## apt-get

apt-get install xxx无法安装时：

/etc/resolv.conf添加 nameserver 8.8.8.8

source /etc/resolv.conf

apt-get update

apt-get install xx

参考：blog.csdn.net/tiny\_lxf/article/details/75027865

【apt-get upgrade、pat-get update】

apt-get 安装默认目录：/use/share，可执行文件在/usr/bin，安装的库在/usr/lib，第三方库/usr/local/lib

## snap

ubuntu16.04引入了snap包管理方式，它和dpkg/apt有本质的区别。snap安装扩展名是.snap，类似一个容器，包含一个应用用到的所有文件和库。它解决了包依赖问题，也是的应用程序更加容易管理。

snap list：列出以安装

snap find：查询

snap find app\_name

snap install app\_name

snap refresh

snap remove

## vim

<https://coolshell.cn/articles/5426.html>

### 查看不可见字符：

：set invlist //显示不可见字符

：set nolist //不显示不可见字符

### 各种插入姿势：

a ：光标后插

o ：下行插

O ：上行插

cw ：替换单词

### 简单光标移动：

0 ：行头

^ ：首字符

$ ：行末尾

g\_ ：尾字符

### 拷贝粘贴：

p ：粘贴

yy ：复制一行

### undo/redo：

u ：undo

ctrl+r ：redo

### 打开/保存/退出/改变文件：

：e filepath -》打开文件

：w -〉存盘

：x 、：wq -》保存并退出

：q！ -〉强制退出

：n、：p -》打开多个文件时，切换文件

### 命令重复：

.（小数点） :重复上次命令

N<command> :重复某个命令N次

### 更强光标移动：

：N -》移动到第N行

gg ： 移动到第一行

G ：移动到最后一行

w ：移动到下一个单词开头

e ：移动到下一个单词末尾

% ：匹配括号

\* 、# ：匹配光标所在单词

fa ：到下一个为a的字符处

Fa ：到上一个为a的字符处

Ctrl+f 和ctrl+b：上下翻页

Ctrl+u和ctrl+d：上下翻半页

### 更快

<start position> <commands><end poition>

0y$：先到行头，拷贝，到行末尾

gU：行变大写

gu：行变小写

### vim超能力：

#### 区域选择：

<action>a<object>

<action>i<opbject>

acrtion 可以是任何命令，如d，y，v

object 可以是单词w，以空格为分割的单词W，句子s（以 . 分隔），段落p（空行分隔）,特殊字符“、‘、）、]、}

比如：字符串（map (+) (“foo”)）,光标在o处：

viw ->foo

vaw -> foo

vi”->foo

va”-> “foo”

vi) -> “foo”

va) -> (“foo”)

v2i) -> map (+) (“foo”)

v2a) -> (map (+) (“foo”))

#### 块操作：

<c-v> hjkl移动 I // [esc]

#### 自动提示：

<c-n> <c-p>

#### 宏录制：qa操作序列q，@a，@@

qaYp<c-a>q

qa开始录制

Yp复制行粘贴

<c-a> 增加1

q 停止录制

@a：在1下面写2

@@ ：重复上一个命令

100@a ：重复100次@a

#### 可视化选择

<c-v> -》选择块

J：变成一行

< 或者> ：左右缩进

= ：自动缩进

#### 分屏

分屏开启vim：

Vim -o2 f1 f2 //水平分屏

Vim -O2 f1 f2 //垂直分屏

<c-w> s //水平分隔当前文件

<c-w> v //垂直分隔当前文件

:sp f2 //水平分割打开文件

：vsp f2 //垂直分割打开文件

关闭分屏

<c-w> c //若只有一个，则不能关闭

<c-w> q //关闭当前窗口，如果剩下最后一个，则退出vim

分屏间移动

<c-w> h : 移动到左边屏幕

<c-w> j ：移动到下边屏幕

<c-w> k ：移动到上边屏幕

<c-w> l ：移动到右边屏幕

<c-w> w：移动到下一个屏幕

移动分屏：

<c-w> H/J/K/L

屏幕尺寸：

<c-w> = :所有屏幕一样高

<c-w> +: 增加高度

<c-w> -：减少高度

<c-w> </>:扩展宽度

move between windows via 'CTRL-W w' (or CTRL-W arrow key, or CTRL-W h/j/k/l for left/up/down/right), close a window via 'CTRL-W c' (or good old ':q'), make the current window the only one via 'CTRL-W o', split a window into two via 'CTRL-W s' (or 'CTRL-W v' for a vertical split), open a file in a new window via ':spl[it] filename']

### Ctags

ctags -R构造tag文件

<c-]>跳转到光标处的内容

g <c-]>选择一个位置进行跳转

<c-t>返回

用于go：

～/.ctags:

--langdef=Go

--langmap=Go:.go

--regex-Go=/func([ \t]+\([^)]+\))?[ \t]+([a-zA-Z0-9\_]+)/\2/d,func/

--regex-Go=/var[ \t]+([a-zA-Z\_][a-zA-Z0-9\_]+)/\1/d,var/

--regex-Go=/type[ \t]+([a-zA-Z\_][a-zA-Z0-9\_]+)/\1/d,type/

### cscope

<http://cscope.sourceforge.net/cscope_vim_tutorial.html>

1. download [cscope\_maps.vim](http://cscope.sourceforge.net/cscope_maps.vim), 放在.vim/plugin目录下或者将该文件source进 .vimrc。
2. cscope -R构建cscope.out.
3. vim打开文件。

出错：Error detected while processing  
/research/junaid/.vim/plugin/cscope\_maps.vim  
line 42:  
E568: duplicate cscope database not added  
E568: duplicate cscope database not added

解决：<http://alt.os.linux.narkive.com/cjX4Mw0Q/cscope-error>

删除42行所在的if段落

1. CTRL+spacebar s横向分屏，CTRL+space\*2 s纵向

find : Query for a pattern (Usage: find c|d|e|f|g|i|s|t name)

c: Find functions calling this function

d: Find functions called by this function

e: Find this egrep pattern

f: Find this file

g: Find this definition

i: Find files #including this file

s: Find this C symbol

t: Find this text string

golang支持：

～/.ctags

|  |
| --- |
| --langdef=Go  --langmap=Go:.go  --regex-Go=/func([ \t]+\([^)]+\))?[ \t]+([a-zA-Z0-9\_]+)/\2/d,func/  --regex-Go=/var[ \t]+([a-zA-Z\_][a-zA-Z0-9\_]+)/\1/d,var/  --regex-Go=/type[ \t]+([a-zA-Z\_][a-zA-Z0-9\_]+)/\1/d,type/ |

### 自动补全类名

<https://blog.csdn.net/echoes_of_rainbow/article/details/6419701>

1. ctags，创建时需要参数：

--c++-kinds=+p : Adds prototypes in the database for C/C++ files.

--fields=+iaS : Adds inheritance (i), access (a) and function signatures (S) information.

--extra=+q : Adds context to the tag name.

1. vim ~/.bash\_aliases 添加：

alias ctags='ctags -R --c++-kinds=+p --fields=+ia --extra=+q'

1. 下载omnicppcomplete插件

http://www.vim.org/scripts/script.php?script\_id=1520

直接在~/.vim 目录下解压就可以正常使用。

**ubuntu下vim调字体颜色**

<http://www.linuxidc.com/Linux/2013-11/93013.htm>

/etc/vim/vimrc：

syntax on//设置语法高亮

set tabstop=4//设置制表符宽

set softtabstop=4//设置软制表符宽

set shiftwidth=4//设置缩进

set autoindent//设置自动缩进

set cindent//设置c/c++语言的自动缩进

hi comment ctermfg=6//设置注释颜色，默认4，可选0-7，不要选择0（黑色）和7（白色）

**整体多行缩进**：按v进入visual状态，选择多行，用>或<缩进或缩出

**多行注释消除注释：**多行注释：

1. 进入命令行模式，按ctrl + v进入 visual block模式，然后按j, 或者k选中多行，把需要注释的行标记起来

2. 按大写字母I，再插入注释符，例如//

3. 按esc键就会全部注释了

取消多行注释：

1. 进入命令行模式，按ctrl + v进入 visual block模式，按字母l横向选中列的个数，例如 // 需要选中2列

2. 按字母j，或者k选中注释符号

3. 按d键就可全部取消注释

## 端口扫描

lsof -i:80

lsof -i

netstat -a

nmap 127.0.0.1

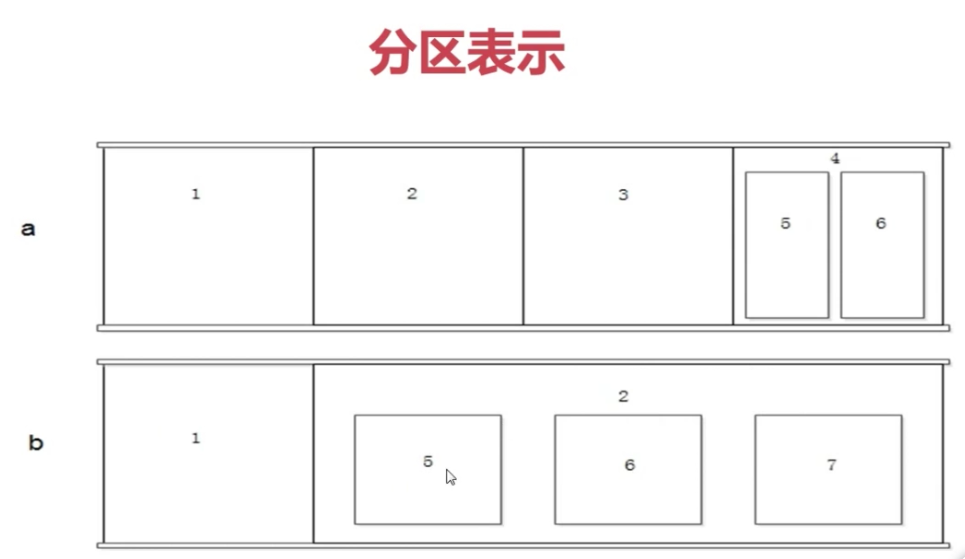
## iptables

## 分区

一个盘只能有4个主分区。为了突破限制，将最后一个分区用作扩展分区，这个扩展分区可分为多个逻辑分区：

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 7  6  5 |

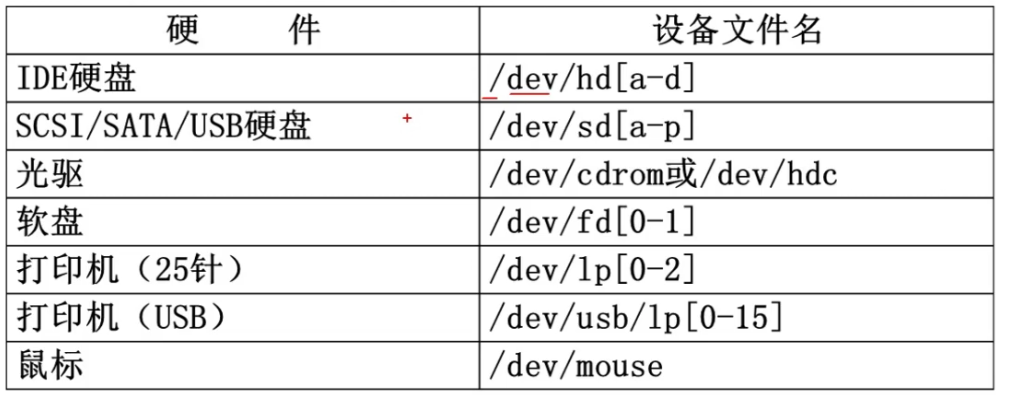
逻辑分区只能从5开始，如硬盘b，没有2，3，4主分区，逻辑分区也只能从5开始



格式化：将分区分块，并建立inode表

使用分配的分区：

1. 硬件设备文件名



分区设备文件名

/dev/hda1(ide接口硬盘)：a代表第一个硬盘，1第一个分区。133mb/s，太慢。

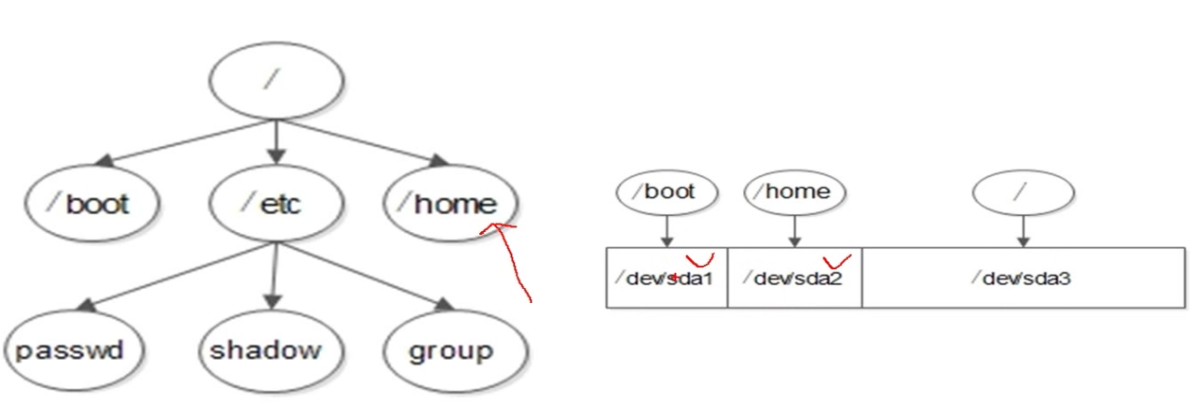
/dev/sda1(SCSI || SATA接口硬盘)：scsi<sata（500mb/s）

1. 挂载（就是将分区和目录联系起来的过程）

必须分区：/（根分区） 、 swap分区（交换分区，当内存不过用时，硬盘的swap分区当作内存使用，当内存<4g,swap大小为内存2倍，否则=内存大小）

推荐分区：/boot（启动分区，200MB）

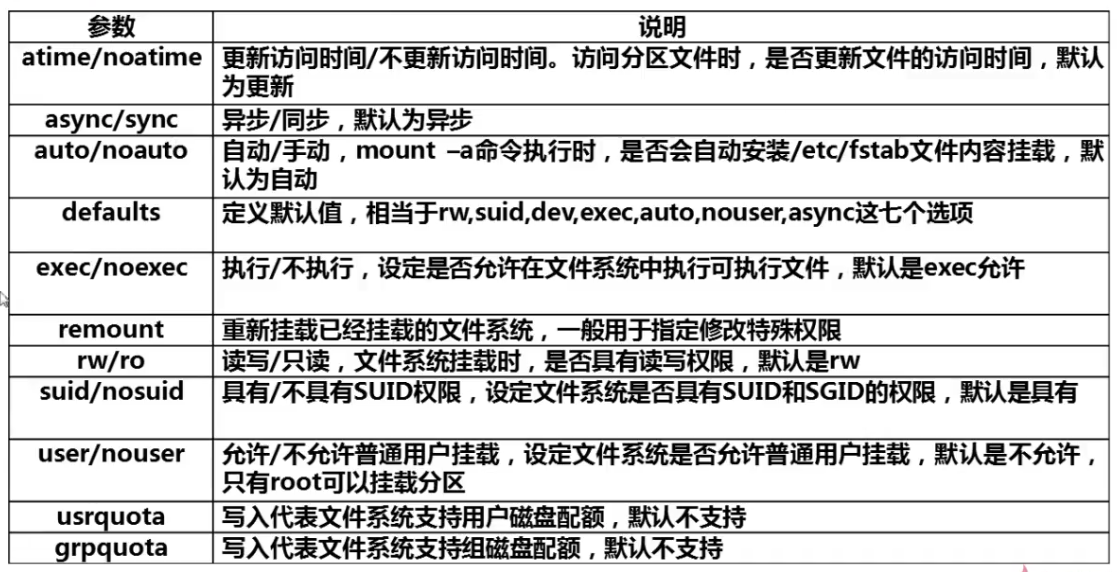
从文件系统上看，/boot在/（根目录）的二级目录，分区后实际在硬盘中的存储是分割开的。



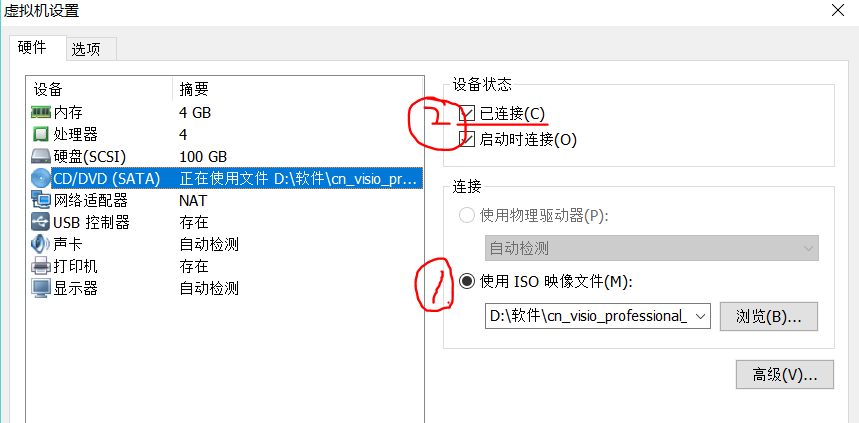
自动挂载文件：/etc/fstab

mount -t 文件类型 -o 特殊选项 设备文件名 挂载点

特殊选项：



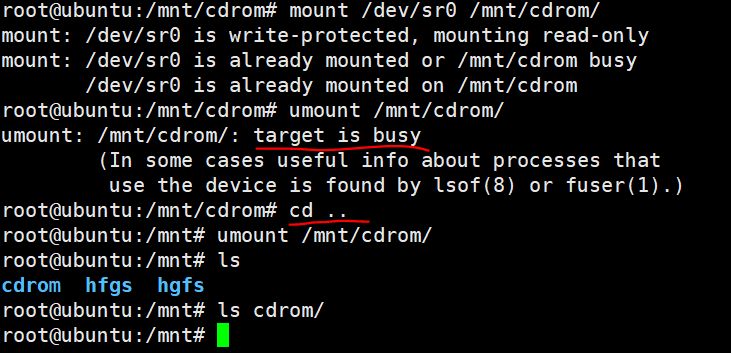
### 挂载光盘：







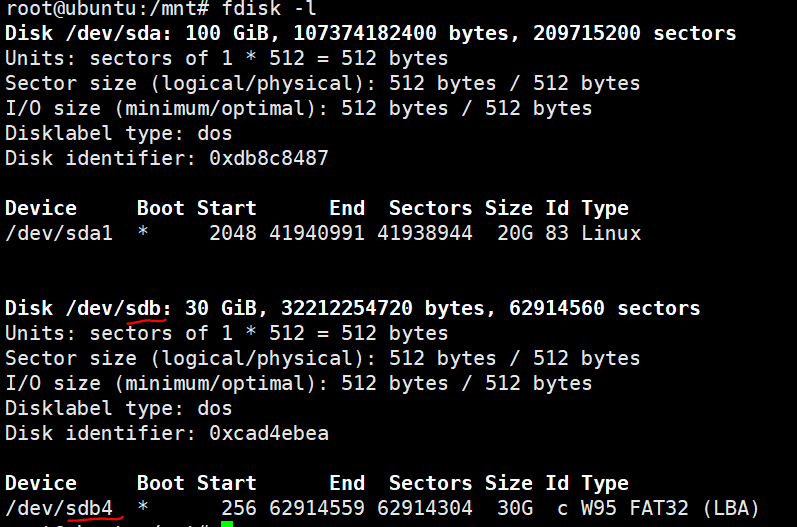
卸载光盘（必须卸载）：



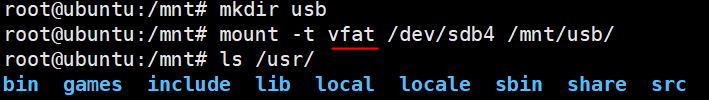
出错是因为当前目录为cdrom，退出然后umount

### 挂载u盘：

查看设备名：fdisk -l（查看系统识别的硬盘）



挂载：



卸载：

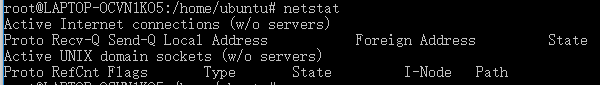


linux默认不支持ntfs文件系统

## netstat

<https://www.cnblogs.com/ggjucheng/archive/2012/01/08/2316661.html>

netstat



包括两部分：activie internet connections(w/o server)和active unix domain sockets（w/o servers）,internet connection是网络连接包括tcp或者udp，domain sockets表示unix域本地连接。

proto（协议）列，recv-Q（接收列），Send-Q（发送列），local-address（本机地址），froeign address（外部地址），state（状态，比如listen、established etc.）

at ，pre ，ls，cun

-a ：显示全部，默认不显示listen状态

-t ：tcp

-u：udp

-p ：显示程序信息

-r ：显示路由信息

-e：显示扩展信息，如uid

-l ：显示listen状态

-s：显示按照协议统计信息

-c：每隔一个固定时间执行netstat

-n：拒绝显示别名，能显示数字尽量显示数字

## sed

<https://coolshell.cn/articles/9104.html>

命令as，cd，iN，pn，hgHGx

Hold space & pattern space

## awk

<https://coolshell.cn/articles/9070.html>

Awk 单引号:和sed一样，awk命令写在单引号内

$0表示该行

$1…$n表示第n列

单引号中先是‘过滤条件 {命令}’

|  |  |
| --- | --- |
| 输出第一列和第四列 | awk '{print $1,$4}' netstat.txt |
| 格式化输出 | awk '{printf"%-10s %-10s\n",$1,$4}' netstat.txt. -号表示左对齐 |
| 第三列值为0&&第六列值为LISTEN | awk '$3==0 && $6=="LISTEN"' netstat.txt |
| 第三列大于0 | awk '$3>0' netstat.txt |
| 加上表头 | awk '$3 == 0 && $6 == "LISTEN" || NR == 1' netstat.txt |
| 指定分隔符 | awk -F: 'print $1,$4' /etc/passwd |
| 以\t作为输出分隔符 | awk -F: '{print $1,$4}' OFS="\t" /etc/passwd |
| 第六列正则匹配/FIN/ | awk '$6 ~ /FIN/ {print $4,$5,$6}' OFS='\t' netstat.txt |
| 第六列匹配FIN或者TIME | awk '$6 ~ /FIN|TIME/ {print $4,$5,$6}' OFS='\t' netstat.txt |
| 第六列匹配初WAIt之外的行 | awk '$6 !~ /WAIT/ {print $4,$5,$6}' OFS='\t' netstat.txt |
| 按照第六列拆分文件 | awk '{print > $6}' netstat.txt  awk 'NR!=1{if($6 ~ /TIME/) print>"TIME"; else if($6 ~ /FIN/) print > "FIN"; else print>"Other"}' netstat.txt |
| 统计第六列各状态出现的次数 | awk 'NR!=1{a[$6]++} END{for(i in a) print i, a[i]}' netstat.txt |
| 从文件中找出长度大于80的行 | awk 'length>80' netstat.txt |
| 打印99乘法表 | seq 9|sed 'H;g'|awk 'BEGIN{RS=""}{for(i=1;i<=NF;i++)printf("%dx%d=%d%s",i,NR,i\*NR,i==NR?"\n":"\t")}' |
| 使用外部变量 | top |gawk **-v a="$1**" '$12==a{print $1,$9,$10}' |
| awk不能实时刷新 | script -c xx.sh >>test |

内建变量：

|  |  |
| --- | --- |
| FS | 列分隔符 |
| RS | 行分隔符 |
| NR | 记录数，行数 |
| FNR | 行数，新文件从头开始计数 |
| NF | 列数 |
| OFS | 输出列分隔符 |
| ORS | 输出行分隔符 |
| FILENAME | 输入文件名称 |

## screen

<https://www.rackaid.com/blog/linux-screen-tutorial-and-how-to/>

screen：

使用一个ssh连接开多个窗口

网络中断时shell仍然存活

耗时任务不必保持该会话活动

命令：

Screen：开启screen

Ctrl+a c：创建一个窗口

Ctrl+a n：切换下一个窗口

Ctrl+a p：切换到上一个窗口

Ctrl+a d：回到shell

Screen -r ：回到sreen

Ctrl+a M：等到程序输出时提醒

Ctrl+a \_：等到程序输出完成后提醒

Ctrl+a H：打开log

## 解压

**.tar**  
解包：tar xvf FileName.tar  
打包：tar cvf FileName.tar DirName

**.gz**  
解压1：gunzip FileName.gz  
解压2：gzip -d FileName.gz  
压缩：gzip FileName  
**.tar.gz 和 .tgz**  
解压：tar zxvf FileName.tar.gz  
压缩：tar zcvf FileName.tar.gz DirName

**.bz2**  
解压1：bzip2 -d FileName.bz2  
解压2：bunzip2 FileName.bz2  
压缩： bzip2 -z FileName  
**.tar.bz2**  
解压：tar jxvf FileName.tar.bz2  
压缩：tar jcvf FileName.tar.bz2 DirName

**.bz**  
解压1：bzip2 -d FileName.bz  
解压2：bunzip2 FileName.bz  
压缩：未知  
**.tar.bz**  
解压：tar jxvf FileName.tar.bz  
压缩：未知  
---------------------------------------------  
**.Z**  
解压：uncompress FileName.Z  
压缩：compress FileName  
**.tar.Z**  
解压：tar Zxvf FileName.tar.Z  
压缩：tar Zcvf FileName.tar.Z DirName  
---------------------------------------------  
**.zip**  
解压：unzip FileName.zip  
压缩：zip FileName.zip DirName

**.rar**  
解压：rar x FileName.rar  
压缩：rar a FileName.rar DirName  
  
rar请到：<http://www.rarsoft.com/download.htm> 下载！  
解压后请将rar\_static拷贝到/usr/bin目录（其他由$PATH环境变量指定的目录也可以）：  
[root@www2 tmp]# cp rar\_static /usr/bin/rar  
---------------------------------------------  
**.lha**  
解压：lha -e FileName.lha  
压缩：lha -a FileName.lha FileName  
  
lha请到：<http://www.infor.kanazawa-it.ac.jp/~ishii/lhaunix/>下载！  
>解压后请将lha拷贝到/usr/bin目录（其他由$PATH环境变量指定的目录也可以）：  
[root@www2 tmp]# cp lha /usr/bin/  
---------------------------------------------  
**.rpm**  
解包：rpm2cpio FileName.rpm | cpio -div  
---------------------------------------------  
**.deb**  
解包：ar p FileName.deb data.tar.gz | tar zxf -  
---------------------------------------------  
**.tar .tgz .tar.gz .tar.Z .tar.bz .tar.bz2 .zip .cpio .rpm .deb .slp .arj .rar .ace .lha .lzh .lzx .lzs .arc .sda .sfx .lnx .zoo .cab .kar .cpt .pit .sit .sea**  
解压：sEx x FileName.\*  
压缩：sEx a FileName.\* FileName  
sEx只是调用相关程序，本身并无压缩、解压功能，请注意！  
sEx请到： <http://sourceforge.net/projects/sex>下载！  
解压后请将sEx拷贝到/usr/bin目录（其他由$PATH环境变量指定的目录也可以）

## update-alternatives

查看可用版本：update-alternatives --list name

安装可用版本：update-alternatives --install link name path priority

删除版本：update-alternatives --remove name path

切换版本：update-alternatives --config name

查看默认版本：name –version

## wget&curl

<https://blog.csdn.net/hzraymond/article/details/8181349>

### wget

-x按照建立和服务器一样的目录

-r递归的下载服务器目录上的文件

-t重试几次

-o 将屏幕输出重定向到文件，相当于>1.txt

-O 将下载文件重命名

-nd 所有文件放到当前目录下

-N只下载比当前文件新的文件

-P存放在指定目录下

## 更新时间

<https://blog.csdn.net/qq_20480611/article/details/50325653>

ntpdate us.pool.ntp.org

## ssh vbox下的linux

machine—》setting—〉network—》 attached to（NAT更改为bridge adapter）

中文乱码和中文不能输入？

解决：linux下vim ～/.bashrc添加：

**export** LC\_ALL=en\_US.UTF-8

**export** LANG=en\_US.UTF-8

mac下vim ~/.profile添加：

**export** LC\_ALL=en\_US.UTF-8

**export** LANG=en\_US.UTF-8

## vbox扩展硬盘

<https://blog.csdn.net/ouyang_peng/article/details/53261599>

VBoxManage list hdds

VBoxManage modifymedium 7893f7fb-5bbe-4e9e-951a-8448abb6d383 -resize 51200

分区：fdisk /dev/sda；n；w

格式化：mkfs -t ext4 /dev/sda4

挂载：mount /dev/sda4 /home/Ubuntu/sda4

开机自动挂载：vim /etc/fstab:

/dev/sda4 /home/ouyangpeng/sda4/ ext4 defaults 0 1

## 查看系统版本

lsb\_realease

dmesg|grep linux

uname -a

cat /etc/\*-release

cat /proc/version

## at命令

at 3:14 PM Mon

## nc命令

<http://www.linuxso.com/command/nc.html>

-i<延迟秒数> 设置时间间隔

-l 使用监听模式

-n 直接使用IP地址

-o<输出文件> 指定文件名称，把往来数据以16进制字码倾倒成该文件存储

-p<通信端口> 设置本机使用的通信端口

-r 乱数指定本地与远程主机的通信端口

-s<来源地址> 设置本机送出的数据包的ip地址

-u 使用udp传输协议

-v 显示指令执行过程

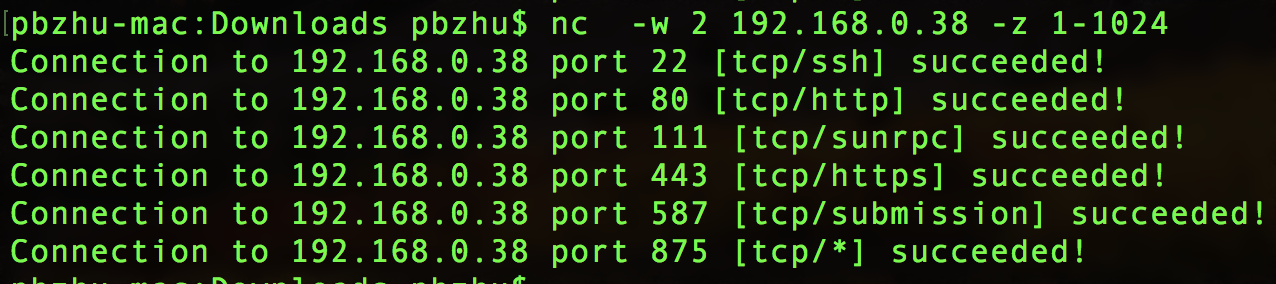
-w<超时秒数> 设置等待连接时间

-z 使用0输入/输出模式，只在扫描通信端口时使用

示例：

1. 端口扫描：

nc -v -w 2 192.168.0.38 -z 1-1024



1. 拷贝文件

IP1: nc -l 1234 > test.txt

ip2: nc ip1 1234 < 1.txt

1. 简单聊天工具

ip1: nc -l 1234

ip2: nc ip1 1234

双方就可以相互交流了

## 共享库

缺少共享库时，将该so文件放置在/usr/lib下，并执行ldconfig