、文件头以及所有或部分符号表。 一旦您使用该命令，则很难调试文件的符号；因此，通常应该只在已经调试和测试过的生成模块上使用 strip 命令。

1. grep命令：用于过滤/搜索的特定字符。可使用正则表达式能多种命令配合使用

命令参数：

-a --text #不要忽略二进制的数据。

-A<显示行数> --after-context=<显示行数> #除了显示符合范本样式的那一列之外，并显示该行之后的内容。

-b --byte-offset #在显示符合样式的那一行之前，标示出该行第一个字符的编号。

-B<显示行数> --before-context=<显示行数> #除了显示符合样式的那一行之外，并显示该行之前的内容。

-c --count #计算符合样式的列数。

-C<显示行数> --context=<显示行数>或-<显示行数> #除了显示符合样式的那一行之外，并显示该行之前后的内容。

-d <动作> --directories=<动作> #当指定要查找的是目录而非文件时，必须使用这项参数，否则grep指令将回报信息并停止动作。

-e<范本样式> --regexp=<范本样式> #指定字符串做为查找文件内容的样式。

-E --extended-regexp #将样式为延伸的普通表示法来使用。

-f<规则文件> --file=<规则文件> #指定规则文件，其内容含有一个或多个规则样式，让grep查找符合规则条件的文件内容，格式为每行一个规则样式。

-F --fixed-regexp #将样式视为固定字符串的列表。

-G --basic-regexp #将样式视为普通的表示法来使用。

-h --no-filename #在显示符合样式的那一行之前，不标示该行所属的文件名称。

-H --with-filename #在显示符合样式的那一行之前，表示该行所属的文件名称。

-i --ignore-case #忽略字符大小写的差别。

-l --file-with-matches #列出文件内容符合指定的样式的文件名称。

-L --files-without-match #列出文件内容不符合指定的样式的文件名称。

-n --line-number #在显示符合样式的那一行之前，标示出该行的列数编号。

-q --quiet或--silent #不显示任何信息。

-r --recursive #此参数的效果和指定“-d recurse”参数相同。

-s --no-messages #不显示错误信息。

-v --revert-match #显示不包含匹配文本的所有行。

-V --version #显示版本信息。

-w --word-regexp #只显示全字符合的列。

-x --line-regexp #只显示全列符合的列。

-y #此参数的效果和指定“-i”参数相同。

grep的规则表达式:

^ #锚定行的开始 如：'^grep'匹配所有以grep开头的行。

$ #锚定行的结束 如：'grep$'匹配所有以grep结尾的行。

. #匹配一个非换行符的字符 如：'gr.p'匹配gr后接一个任意字符，然后是p。

\* #匹配零个或多个先前字符 如：'\*grep'匹配所有一个或多个空格后紧跟grep的行。

.\* #一起用代表任意字符。

[] #匹配一个指定范围内的字符，如'[Gg]rep'匹配Grep和grep。

[^] #匹配一个不在指定范围内的字符，如：'[^A-FH-Z]rep'匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。

\(..\) #标记匹配字符，如'\(love\)'，love被标记为1。

\< #锚定单词的开始，如:'\<grep'匹配包含以grep开头的单词的行。

\> #锚定单词的结束，如'grep\>'匹配包含以grep结尾的单词的行。

x\{m\} #重复字符x，m次，如：'0\{5\}'匹配包含5个o的行。

x\{m,\} #重复字符x,至少m次，如：'o\{5,\}'匹配至少有5个o的行。

x\{m,n\} #重复字符x，至少m次，不多于n次，如：'o\{5,10\}'匹配5--10个o的行。

\w #匹配文字和数字字符，也就是[A-Za-z0-9]，如：'G\w\*p'匹配以G后跟零个或多个文字或数字字符，然后是p。

\W #\w的反置形式，匹配一个或多个非单词字符，如点号句号等。

\b #单词锁定符，如: '\bgrep\b'只匹配grep。

1. tree命令：显示指定驱动器上所有[目录路径](http://baike.baidu.com/view/2725072.htm)和这些目录下的所有文件名。
2. SUID、SGID详解

<http://www.cnblogs.com/fhefh/archive/2011/09/20/2182155.html>

给文件加SUID和SUID的命令如下：

　　chmod u+s filename //设置SUID位

　　chmod u-s filename //去掉SUID设置

　　chmod g+s filename //设置SGID位

　　chmod g-s filename //去掉SGID设置

例：# chmod 4777 a.txt ===== > -rwsrwxrwx //注意s位置

# chmod 2777 a.txt 　 ===== >-rwxrwsrwx //注意s位置

#chmod 7777 a.txt  ==== > -rwsrwxswt //出现了t,t的作用在内存中尽量保存a.txt,节省系统再加载的时间。

find / -perm -04000 -type f –ls //找到系统上所有的设置了suid的文件

1. tar命令

<http://blog.csdn.net/eroswang/article/details/5555415>

[root@linux ~]# tar [-cxtzjvfpPN] 文件与目录 ....

参数：

-c ：建立一个压缩文件的参数指令(create 的意思)；

-x ：解开一个压缩文件的参数指令！

-t ：查看 tarfile 里面的文件！

特别注意，在参数的下达中， c/x/t 仅能存在一个！不可同时存在！

因为不可能同时压缩与解压缩。

-z ：是否同时具有 gzip 的属性？亦即是否需要用 gzip 压缩？

-j ：是否同时具有 bzip2 的属性？亦即是否需要用 bzip2 压缩？

-v ：压缩的过程中显示文件！这个常用，但不建议用在背景执行过程！

-f ：使用档名，请留意，在 f 之后要立即接档名喔！不要再加参数！

　　　例如使用『 tar -zcvfP tfile sfile』就是错误的写法，要写成

　　　『 tar -zcvPf tfile sfile』才对喔！

-p ：使用原文件的原来属性（属性不会依据使用者而变）

-P ：可以使用绝对路径来压缩！

-N ：比后面接的日期(yyyy/mm/dd)还要新的才会被打包进新建的文件中！

--exclude FILE：在压缩的过程中，不要将 FILE 打包！

--strip-components=NUMBER //去除文件名前的NUMBER 层路径

例：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar

tar -cvf /tmp/etc.tar /etc <==仅打包，不压缩！

tar -zcvf /tmp/etc.tar.gz /etc <==打包后，以 gzip 压缩

tar -jcvf /tmp/etc.tar.bz2 /etc <==打包后，以 bzip2 压缩

1. curl命令

<http://www.cnblogs.com/gbyukg/p/3326825.html>

1. 下载单个文件，默认将输出打印到标准输出中(STDOUT)中

curl http://www.centos.org

1. 通过-o/-O选项保存下载的文件到指定的文件中：

-o：将文件保存为命令行中指定的文件名的文件中

-O：使用URL中默认的文件名保存文件到本地

//将文件下载到本地并命名为mygettext.html

curl -o mygettext.html http://www.gnu.org/software/gettext/manual/gettext.html

//将文件保存到本地并命名为gettext.html

curl -O http://www.gnu.org/software/gettext/manual/gettext.html

1. 使用-L选项进行强制重定向

//让curl使用地址重定向，此时会查询google.com.hk站点

curl -L http://www.google.com

1. 使用-C选项可对大文件使用断点续传功能

//当文件在下载完成之前结束该进程

$ curl -O http://www.gnu.org/software/gettext/manual/gettext.html

############## 20.1%

//通过添加-C选项继续对该文件进行下载，已经下载过的文件不会被重新下载

curl -C - -O http://www.gnu.org/software/gettext/manual/gettext.html

############### 20.1%

1. 通过--limit-rate选项对CURL的最大网络使用进行限制

//下载速度最大不会超过1000B/second

curl --limit-rate 1000B -O http://www.gnu.org/software/gettext/manual/gettext.html

1. 在访问需要授权的页面时，可通过-u选项提供用户名和密码进行授权

curl -u username:password URL

//通常的做法是在命令行只输入用户名，之后会提示输入密码，这样可以保证在查看历史记录时不会将密码泄露

curl -u username URL

1. CURL同样支持FTP下载，若在url中指定的是某个文件路径而非具体的某个要下载的文件名，CURL则会列出该目录下的所有文件名而并非下载该目录下的所有文件

//列出public\_html下的所有文件夹和文件

curl -u ftpuser:ftppass -O ftp://ftp\_server/public\_html/

//下载xss.php文件

curl -u ftpuser:ftppass -O ftp://ftp\_server/public\_html/xss.php

1. 通过 -T 选项可将指定的本地文件上传到FTP服务器上

//将myfile.txt文件上传到服务器

curl -u ftpuser:ftppass -T myfile.txt ftp://ftp.testserver.com

//同时上传多个文件

curl -u ftpuser:ftppass -T "{file1,file2}" ftp://ftp.testserver.com

//从标准输入获取内容保存到服务器指定的文件中

curl -u ftpuser:ftppass -T - ftp://ftp.testserver.com/myfile\_1.txt

1. -x 选项可以为CURL添加代理功能

//指定代理主机和端口

curl -x proxysever.test.com:3128 http://google.co.in

1. 保存与使用网站cookie信息

//将网站的cookies信息保存到sugarcookies文件中

curl -D sugarcookies http://localhost/sugarcrm/index.php

//使用上次保存的cookie信息

curl -b sugarcookies http://localhost/sugarcrm/index.php

1. 传递请求数据

默认curl使用GET方式请求数据，这种方式下直接通过URL传递数据，可以通过 --data/-d 方式指定使用POST方式传递数据

//GET

curl -u username https://api.github.com/user?access\_token=XXXXXXXXXX

//POST

curl -u username --data "param1=value1&param2=value" https://api.github.com

//也可以指定一个文件，将该文件中的内容当作数据传递给服务器端

curl --data @filename https://github.api.com/authorizations

注：默认情况下，通过POST方式传递过去的数据中若有特殊字符，首先需要将特殊字符转义在传递给服务器端，如value值中包含有空格，则需要先将空格转换成%20，如： curl -d "value%201" http://hostname.com

//在新版本的CURL中，提供了新的选项 --data-urlencode，通过该选项提供的参数会自动转义特殊字符。

curl --data-urlencode "value 1" http://hostname.com

//除了使用GET和POST协议外，还可以通过 -X 选项指定其它协议，如：

curl -I -X DELETE https://api.github.cim

//上传文件

curl --form "fileupload=@filename.txt" http://hostname/resource

1. other
2. diff命令 用于比较文件的内容，特别是比较两个版本不同的文件以找到改动的地方。

如果指定比较的是文件，则只有当输入为文本文件时才有效。以逐行的方式，比较文本文件的异同处。如果指定比较的是目录的的时候，diff 命令会比较两个目录下名字相同的文本文件。

diff [参数] [文件1或目录1] [文件2或目录2]

-c 显示全部内文，并标出不同之处。

-y或--side-by-side 以并列的方式显示文件的异同之处。

-W或--width 在使用-y参数时，指定栏宽。

如：

“3c3”和“8c8”表示log2014.log和log2015.log文件在3行和第8行内容有所不同

diff log2013.log log2014.log -y -W 50

“|”表示前后2个文件内容有不同

“<”表示后面文件比前面文件少了1行内容

“>”表示后面文件比前面文件多了1行内容

diff log2014.log log2013.log  -c

“＋” 比较的文件的后者比前着多一行

“－” 比较的文件的后者比前着少一行

“！” 比较的文件两者有差别的行

1. patch命令

利用diff制作的补丁来实现源文件（夹）和目的文件（夹）的转换。

-p0 选项要从当前目录查找目的文件（夹）

-p1 选项要忽略掉第一层目录，从当前目录开始查找。

-E 选项说明如果发现了空文件，那么就删除它

-R 选项说明在补丁文件中的“新”文件和“旧”文件现在要调换过来了（实际上就是给新版本打补丁，让它变成老版本）

例：

patch 2013.log log.patch

patch 2014.log log.patch -R

1. wget命令

<http://www.cnblogs.com/peida/archive/2013/03/18/2965369.html>

用于从网络上下载资源，没有指定目录，下载资源回默认为当前目录。

wget [参数] [URL地址]

-P, –directory-prefix=PREFIX 将文件保存到目录

1）、//使用wget -O下载并以不同的文件名保存

wget -O wordpress.zip http://www.minjieren.com/download.aspx?id=1080

2）、//使用wget –limit -rate限速下载

wget --limit-rate=300k http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

3）、//使用wget -c断点续传

wget -c http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

4）、//使用wget -b后台下载

wget -b http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

1. sort命令

sort命令是帮我们依据不同的数据类型进行排序，其语法及常用参数格式：

　　sort [-bcfMnrtk][源文件][-o 输出文件]

补充说明：sort可针对文本文件的内容，以行为单位来排序。

参　　数：

-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。

-f 排序时，忽略大小写字母。

-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。

-u在输出行中去除重复行

1. dpkg命令

//查询deb包的详细信息，没有指定包则显示全部已安装包

dpkg –l

dpkg -l |grep vim

//查看已经安装的指定软件包的详细信息

dpkg -s vim

//列出一个包安装的所有文件清单

dpkg -L vim

//查看系统中的某个文件属于那个软件包

dpkg -S vimrc

//所有deb文件的安装

dpkg -i

//所有deb文件的卸载

dpkg -r

//彻底的卸载，包括软件的配置文件

dpkg -P

//查询deb包文件中所包含的文件

dpkg -c

//查看系统中安装包的的详细清单，同时执行 -c

dpkg -L

1. blkid命令

对查询设备上所采用文件系统类型进行查询。blkid主要用来对系统的块设备（包括交换分区）所使用的文件系统类型、LABEL、UUID等信息进行查询。

1）、sudo blkid //列出当前系统中所有已挂载文件系统的类型

2）、sudo blkid -s UUID /dev/sda5 //显示指定设备 UUID

3）、sudo blkid -s LABEL /dev/sda5 //显示指定设备 LABEL

4）、sudo blkid -s TYPE //显示所有设备文件系统

5）、sudo blkid -o device 8 //显示所有设备

6）、sudo blkid -o list //以列表方式查看详细信息

1. hdparm 命令：获取/设置硬盘参数

hdparm(选项)(参数)

1）、hdparm -T /dev/sda //测试硬盘的读取速度

2）、hdparm -C /dev/sda //检测硬盘的电源管理模式

3）、检查smartctl -l selftest /dev/sda 卸载：umount /dev/sda\* 修复：badblocks /dev/sda

//硬盘坏道修复方法

4）、hdparm /dev/sda //显示硬盘的相关设置

1. udevadm udev 管理工具

udevadm 后接一个命令和命令指定选项。它控制了udev运行的行为，处理内核事件，控制事件队列，并且提供简单的调试机制。

devadm info options

查询udev数据库中的设备信息。也可以从sysfs文件系统中查询到设备的属性以辅助创建udev规则。

--query=type

从数据库中查询指定类型的设备。需要--path和--name来指定设备。合法的查询文件是：设备名，链接，路径，属性

--path=devpath //设备的路径

--name=file //设备节点或者链接

--attribute-walk

打印指定设备的所有sysfs记录的属性，以用来udev规则匹配特殊的设备。该选项打印链上的所有设备信息，最大可能到sys目录。

1）、udevadm info --query=all --path=/block/sda //列出设备信息

1. chroot命令用来在指定的根目录下运行指令。chroot，即 change root directory （更改 root 目录）。在 linux 系统中，系统默认的目录结构都是以/，即是以根 (root) 开始的。而在使用 chroot 之后，系统的目录结构将以指定的位置作为/位置。

例：

将target作为根目录（运行其中的/bin/sh）: # chroot target /bin/sh

改变根目录 # chroot /mnt/ls

1. parted命令 硬盘分区工具 一般用于创建一个大于2TB的分区

<http://blog.csdn.net/bluishglc/article/details/9189437>

1. getconf命令 获取系统信息

<http://blog.chinaunix.net/uid-23105261-id-109513.html>

1）、getconf LONG\_BIT //一般用作获取系统是多少位

1. kexec命令

安装命令：sudo apt-get install kexec-tools

1. setserial 命令用于设置或显示串口的相关信息

sudo apt-get install setserial

setserial [-abgGqvVz][设备][串口参数]

参数说明：

-a 显示详细信息。

-b 显示摘要信息。

-g 显示串口的相关信息。

-G 以指令列表的格式来显示信息。

-q 执行时显示较少的信息。

-v 执行时显示较多的信息。

-V 显示版本信息。

-z 设置前，先将所有的标记归零。

例：

1）# setserial -g /dev/ttyS2 //显示串口信息

1. pkill 命令：是ps命令和kill命令的结合,用来杀死指定进程

-f 显示完整程序

-l 显示源代码

-n 显示新程序

-o 显示旧程序

-v 与条件不符合的程序

-x 与条件符合的程序

-p<进程号> 列出父进程为用户指定进程的进程信息

-t<终端> 指定终端下的所有程序

-u<用户> 指定用户的程序

例：

1）、pkill -t tty1 //杀死终端1下的所有进程

2）、pkill -kill -t pts/2 //踢用户

3）、pkill -9 6985 //彻底杀死进程

4）、pkill -u tank //杀死指定用户的所有进程

5）、pkill -vu root //杀死不属于root用户的所有进程

1. ps命令

-A 显示所有进程（等价于-e）(utility)

-a 显示一个终端的所有进程，除了会话引线

-N 忽略选择。

-d 显示所有进程，但省略所有的会话引线(utility)

-x 显示没有控制终端的进程，同时显示各个命令的具体路径。dx不可合用。（utility）

-p pid 进程使用cpu的时间

-u uid or username 选择有效的用户id或者是用户名

-g gid or groupname 显示组的所有进程。

U username 显示该用户下的所有进程，且显示各个命令的详细路径。如:ps U zhang;(utility)

-f 全部列出，通常和其他选项联用。如：ps -fa or ps -fx and so on.

-l 长格式（有F,wchan,C 等字段）

-j 作业格式

-o 用户自定义格式。

v 以虚拟存储器格式显示

s 以信号格式显示

-m 显示所有的线程

-H 显示进程的层次(和其它的命令合用，如：ps -Ha)（utility）

e 命令之后显示环境（如：ps -d e; ps -a e）(utility)

h 不显示第一行

linux上进程有5种状态:

1. 运行(正在运行或在运行队列中等待)

2. 中断(休眠中, 受阻, 在等待某个条件的形成或接受到信号)

3. 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)

4. 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)

5. 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

< 高优先级

N 低优先级

L 有些页被锁进内存

s 包含子进程

+ 位于后台的进程组；

l 多线程，克隆线程 multi-threaded (using CLONE\_THREAD, like NPTL pthreads do)

ps -elf与ps aux是一样的意思, 显示所有在运行的进程, 一个是用标准格式输出, 一个是用bsd的格式输出.

标准格式: USER , PID , %CPU , %MEM , VSZ , RSS , TTY , STAT , START , TIME , COMMAND  
bsd格式: UID , PID , PPID , C , STIME , TTY , TIME , CMD

1. cron命令表达式是一个字符串，字符串以5或6个空格隔开，分为6或7个域，每一个域代表一个含义，Cron有如下两种语法格式：

<http://www.cnblogs.com/linjiqin/archive/2013/07/08/3178452.html>

cron的系统级配置文件位于/etc/crontab。cron服务每分钟不仅要读一次/var/spool/cron内的所有文件，还需要读一次/etc/crontab配置文件,因此我们配置这个文件也能运用 cron服务做一些事情。用crontab -e进行的配置是针对某个用户的，而编辑/etc/crontab是针对系统的任务。

/etc/init.d/cron force-reload|reload|restart|start|status|stop

Seconds Minutes Hours DayofMonth Month DayofWeek Year或

Seconds Minutes Hours DayofMonth Month DayofWeek

每一个域可出现的字符如下：

Seconds:可出现", - \* /"四个字符，有效范围为0-59的整数

Minutes:可出现", - \* /"四个字符，有效范围为0-59的整数

Hours:可出现", - \* /"四个字符，有效范围为0-23的整数

DayofMonth:可出现", - \* / ? L W C"八个字符，有效范围为0-31的整数

Month:可出现", - \* /"四个字符，有效范围为1-12的整数或JAN-DEc

DayofWeek:可出现", - \* / ? L C #"四个字符，有效范围为1-7的整数或SUN-SAT两个范围。1表示星期天，2表示星期一， 依次类推

Year:可出现", - \* /"四个字符，有效范围为1970-2099年

每一个域都使用数字，但还可以出现如下特殊字符，它们的含义是：

(1)\*：表示匹配该域的任意值，假如在Minutes域使用\*, 即表示每分钟都会触发事件。

(2)?:只能用在DayofMonth和DayofWeek两个域。它也匹配域的任意值，但实际不会。因为DayofMonth和DayofWeek会相互影响。例如想在每月的20日触发调度，不管20日到底是星期几，则只能使用如下写法： 13 13 15 20 \* ?, 其中最后一位只能用？，而不能使用\*，如果使用\*表示不管星期几都会触发，实际上并不是这样。

(3)-:表示范围，例如在Minutes域使用5-20，表示从5分到20分钟每分钟触发一次

(4)/：表示起始时间开始触发，然后每隔固定时间触发一次，例如在Minutes域使用5/20,则意味着5分钟触发一次，而25，45等分别触发一次.

(5),:表示列出枚举值值。例如：在Minutes域使用5,20，则意味着在5和20分每分钟触发一次。

(6)L:表示最后，只能出现在DayofWeek和DayofMonth域，如果在DayofWeek域使用5L,意味着在最后的一个星期四触发。

(7)W:表示有效工作日(周一到周五),只能出现在DayofMonth域，系统将在离指定日期的最近的有效工作日触发事件。例如：在 DayofMonth使用5W，如果5日是星期六，则将在最近的工作日：星期五，即4日触发。如果5日是星期天，则在6日(周一)触发；如果5日在星期一到星期五中的一天，则就在5日触发。另外一点，W的最近寻找不会跨过月份

(8)LW:这两个字符可以连用，表示在某个月最后一个工作日，即最后一个星期五。

(9)#:用于确定每个月第几个星期几，只能出现在DayofMonth域。例如在4#2，表示某月的第二个星期三。

举几个例子:

0 0 2 1 \* ? \* 表示在每月的1日的凌晨2点调度任务

0 15 10 ? \* MON-FRI 表示周一到周五每天上午10：15执行作业

0 15 10 ? 6L 2002-2006 表示2002-2006年的每个月的最后一个星期五上午10:15执行作

一个cron表达式有至少6个（也可能7个）有空格分隔的时间元素。

按顺序依次为

秒（0~59） 分钟（0~59） 小时（0~23）天（月）（0~31，但是你需要考虑你月的天数

月（0~11）天（星期）（1~7 1=SUN 或 SUN，MON，TUE，WED，THU，FRI，SAT）

年份（1970－2099）

其中每个元素可以是一个值(如6),一个连续区间(9-12),一个间隔时间(8-18/4)(/表示每隔4小时),一个列表(1,3,5),通配符。由于"月份中的日期"和"星期中的日期"这两个元素互斥的,必须要对其中一个设置?

0 0 10,14,16 \* \* ? 每天上午10点，下午2点，4点

0 0/30 9-17 \* \* ? 朝九晚五工作时间内每半小时

0 0 12 ? \* WED 表示每个星期三中午12点

"0 0 12 \* \* ?" 每天中午12点触发

"0 15 10 ? \* \*" 每天上午10:15触发

"0 15 10 \* \* ?" 每天上午10:15触发

"0 15 10 \* \* ? \*" 每天上午10:15触发

"0 15 10 \* \* ? 2005" 2005年的每天上午10:15触发

"0 \* 14 \* \* ?" 在每天下午2点到下午2:59期间的每1分钟触发

"0 0/5 14 \* \* ?" 在每天下午2点到下午2:55期间的每5分钟触发

"0 0/5 14,18 \* \* ?" 在每天下午2点到2:55期间和下午6点到6:55期间的每5分钟触发

"0 0-5 14 \* \* ?" 在每天下午2点到下午2:05期间的每1分钟触发

"0 10,44 14 ? 3 WED" 每年三月的星期三的下午2:10和2:44触发

"0 15 10 ? \* MON-FRI" 周一至周五的上午10:15触发

"0 15 10 15 \* ?" 每月15日上午10:15触发

"0 15 10 L \* ?" 每月最后一日的上午10:15触发

"0 15 10 ? \* 6L" 每月的最后一个星期五上午10:15触发

"0 15 10 ? \* 6L 2002-2005" 2002年至2005年的每月的最后一个星期五上午10:15触发

"0 15 10 ? \* 6#3" 每月的第三个星期五上午10:15触发

有些子表达式能包含一些范围或列表

例如：子表达式（天（星期））可以为 “MON-FRI”，“MON，WED，FRI”，“MON-WED,SAT”

“\*”字符代表所有可能的值

因此，“\*”在子表达式（月）里表示每个月的含义，“\*”在子表达式（天（星期））表示星期的每一天

“/”字符用来指定数值的增量

例如：在子表达式（分钟）里的“0/15”表示从第0分钟开始，每15分钟

在子表达式（分钟）里的“3/20”表示从第3分钟开始，每20分钟（它和“3，23，43”）的含义一样

“？”字符仅被用于天（月）和天（星期）两个子表达式，表示不指定值

当2个子表达式其中之一被指定了值以后，为了避免冲突，需要将另一个子表达式的值设为“？”

“L” 字符仅被用于天（月）和天（星期）两个子表达式，它是单词“last”的缩写

但是它在两个子表达式里的含义是不同的。

在天（月）子表达式中，“L”表示一个月的最后一天

在天（星期）自表达式中，“L”表示一个星期的最后一天，也就是SAT

如果在“L”前有具体的内容，它就具有其他的含义了

例如：“6L”表示这个月的倒数第６天，“FRIL”表示这个月的最一个星期五

注意：在使用“L”参数时，不要指定列表或范围，因为这会导致问题

1. cron服务提供crontab命令来设定cron服务的，以下是这个命令的一些参数与说明：

格式： minute hour day month dayofweek command

crontab –u //设定某个用户的cron服务，一般root用户在执行这个命令的时候需要此参数

crontab -l //列出某个用户cron服务的详细内容

crontab -r //删除某个用户的cron服务

crontab –e //编辑某个用户的cron服务

如root查看自己的cron设置：crontab -u root -l

root想删除fred的cron设置：crontab -u fred -r

crontab -u root -e 进入vi编辑模式

例：

1）、30 21 \* \* \* /usr/local/etc/rc.d/lighttpd restart

//上面的例子表示每晚的21:30重启lighttpd 。

2）、\* 23-7/1 \* \* \* /usr/local/etc/rc.d/lighttpd restart

//晚上11点到早上7点之间，每隔一小时重启lighttpd

1. mount命令

mount [-t vfstype] [-o options] device dir

1).-t vfstype 指定文件系统的类型，通常不必指定。mount 会自动选择正确的类型。

常用类型有：

　光盘或光盘镜像：iso9660

　DOS fat16文件系统：msdos

　Windows 9x fat32文件系统：vfat

　Windows NT ntfs文件系统：ntfs

　Mount Windows文件网络共享：smbfs

　UNIX(LINUX) 文件网络共享：nfs

2).-o options 主要用来描述设备或档案的挂接方式。常用的参数有：

　loop：用来把一个文件当成硬盘分区挂接上系统

　ro：采用只读方式挂接设备

　rw：采用读写方式挂接设备

　iocharset：指定访问文件系统所用字符集

3).device 要挂接(mount)的设备。

4).dir设备在系统上的挂接点(mount point)。

1. make-kpkg命令

**用make-kpkg制作内核安装包deb**

sudo apt-get install kernel-package

先配置完内核，然后：

make-kpkg --initrd --revision dlrc.001 --append-to-version -20130121 kernel\_image

1. --initrd： 生成initramfs

2. --revision： deb文件的版本信息，只影响文件名

3. --append-to-version： 内核版本附加信息

4. kernel\_image：内核和模块的安装包

5. kernel\_headers，生成内核头文件的安装包。

make-kpkg --initrd --revision dlrc.001 --append-to-version -20130121 kernel\_image kernel\_headers

1. ar命令

<http://blog.csdn.net/zougangx/article/details/3073604>

功能说明：建立或修改备存文件，或是从备存文件中抽取文件。

指令参数：

　-d 　删除备存文件中的成员文件。

　-m 　变更成员文件在备存文件中的次序。

　-p 　显示备存文件中的成员文件内容。

　-q 　将问家附加在备存文件末端。

　-r 　将文件插入备存文件中。

　-t 　显示备存文件中所包含的文件。

　-x 　自备存文件中取出成员文件。

选项参数：

　a<成员文件> 　将文件插入备存文件中指定的成员文件之后。

　b<成员文件> 　将文件插入备存文件中指定的成员文件之前。

　c 　建立备存文件。

　f 　为避免过长的文件名不兼容于其他系统的ar指令指令，因此可利用此参数，截掉要放入备存文件中过长的成员文件名称。

　i<成员文件> 　将问家插入备存文件中指定的成员文件之前。

　o 　保留备存文件中文件的日期。

　s 　若备存文件中包含了对象模式，可利用此参数建立备存文件的符号表。

　S 　不产生符号表。

　u 　只将日期较新文件插入备存文件中。

　v 　程序执行时显示详细的信息。

　V 　显示版本信息。

例： 1）ar vx linux-image-2.6.39.001\_001\_amd64.deb

1. smbclient命令属于samba套件，它提供一种命令行使用交互式方式访问samba服务器的共享资源

smbclient(选项)(参数)

-B：传送广播数据包时所用的IP地址；

-d<排错层级>：指定记录文件所记载事件的详细程度；

-E：将信息送到标准错误输出设备；

-h：显示帮助；

-i<范围>：设置NetBIOS名称范围；

-I：指定服务器的IP地址；

-l<记录文件>：指定记录文件的名称；

-L：显示服务器端所分享出来的所有资源；

-M：可利用WinPopup协议，将信息送给选项中所指定的主机；

-n：指定用户端所要使用的NetBIOS名称；

-N：不用询问密码；

-O<连接槽选项>：设置用户端TCP连接槽的选项；

-p：指定服务器端TCP连接端口编号；

-R<名称解析顺序>：设置NetBIOS名称解析的顺序；

-s<目录>：指定smb.conf所在的目录；

-t<服务器字码>：设置用何种字符码来解析服务器端的文件名称；

-T：备份服务器端分享的全部文件，并打包成tar格式的文件；

-U<用户名称>：指定用户名称；

-w<工作群组>：指定工作群组名称。

例：

1. smbclient -L 198.168.0.1 -U username%password

//列出某个IP地址所提供的共享文件夹

1. smbclient //192.168.0.1/tmp -U username%password

//像ftp客户端一样使用smbclient

1. smbclient -c "ls" //192.168.0.1/tmp -U username%password

smbclient //192.168.0.1/tmp -U username%password

smb:/>ls

//直接一次性使用smbclient命令

1. smbclient -c "mkdir share1" //192.168.0.1/tmp -U username%password

//创建一个共享文件夹

1. vmstat命令

procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----

r b swpd free buff cache si so bi bo in cs us sy id wa

0 0 100152 2436 97200 289740 0 1 34 45 99 33 0 0 99 0

procs

r 列表示运行和等待cpu时间片的进程数，如果长期大于1，说明cpu不足，需要增加cpu。

b 列表示在等待资源的进程数，比如正在等待I/O、或者内存交换等。

cpu 表示cpu的使用状态

us 列显示了用户方式下所花费 CPU 时间的百分比。us的值比较高时，说明用户进程消耗的cpu时间多，但是如果长期大于50%，需要考虑优化用户的程序。

sy 列显示了内核进程所花费的cpu时间的百分比。这里us + sy的参考值为80%，如果us+sy 大于 80%说明可能存在CPU不足。

wa 列显示了IO等待所占用的CPU时间的百分比。这里wa的参考值为30%，如果wa超过30%，说明IO等待严重，这可能是磁盘大量随机访问造成的，也可能磁盘或者磁盘访问控制器的带宽瓶颈造成的(主要是块操作)。

id 列显示了cpu处在空闲状态的时间百分比

system 显示采集间隔内发生的中断数

in 列表示在某一时间间隔中观测到的每秒设备中断数。

cs列表示每秒产生的上下文切换次数，如当 cs 比磁盘 I/O 和网络信息包速率高得多，都应进行进一步调查。

memory

swpd 切换到内存交换区的内存数量(k表示)。如果swpd的值不为0，或者比较大，比如超过了100m，只要si、so的值长期为0，系统性能还是正常

free 当前的空闲页面列表中内存数量(k表示)

buff 作为buffer cache的内存数量，一般对块设备的读写才需要缓冲。

cache: 作为page cache的内存数量，一般作为文件系统的cache，如果cache较大，说明用到cache的文件较多，如果此时IO中bi比较小，说明文件系统效率比较好。

swap

si 由内存进入内存交换区数量。

so由内存交换区进入内存数量。

IO

bi 从块设备读入数据的总量（读磁盘）（每秒kb）。

bo 块设备写入数据的总量（写磁盘）（每秒kb）

1. setsid命令 能使进程完全独立出来，从而摆脱其他进程的控制

例：

1. setsid ping www.baidu.com //所产生的进程不是当前进程的ID
2. 注意事项：
3. 查看linux和windows系统是32为还是64位:

uname –a 或 file /bin/ls

1. 其它：