

Linux-高阶知识

目录

1、Httpd 服务	1
2、系统管理.....	2
2.1、挂载外部设备.....	2
2.2、统计文件或者文件夹大小.....	3
2.3、系统服务管理.....	4
2.4、修改系统时间.....	4
3、软件安装.....	5
4、配置本地 YUM 源.....	6
5、进程管理.....	8
5.1、进程概念.....	8
5.2、进程状态.....	8
5.4、进程管理常用操作.....	8
5.4.1、PS 命令.....	8
5.4.2、Kill 命令.....	10
5.4.3、进程切换.....	10
5.4.4、top 命令.....	10
5.4.4、pstree 命令.....	11
5.4.5、JPS 命令.....	11
6、计划任务.....	11
6.1、概念.....	11
6.2、命令服务管理 crontab.....	12
6.4、crontab 功能使用.....	12
7、克隆虚拟机.....	14
8、SSH 免密登录配置.....	14

1、Httpd 服务

- 1、先检查本机的 httpd 服务是否开启，使用命令：

service --status-all | grep httpd

```
[root@localhost ~]# service --status-all | grep httpd
httpd is stopped
grep: /proc/fs/nfsd/portlist: No such file or directory
[root@localhost ~]#
```

- 2、开启 httpd 服务

```
grep: /proc/fs/nfsd/portlist: No such file or directory
[root@localhost .ssh]# service httpd start
Starting httpd: httpd: Could not reliably determine the serve
for ServerName
[ OK ]
[root@localhost .ssh]# service --status-all | grep httpd
httpd (pid 27228) is running...
grep: /proc/fs/nfsd/portlist: No such file or directory
[root@localhost .ssh]#
```

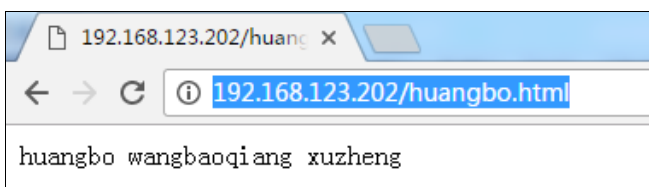
为确保是否开启成功，请再重新检查一次

- 我们进入到/var/www/html 目录下，新建一个 huangbo.html 文件，往文件里加入一个字符串“huangbo wangbaoqiang xuzheng”，然后保存

```
cd /var/www/html
```

```
vi huangbo.html
```

- 在 windows 系统中，打开浏览器，在 url 地址栏输入：<http://192.168.123.202/huangbo.html> 看看是不是能看到我们往里加入到文件当中的字符串：huangbo wangbaoqiang xuzheng，如果看到如下效果，则证明 httpd 服务使用成功



- 在我们进入到/var/www/html 目录下，新建一个 mingxing 文件夹，往文件夹新建一个文件 wangbaoqiang.jpg，然后保存，访问 <http://192.168.123.202/mingxing>



2、系统管理

2.1、挂载外部设备

挂载是一个非常重要的功能，使用非常频繁。它指将一个设备（通常是存储设备，可以挂载光盘、硬盘、磁带、光盘镜像文件等）挂接到一个已存在的目录上（这个目录可以为空，但挂载后这个目录下以前的内容将不可用）。

需要理解的是，Linux 操作系统将所有的设备都看作文件，它将整个计算机的资源都整合成一个大的文件目录。我们要访问存储设备中的文件，必须将文件所在的分区挂载到一个已存在的目录上，然后通过访问这个目录来访问存储设备。

挂载需要有**挂载源**和**挂载点**

1、挂载光驱

在常用挂载目录/mnt 下创建一个目录 cdrom 用来进行挂载：

mkdir /mnt/cdrom

将挂载源设备/dev/cdrom 挂载到该挂载点（/mnt/cdrom）上：

mount -t iso9660 -o ro /dev/cdrom /mnt/cdrom/

注：-t：文件系统类型，iso9660 表示光盘或者光盘镜像

-o：挂载方式，ro 表示以只读方式，loop 表示把挂载的设备当做一个磁盘分区

2、挂载光盘镜像文件（.iso 文件）

我们准备一个镜像文件：CentOS-6.7-x86_64-bin-DVD1.iso

再准备一个挂载目录：mkdir /mnt/centos

mount -t iso9660 -o loop /root/CentOS-6.7-x86_64-bin-DVD1.iso /mnt/centos

3、挂载的资源在重启后即失效，需要重新挂载。要想自动挂载，可以将挂载信息设置到 **/etc/fstab** 配置文件中，如下：

/dev/cdrom	/mnt/cdrom	iso9660	defaults	0 0
/root/CentOS-6.7-x86_64-bin-DVD1.iso	/mnt/centos	iso9660	defaults,ro,loop	0 0

第一列：/dev/cdrom：挂载源，要挂在的磁盘分区或者存储设备，或者文件系统

第二列：/mnt/cdrom：挂载点，挂载位置

第三列：iso9660：文件系统类型，ext4 文件系统，swap 表示交换分区，.....

第四列：defaults：挂载参数

第五列：0：表示 dump 选项，dump 工具通过这个选项位置上的数字来决定文件系统是否需要备份。如果是 0，dump 就会被忽略，事实上大多数的 dump 设置都是 0

第六列：0：读取文件系统时候的检查顺序，允许的数字是 0,1,2，根目录应当获得最高的优先权 1，其它所有需要被检查的设备设置为 2，0 表示不检查

4、卸载，使用 umount 命令

umount /mnt/cdrom

2.2、统计文件或者文件夹大小

1、存储空间查看

df -h

2、统计指定路径下的所有子目录和文件的大小

```
du -sh /mnt/cdrom/packages
```

2.3、系统服务管理

1、跟服务相关的基本使用命令：

```
service --status-all # 查看系统所有的后台服务进程
service sshd status  # 查看指定的后台服务进程的状态
service sshd stop    # 停止
service sshd start   # 开始
service sshd restart  # 重启
```

2、配置后台服务进程的开机自启

```
chkconfig httpd on ## 让 httpd 服务开机自启
chkconfig httpd off ## 让 httpd 服务开机不要自启
```

```
[root@hadoop01 mnt]# chkconfig httpd off
[root@hadoop01 mnt]# chkconfig --list | grep httpd
httpd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@hadoop01 mnt]# chkconfig --level 35 httpd on
[root@hadoop01 mnt]# chkconfig --list | grep httpd
httpd          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

2.4、修改系统时间

1、查看系统当前时间和时区

查看时间：下面两种都 OK

```
[root@hadoop01 ~]# date -R
[root@hadoop01 ~]# date
[root@hadoop01 ~]# date '+%Y-%m-%d %H:%M:%S'
```

```
[root@hadoop01 ~]# date -R
Sun, 25 Dec 2016 19:46:02 +0800
[root@hadoop01 ~]#
```

查看时区：cat /etc/sysconfig/clock

```
[hadoop@hadoop02 conf]$ cat /etc/sysconfig/clock
ZONE="America/Los_Angeles"
```

2、CentOS 中时区是以文件形式存在，当前正在使用的时区文件位于

/etc/localtime

其他时区文件则位于

/usr/share/zoneinfo

其中中国时区使用

/usr/share/zoneinfo/Asia/Shanghai

则，更改时区的话，直接使用如下命令就 OK

```
cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

PS: 如果没有 Asia/Shanghai 时区文件, 请使用 `tzselect` 命令去生成时区文件, 生成好的时区文件就在 `/usr/share/zoneinfo` 目录下

3、我们一般使用“date -s”命令来修改系统时间

```
[root@hadoop01 ~]# date -s 12/25/2016
[root@hadoop01 ~]# date -s 19:57:30
[root@hadoop01 ~]# date -s "2016-12-25 19:57:30"
```

```
[root@hadoop01 ~]# date -s 12/25/2016
Sun Dec 25 00:00:00 CST 2016
[root@hadoop01 ~]# date -s 19:57:30
Sun Dec 25 19:57:30 CST 2016
[root@hadoop01 ~]# date -s "2016-12-25 19:57:30"
Sun Dec 25 19:57:30 CST 2016
[root@hadoop01 ~]#
```

4、每次都手动来修改, 比较麻烦, 这里教大家一个简单的方法, 用来同步网络时间。使用命令:

```
[root@hadoop01 ~]# ntpdate time.nist.gov
[root@hadoop01 ~]# ntpdate 202.120.2.101
```

```
[root@hadoop01 ~]# date
Sun Dec 25 21:54:36 CST 2016
[root@hadoop01 ~]# ntpdate time.nist.gov
25 Dec 21:56:47 ntpdate[12333]: step time server 132.163.4.101 offset 119.809453 sec
[root@hadoop01 ~]# date
Sun Dec 25 21:57:34 CST 2016
[root@hadoop01 ~]#
```

上海交通大学网络中心 NTP 服务器地址: 202.120.2.101 (`ntp.sjtu.edu.cn`)

中国国家授时中心服务器地址: `cn.pool.ntp.org`

若以上提供的网络时间服务器不可用, 请自行上网寻找可用的网络时间服务器

5、现在告诉大家一个一劳永逸的方法

1、使用命令: `crontab -e`

```
[root@hadoop01 ~]# crontab -e
```

`crontab` 的使用后面会有详细讲解

2、然后往里加入一行内容

```
*/10 * * * * ntpdate 202.120.2.101
```

上面的配置表示, 每隔十分钟从 202.120.2.101 该时间服务器同步一次时间。

3、保存退出即可

3、软件安装

1、二进制发布包

软件已经针对具体平台编译打包发布, 只要解压, 修改配置即可

演示过程见 JDK 安装文档“[资料-jdk 安装](#)”
或者参考 Linux 系统中的 Tomcat 的安装

2、RPM 发布包

软件已经按照 RedHat 的包管理工具规范 RPM 进行打包发布，需要获取到相应的软件 RPM 发布包，然后用 RPM 命令进行安装

演示过程见 MySQL 安装文档“[资料-MySQL 安装](#)”

3、Yum 在线安装

Yum (**Y**ellow **d**og **U**pdater, **M**odified) 是一个在 CentOS 和 RedHat 以及 SUSE 中的 Shell 前端软件包管理器。比如软件已经以 RPM 规范打包，但发布在了网络上的一些服务器上，可用 yum 在线安装服务器上存在的 rpm 软件，并且会自动解决软件安装过程中的库依赖问题（注：类似于 maven），yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令，而且命令简洁而又好记

yum 仓库服务器本质上就是一台 http 服务器，服务器的目录中放置了 rpm 包，及 rpm 包的索引信息文件，即可为 yum 客户端提供 rpm 文件下载

安装示例：yum install nc

yum 命令工具使用举例

yum update 升级系统

yum install ~ 安装指定软件包

yum update ~ 升级指定软件包

yum remove ~ 卸载指定软件

4、源码编译安装

软件以源码工程的形式发布，需要获取到源码工程后用相应开发工具进行编译打包部署

安装示例：资料-Redis 的安装

4、配置本地 YUM 源

1、准备一个 http 服务器，这里我们已经配置好了 httpd 的服务，如果系统不自带，请自行安装，可以使用命令在线安装：

yum install -y httpd

2、安装好了，或者重新安装完成后，检查 httpd 的运行状态

service httpd status

3、在 httpd 服务的 web 目录中放入 rpm 库，

比如，将 centos 安装光盘中的 rpm 库放入：

mkdir /var/www/html/centos

cp -r /mnt/cdrom/* /var/www/html/centos/

注：上面这种方式比较浪费空间，可以不用拷贝，其实只要创建一个软连接即可

ln -s /mnt/cdrom /var/www/html/centos

- 4、用浏览器访问一下 httpd 服务，看看是不是能看到这些东西

地址：<http://192.168.123.202/centos>

- 5、配置 yum 客户端 repo 地址文件

将本地 http 服务器加入 repo 地址

首先，将内置的源全部 disable 掉

cd /etc/yum.repos.d/

rename .repo .repo.bak *

然后，新建一个 repo 源，

vi CentOS-Local.repo

```
[CentOS-Local]
name=CentOS-Local
baseurl=http://192.168.123.202/centos/cdrom0
http://192.168.123.202/centos/cdrom1
gpgcheck=0
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

接下来，我们检查我们自定义的源是否生效，使用命令：**yum repolist**

```
[root@localhost yum.repos.d]# yum repolist
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
CentOS-Local
CentOS-Local/primary_db
repo id                               repo name
CentOS-Local
repolist: 6,575
[root@localhost yum.repos.d]#
```

- 6、再接下来，就可以使用 yum 像从公网安装软件一样在内网服务器上下载软件进行安装了
yum install -y 你要的软件

- 7、补充：

国内有名的 Yum 源：

网易：

CentOS5：<http://mirrors.163.com/.help/CentOS5-Base-163.repo>

CentOS6：<http://mirrors.163.com/.help/CentOS6-Base-163.repo>

CentOS7：<http://mirrors.163.com/.help/CentOS7-Base-163.repo>

5、进程管理

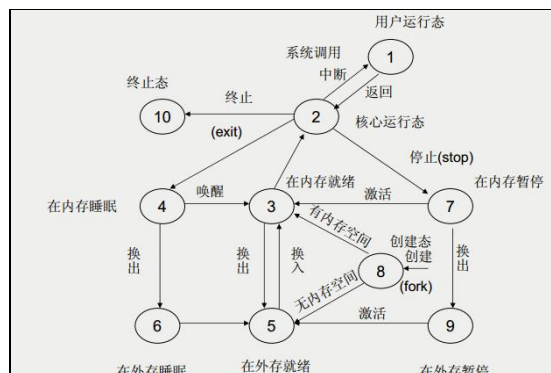
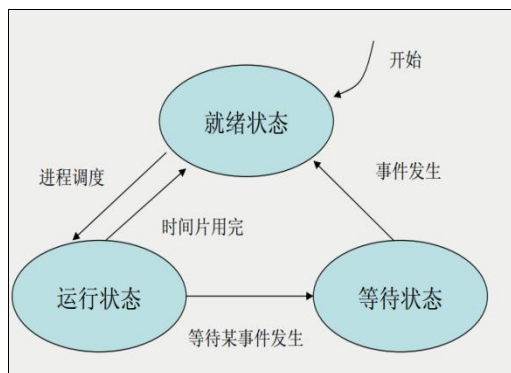
5.1、进程概念

进程是操作系统中非常重要的一个概念，进程是程序的执行过程，相对于程序，进程是动态的，在 linux 系统中，它与用户权限相关，程序与进程并没有一一对应，一个程序可能对应多个进程

子进程与父进程：一个进程产生另外一个进程，产生的进程称为子进程，生成另外一个进程的进程称为父进程

5.2、进程状态

进程状态总体来看主要包括三种，分别是就绪状态、运行状态、等待状态，就绪状态指的是进程除了 CPU 资源，其他运行资源都已获得，当就就绪状态的进程得到 CPU 资源时，则进入进程进入运行状态，运行状态的进程在时间片用完后会重新转换成就绪状态，如果在运行时进程遇到 IO 等阻塞操作，则进程会进入等待状态，等待状态的进程当阻塞操作完成后，进程将进入就绪状态



5.4、进程管理常用操作

5.4.1、PS 命令

ps 命令用于报告当前系统的进程状态。可以搭配 kill 指令随时中断、删除不必要的程序。ps 命令是最基本同时也是非常强大的进程查看命令，使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等，总之大部分信息都是可以通过执行该命令得到的。

ps 命令常用选项包括：

- 1、a 显示所有用户的进程
- 2、u 显示用户名和启动时间
- 3、x 显示所有进程，包括没有控制终端的进程

- 4、e 显示所有进程，包括没有控制终端的进程，较 x 选项，信息更为简略
- 5、l 显示进程详细信息，按长格式显示

常用组合

ps -au	显示所有用户进程，并给出用户名和启动时间等详细信息
ps -aux	显示所有用户进程，包括没有控制终端的进程，并给出用户和启动埋单等详细信息
ps -el	按长格式显示进程详细信息

以上命令执行结果中可能出现的一个字段含义：

USER: 进程所有者
 PID: 进程号
 PPID: 进程的父进程 ID
 %CPU: CPU 占用率
 C: 进程的 CPU 占用率
 %MEM: 内存占用率
 VSZ: 表示如果一个程序完全驻留在内存的话需要占用多少内存空间;
 RSS: 指明了当前实际占用了多少内存;
 TTY: 终端的次要装置号码 (minor device number of tty)
 F: 进程的标志
 S: 进程的状态
 STAT: 该进程的状态，有以下值
 D: 不可中断的静止
 R: 正在执行中
 S: 静止状态
 T: 暂停执行
 Z: 不存在但暂时无法消除
 W: 没有足够的记忆体分页可分配
 <: 高优先序的进程
 N: 低优先序的进程
 L: 有记忆体分页分配并锁在记忆体内
 PRI: 进程的优先权
 NI: 进程的 Nice 值
 ADDR: 进程的地址空间
 SZ: 进程占用内存的大小
 WCHAN: 进程当前是否在运行
 TTY: 进程所属终端
 START: 进程开始时间
 TIME: 执行的运行时间
 COMMAND: 所执行的指令
 CMD: 进程的命令

5.4.2、Kill 命令

有时候某个进程可能会长期占用 CPU 资源或无法正常执行或超出运行时间等，此时可能希望人工干预直接将进程杀死，这时候 kill 命令可以派上用场

- 1、kill pid 直接杀死进程，但不能保证一定能杀死
- 2、kill -9 pid 强制杀死进程

5.4.3、进程切换

前台进程指的是进程在执行时会将命令行阻塞，直到进程执行完毕；后台进程指的是进程在执行时不会阻塞当前命令行，而是在系统后台执行

- 1、ctrl + c 终止进程
- 2、ctrl + z 挂起进程
- 3、fg 命令将进程转换到前台执行
- 4、bg 命令将进程转换到后台执行
- 5、jobs 命令查看任务

5.4.4、top 命令

Top 命令可以定期显示所有正在运行和实际运行并且更新到列表中，它显示出 CPU 的使用、内存的使用、交换内存、缓存大小、缓冲区大小、过程控制、用户和更多命令。它也会显示内存和 CPU 使用率过高的正在运行的进程。

```
top - 15:48:27 up 6:59, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 96 total, 1 running, 95 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3%us, 0.3%sy, 0.0%ni, 99.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1004768k total, 765848k used, 238920k free, 24412k buffers
Swap: 2031612k total, 0k used, 2031612k free, 143228k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4548	hadoop	20	0	15032	1192	924	R	0.3	0.1	0:00.01	top
1	root	20	0	19364	1544	1228	S	0.0	0.2	0:01.58	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.42	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.07	watchdog/0
7	root	20	0	0	0	0	S	0.0	0.0	0:24.29	events/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events_long/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events_power_ef
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cgroup
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
16	root	20	0	0	0	0	S	0.0	0.0	0:00.22	sync_supers
17	root	20	0	0	0	0	S	0.0	0.0	0:00.22	bdi-default
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kintegrityd/0

按 q 键退出查看

5.4.4、pstree 命令

将进程间的关系以树结构的形式展示，能清楚看各进程之间的父子关系

pstree : 以树状形式显示进程

```
[hadoop@hadoop02 ~]$ pstree
init--ManagementAgent--2*[{ManagementAgen}]
      |
      |--NetworkManager
      |--VGAuthService
      |--abrttd
      |--acpid
      |--atd
      |--auditd--{auditd}
      |--bluetoothd
      |--console-kit-dae--63*[{console-kit-da}]
      |--crond
      |--cupsd
      |--dbus-daemon--{dbus-daemon}
      |--haldaemon--haldaemon-runner--haldaemon-addon-acpi
      |               |               |
      |               |               |--haldaemon-addon-input
      |               |               |--haldaemon-addon-rfki
      |               |
      |               |--{haldaemon}
      |
      |--6*[{mingetty}]
      |--modem-manager
```

pstree -p : 以树状形式显示进程，并且显示进程号

```
[hadoop@hadoop04 data]$ pstree -p
init(1)--ManagementAgent(1687)--{ManagementAgen}(1700)
      |                               |
      |                               |--{ManagementAgen}(1702)
      |
      |--NetworkManager(1992)
      |--VGAuthService(1622)
      |--abrttd(2415)
      |--acpid(2051)
      |--atd(2462)
      |--auditd(1893)--{auditd}(1894)
      |--bluetoothd(2135)
      |--console-kit-dae(5894)--{console-kit-da}(5895)
      |                           |
      |                           |--{console-kit-da}(5897)
      |                           |--{console-kit-da}(5898)
      |                           |--{console-kit-da}(5899)
      |                           |--{console-kit-da}(5900)
```

5.4.5、JPS 命令

JPS 命令是 JDK 提供的一个检查系统是否启动了 JVM 进程的一个进程。不是 linux 系统自带的。主要任务就是用来检查 java 进程的。

6、计划任务

6.1、概念

计划任务在 Linux 的体现主要分为 at 和 crontab，其中：

at: 通过 at 命令安排任务在某一时刻执行一次

crontab: 通过 **crontab** 命令，我们可以在固定的间隔时间执行指定的系统指令或 **shell script** 脚本。时间间隔的单位可以是分钟、小时、日、月、周及以上的任何组合。这个命令非常适合周期性的日志分析或数据备份等工作。

6.2、命令服务管理 **crontab**

在 CentOS 系统上，**crontab** 服务的名称叫做 **crond**

1、安装

```
yum -y install crontabs
```

2、服务操作说明

```
service crond start    ## 启动服务
service crond stop     ## 关闭服务
service crond restart  ## 重启服务
service crond reload    ## 重新载入配置
service crond status    ## 服务状态
```

查看 **crontab** 服务是否已设置为开机启动，执行命令：

```
chkconfig --list
```

加入开机自动启动：

```
chkconfig --level 35 crond on
```

6.4、**crontab** 功能使用

1、命令格式

```
crontab [-u user] file
crontab [-u user] [-e | -l | -r ]
```

参数说明：

-u user: 用来设定某个用户的 **crontab** 服务，例如，“-u ixdba”表示设定 ixdba 用户的 **crontab** 服务，此参数一般有 root 用户来运行。

file: file 是命令文件的名字,表示将 file 做为 **crontab** 的任务列表文件并载入 **crontab**。

-e: 编辑某个用户的 **crontab** 文件内容。如果不指定用户，则表示编辑当前用户的 **crontab** 文件。

-l: 显示某个用户的 **crontab** 文件内容，如果不指定用户，则表示显示当前用户的 **crontab** 文件内容。

-r: 删除定时任务配置，从/var/spool/cron 目录中删除某个用户的 **crontab** 文件，如果不指定用户，则默认删除当前用户的 **crontab** 文件。

-i: 在删除用户的 **crontab** 文件时给确认提示。

命令示例：

`crontab file [-u user]` ## 用指定的文件替代目前的 `crontab`。

必须掌握：

`crontab -l [-u user]` ## 列出用户目前的 `crontab`。

`crontab -e [-u user]` ## 编辑用户目前的 `crontab`。

补充：通过 `crontab` 添加的计划任务都会存储在 `/var/spool/cron/` 目录里

2、配置说明

基本格式：

*** * * * * command**

分 时 日 月 周 命令

第 1 列表示分钟 1~59 每分钟用*或者 */1 表示

第 2 列表示小时 0~23 (0 表示 0 点) 7-9 表示：8 点到 10 点之间

第 3 列表示日期 1~31

第 4 列表示月份 1~12

第 5 列标识号星期 0~6 (0 表示星期天)

第 6 列要运行的命令

3、配置示例

`*/* * * * * date >> /root/date.txt`

上面的例子表示每分钟执行一次 `date` 命令

`30 21 * * * /usr/local/etc/rc.d/httpd restart`

上面的例子表示每晚的 21:30 重启 `apache`

`45 4 1,10,22 * * /usr/local/etc/rc.d/httpd restart`

上面的例子表示每月 1、10、22 日的 4:45 重启 `apache`

`10 1 * * 6,0 /usr/local/etc/rc.d/httpd restart`

上面的例子表示每周六、周日的 1:10 重启 `apache`

`0,30 18-23 * * * /usr/local/etc/rc.d/httpd restart`

上面的例子表示在每天 18:00 至 23:00 之间每隔 30 分钟重启 `apache`

`0 23 * * 6 /usr/local/etc/rc.d/httpd restart`

上面的例子表示每星期六的 11:00 pm 重启 `apache`

`* */1 * * * /usr/local/etc/rc.d/httpd restart`

上面的例子每一小时重启 `apache`

`* 23-7/1 * * * /usr/local/etc/rc.d/httpd restart`

上面的例子晚上 11 点到早上 7 点之间，每隔一小时重启 `apache`

```
0 11 4 * mon-wed /usr/local/etc/rc.d/httpd restart
```

上面的例子每月的 4 号与每周一到周三的 11 点重启 apache

```
0 4 1 jan * /usr/local/etc/rc.d/httpd restart
```

上面的例子一月一号的 4 点重启 apache

7、克隆虚拟机

见文档“[资料-vmware 克隆虚拟机](#)”

8、SSH 免密登录配置

1、SSH 概念

SSH 为 Secure Shell（安全外壳协议）的缩写，简单说，SSH 只是一种网络协议，用于计算机之间的加密登录，很多 ftp、pop 和 telnet 在本质上都是不安全的，因为它们在网上用明文传送口令和数据，别有用心的非常容易就可以截获这些口令和数据。而 SSH 就是专为远程登录会话和其他网络服务提供安全性的协议。

SSH 是由客户端和服务端的软件组成的

服务端是一个守护进程(sshd)，他在后台运行并响应来自客户端的连接请求。

客户端包含 ssh 程序以及像 scp（远程拷贝）、slogin（远程登陆）、sftp（安全文件传输）等其他的应用程序。

2、SSH 认证机制

从客户端来看，SSH 提供两种级别的安全验证。

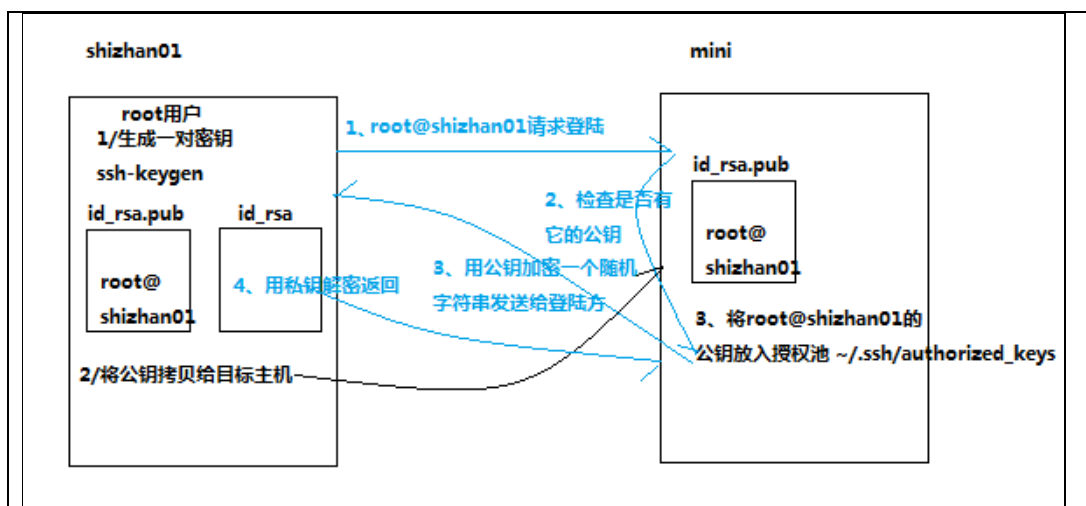
第一种级别（基于口令的安全验证）

只要你知道自己帐号和口令，就可以登录到远程主机。

第二种级别（基于密钥的安全验证）

需要依靠密钥，也就是你必须为自己创建一对密钥，并把公用密钥放在需要访问的服务器上。如果你要连接到 SSH 服务器上，客户端软件就会向服务器发出请求，请求用你的密钥进行安全验证。服务器收到请求之后，先在该服务器上你的主目录下寻找你的公用密钥，然后把它和你发送过来的公用密钥进行比较。如果两个密钥一致，服务器就用公用密钥加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用你的私人密钥解密再把它发送给服务器。

原理图：



3、配置 SSH 认证机制

假如 A 要登录 B:

在 A 上进行操作:

- 1、首先生成密钥对，使用命令:

ssh-keygen

或者

ssh-keygen -t rsa

上面一种是它的简写形式，提示要输入信息的时候不用输入任何东西，直接回车三次就 OK

上述命令执行完成之后，会生成这么两个文件，其中 **id_rsa.pub** 是公钥

```
[root@localhost .ssh]# ll
total 8
-rw-----. 1 root root 1679 Dec 27 01:55 id_rsa
-rw-r--r--. 1 root root 408 Dec 27 01:55 id_rsa.pub
```

- 2、将 A 自己的公钥放置到授权列表文件 **authorized_keys** 中

cat id_rsa.pub > authorized_keys

或者

cp id_rsa.pub authorized_keys

- 3、将得到的 **authorized_keys** 发送给 B 机器对应的文件夹下

scp -r authorized_keys root@192.168.123.202:\$PWD

或者

scp -r authorized_keys root@192.168.123.202:/root/.ssh/

这是一个远程拷贝命令，意思是把 **authorized_keys** 拷贝到 192.168.123.202 这台机器对应目录下

- 4、这样我们的机器 A 就能访问到机器 B 了。验证方法使用命令:

ssh 192.168.123.202

重点理解之处: A 将公钥发给 B，不是 B 可以访问 A，而是 A 可以访问 B，所以任何一

台机器想要能访问其他机器，那么就把自己的公钥内容发送到别的机器的 `authorized_keys` 中去，记住，是追加，不是覆盖