Linux 开机启动(bootstrap)：

BIOS -> MBR -> boot loader -> kernel -> init process -> login

BIOS: Basic Input/Output System, 这一程序通常知道一些直接连接在主板上的硬件(硬盘，网络接口，键盘，串口，并口)。现在大部分的BIOS允许你从软盘、光盘或者硬盘中选择一个来启动计算机。

MBR: master boot record MBR会告诉电脑从该设备的某一个分区(partition)来装载引导加载程序(boot loader)

Boot loader: Boot loader储存有操作系统(OS)的相关信息，比如操作系统名称，操作系统内核 (kernel)所在位置等。常用的boot loader有GRUB和LILO。

Knernel: kernel实际上是一个用来操作计算机的程序，它是计算机操作系统的内核，主要的任务是管理计算机的硬件资源，充当软件和硬件的接口。操作系统上的任何操作都要通过kernel传达给硬件。Windows和Linux各自有自己kernel。

Init process: init会运行一系列的初始脚本(startup scripts)，这些脚本是Linux中常见的shell scripts。这些脚本执行如下功能：

设置计算机名称，时区，检测文件系统，挂载硬盘，清空临时文件，设置网络……

当这些初始脚本，操作系统已经完全准备好了，只是，还没有人可以登录.

**Linux 文件信息：**



**Linux 多线程和多进程：**

同步(synchronization)是指在一定的时间内只允许某一个线程访问某个资源 。而在此时间内，不允许其它的线程访问该资源。我们可以通过**互斥锁(mutex)，条件变量(condition variable)和读写锁(reader-writer lock)**来同步资源。

互斥锁：

互斥锁是一个特殊的变量，它有锁上(lock)和打开(unlock)两个状态。互斥锁一般被设置成全局变量。打开的互斥锁可以由某个线程获得。一旦获得，这个互斥锁会锁上，此后只有该线程有权打开。其它想要获得互斥锁的线程，会等待直到互斥锁再次打开的时候。

条件变量：

条件变量是另一种常用的变量。它也常常被保存为全局变量，并和互斥锁合作。

读写锁：

读写锁与互斥锁非常相似。r、RW lock有三种状态: **共享读取锁(shared-read)**, **互斥写入锁(exclusive-write lock), 打开(unlock)**。后两种状态与之前的互斥锁两种状态完全相同

如果被一个线程获得R锁，RW lock可以被其它线程继续获得R锁，而不必等待该线程释放R锁。但是，如果此时有其它线程想要获得W锁，它必须等到所有持有共享读取锁的线程释放掉各自的R锁。

如果一个锁被一个线程获得W锁，那么其它线程，无论是想要获取R锁还是W锁，都必须等待该线程释放W锁。

用户的账号信息存储在etc/passwd里面，其相应的密码存储在etc/shadow。

软链接和硬链接：

硬链接是指向相同inode和data block块的不同别名，换言之我们创建一个文件的应链接就创建了一个该文件的别名。

**ln –P 要创建的source文件 硬链接的名字**

软链接类似于快捷方式，其本身也是一个文件，但是其data block 存储的是source文件的inode 号以及相关的data block ,若source 文件删除了，则该软链接变成死链接了

**ln –s 要创建的source文件 软链接的名字**

bash中的环境变量：

1:系统默认的环境的环境变量全是大写，如PATH,HOME,

2: 显示环境变量，用echo, 如 echo $PATH, 注意$不能丢

设置变量注意：

1:用等号连结：myname=xxx

2:等号两边不能直接接空格

3:变量名只能是英文字母和数字，但变量名开头不能是字母。

4:“”和‘’，“”里面的变量内容保持特殊字符的特性，如$

‘ ’里面的变量内容则是一般的字符，即纯文本。

5:跳脱字符\,可以将特殊符号$,\等变成一般字符。

6:定义一个变量的时候要借用其他指令的信息，可以用反单引号（1旁边的键） 或者 （$(指令)），例如： version=$(uname -r)

7: 扩增变量内容，可用“$变量名称”或者${变量} + 累加内容

例如：PATH=”$PATH”:/home/bin

Linux下开发java和c++程序：

java:

1:javac hello.java =>编译

2 :java hello

c++:

1: gcc –o myname hello.cpp => 编译，生成myname文件。

2: ./myname

改变shell类型：

chsh –s /bin/bash

关于ip:

ifconfig =>在linux 和unix下，而在windows下是ipconfig

ping 域名 =>测试两个ip是否畅通。Ping 127.0.0.1 代表ping自己，即是一个回路，可以测试自己的网卡是否工作。

traceroute 域名 => 查看中间的路由ip等信息。

关于设置ip等信息－三种方法：

1:setup

配置完后，需要执行 /etc/rc.d/init.d/network restart 刚才的配置才生效。

2:ifconfig eth0 x.x.x.x 对网卡进行设置(在mac 下eth0是en0)

ifconfig eth0 network x.x.x.x 对子网掩码进行设置。

注意：注意这种设置的方法可以及时生效，但是重新启动后又是原来的ip。

3:根本的方法，(linux下所有的设备都是一个文件)

修改/etc/sysconfig/network-scripts/ifcfg-eth0 这个文件里各个属性，如ip,子网掩码。

然后再运行 /etc/rc.d/init.d/network restart

关于rpm包的管理：

rpm

关于任务调度命令：

chrontab –e 设置任务

结构如下：

0 2 \* \* \* +命令 =>每天凌晨两点执行命令

\* \* \* \* \* +命令 =>每分钟都执行

分钟(0-59) 小时(0-23) 日期(1-31) 月(1-12) 星期几（0-6）

怎样调度多个任务：

1:在crontab +e写入多个命令。（比较麻烦）

2:可以把所有的任务写入到一个可执行文件（shell 编程），然后在crontab里面加入那个文件。如：

1:建立一个mytask.sh:写入date >> ~/desktop/mydate1

2:在crontab 里面加入 \* \* \* \* \* ~/desktop/mytask.sh(表示每分钟执行这个shell文件)

如何终止任务：

crontab –r (即删除了所有的任务)

列出所有的任务列表：

crontab –l

关于进程：

kill :杀死进程 (kill -9 强制杀死这个进程)(killall 杀死此进程和其子进程)

ps： 列出进程

top: 动态监控进程。