## 1 定时任务crond介绍

### 1．1 crond是什么？

Crond是linux系统中用来定期执行命令或指定程序任务的一种服务或软件。一般情况下，我们安装完CentOS5/6linux操作系统之后，默认便会启动crond任务调用服务，在我们前面的系统安装及开机优化的设置中，我们也设置保留了crond开机自启动。Crond服务会定期（默认**每分钟**检查一次）检查系统中是否有要执行的任务工作。如果有，便会根据预先设定的定时任务规则自动执行该定时任务工作。这个crond定时任务服务就相当于我们平时早起使用的闹钟一样。

特殊需求：crond服务搞不定了，一般工作中写脚本守护进程执行，

[root@oldboylinux ~]# cat cron.sh

while true

do

echo "I am oldboy teacher."

sleep 1

done

[root@oldboylinux ~]#

程序文件：程序代码组成，但是没有在计算机内执行。当前没有执行。

守护程序或守护进程：进程：所谓进程就是计算机中正在执行的程序，守护进程就是一直运行的程序。

回忆一下前面精简开机系统启动的服务操作：

LANG=en

chkconfig --list |grep "3:on" |awk '{print $1}' |grep -vE "**sshd|network|crond|rsyslog|sysstat**"|sed -r 's#(.\*)#chkconfig --level 3 \1 off#g' |bash

[root@oldboylinux ~]# chkconfig --list |grep 3:on

**crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off**

network 0:off 1:off 2:on **3:on**  4:on 5:on 6:off

rsyslog 0:off 1:off 2:on **3:on**  4:on 5:on 6:off

sshd 0:off 1:off 2:on **3:on** 4:on 5:on 6:off

sysstat 0:off 1:on 2:on **3:on**  4:on 5:on 6:off

[root@oldboylinux ~]#

### 1．2 为什么要所使用crond定时任务？

服务器：7\*24小时开机提供服务是网站的基本特征。

有重要数据：服务器上的数据一般最低1天备一次，每分钟（实时备份）再增量备份。

定时任务就产生了。定时任务可以实现帮助我们制动备份数据。

举个例子说明：我们的**数据库或者代码程序**需要每天**晚上0点做**一次全备，这样每天夜里都需要执行的**周期性的工作**，如果要让人操作的话，就得每天半夜爬起来，然后登录系统执行任务，执行完接着睡觉。另外，执行任务的过程也可能持续几个小时，这样以来，一个晚上就不用睡觉了。那么有什么办法来解决这个周期性的执行任务需求呢？

这就用到linux系统的**定时任务crond**，这相当于我们平时生活中的闹钟的功能，每天晚上提前设置定时，早上按时叫醒你。

实时备份，定时备份

## 1．3 不同系统的定时任务与种类

### 1．3．1 windows7系统的定时任务

1．点击“开始”----->“所有程序”----->“附件“----->“系统工具“----->选择“任务计划程序“

### 1．3．2 linux系统的定时任务

Linux系统中定时任务调度的工作可以分为以下两种情况：

1，linux系统自身定期执行的任务工作： 系统周期性自行执行的任务工作，比如轮询系统日志，备份系统数据，清理系统缓存等，这些任务无需认为干预。例如：

[root@oldboylinux ~]# ls -l /var/log/messages\* <====系统的日志

-rw------- 1 root root 87337 Oct 21 10:20 /var/log/messages

-rw-------. 1 root root 171142 Sep 24 17:53 /var/log/messages-20151007

-rw------- 1 root root 337448 Oct 19 10:19 /var/log/messages-20151019

[root@oldboylinux ~]# ls -l /var/log/secure\*

-rw------- 1 root root 7619 Oct 21 10:43 /var/log/secure

-rw-------. 1 root root 9143 Oct 7 10:28 /var/log/secure-20151007

-rw------- 1 root root 12421 Oct 9 12:54 /var/log/secure-20151019

[root@oldboylinux ~]#

提示：centos6.4日志轮询结尾就是按时间了。

[root@oldboylinux ~]# ls -l /var/log/messages\*

-rw------- 1 root root 87337 Oct 21 10:20 /var/log/messages

-rw-------. 1 root root 171142 Sep 24 17:53 /var/log/messages-**20151007**

-rw------- 1 root root 337448 Oct 19 10:19 /var/log/messages-**20151019**

[root@oldboylinux ~]# ls -l /var/log/secure\*

-rw------- 1 root root 7619 Oct 21 10:43 /var/log/secure

-rw-------. 1 root root 9143 Oct 7 10:28 /var/log/secure-**20151007**

-rw------- 1 root root 12421 Oct 9 12:54 /var/log/secure-**20151019**

[root@oldboylinux ~]#

像这样的工作就是由系统自身来完成的，不需要系统管理员来设置。

系统自动轮询任务的设置配置路径：

[root@oldboylinux ~]# ll /etc/ |grep cron

-rw-------. 1 root root 541 Mar 30 2015 anacrontab

drwxr-xr-x. 2 root root 4096 Sep 23 22:24 cron.d

drwxr-xr-x. 2 root root 4096 Sep 23 22:25 cron.daily

-rw-------. 1 root root 0 Mar 30 2015 cron.deny

drwxr-xr-x. 2 root root 4096 Sep 23 22:25 cron.hourly

drwxr-xr-x. 2 root root 4096 Sep 23 22:24 cron.monthly

-rw-r--r--. 1 root root 457 Sep 27 2011 crontab

drwxr-xr-x. 2 root root 4096 Sep 27 2011 cron.weekly

[root@oldboylinux ~]# cd /etc/logrotate.d

[root@oldboylinux logrotate.d]# cat syslog

/var/log/cron

/var/log/maillog

/var/log/messages

/var/log/secure

/var/log/spooler

{

sharedscripts

postrotate

/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true

endscript

}

[root@oldboylinux logrotate.d]#

2，用户执行的任务工作：某个用户或系统管理员定义要做的任务工作，例如每隔5分钟和互联网上时间服务器进行时间同步**（这是安装系统基础优化部分的内容）**

例如：服务器的时间同步

[root@oldboylinux ~]# crontab -l

#time sync by oldboy at 2010-2-1

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

[root@oldboylinux ~]#

用户执行的任务工作，也就是运维管理人员执行的任务工作。

### 1．3．3 linux系统下定时任务软件种类

严格的说，linux系统下的定时任务软件不少，例如：at，**crontab**，anacron。

* at：适合执行一次就结束的调度任务命令，例如：某天晚上需要处理一个任务，仅仅是这一天的晚上，属于突发性的工作任务。要执行at命令，还需要启动一个名为atd的服务才行，在生产环境从来都不会有需求用这个。因此，建议大家不要深入研究了。

[root@oldboylinux ~]# chkconfig --list |grep atd

atd 0:off 1:off 2:off 3:off 4:on 5:on 6:off

[root@oldboylinux ~]# chkconfig --list atd

atd 0:off 1:off 2:off 3:off 4:on 5:on 6:off

[root@oldboylinux ~]#

* **crontab：（是个命令）**正如前面所说这个命令可以周期性的执行任务工作，例如：每五分钟做一次服务器时间同步。要执行crontab这个命令，也需要启动一个服务crond才行，这个crond命令是在生产工作中常用到的命令。

[root@oldboylinux ~]# chkconfig --list crond

crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@oldboylinux ~]#

* anacron：这个命令主要用于非**7\*24**小时开机的服务器准备的，anacron并不能指定具体时间执行任务工作，而是以天为周期或者在系统每次开机后执行的任务工作。它会检测服务器停机期间应该执行，但是并没有进行的任务工作，并将该任务执行一遍。

**提示：**

1. **我们所说的crond服务是运行的程序，而crontab命令用户用来设置定时规则的命令。**
2. **Crond服务是企业生产工作中常用的重要服务，at和anacron很少使用，可以忽略。**
3. **几乎每个服务器都会用到crontab**

## 2 定时任务crond使用说明

### 2．1指令语法

crontab [-u user] [-l | -r | -e] [-i] [-s]

***查看系统帮助：***

[root@oldboylinux ~]# crontab --help

crontab: invalid option -- '-'

crontab: usage error: unrecognized option

usage: crontab [-u user] file

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

(default operation is replace, per 1003.2)

**-e (edit user's crontab)**

**-l (list user's crontab)**

-r (delete user's crontab)

-i (prompt before deleting user's crontab)

-s (selinux context)

-u **指定用户**

[root@oldboylinux ~]#

**范例：**

[root@oldboylinux ~]# crontab -l

#time sync by oldboy at 2010-2-1

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

[root@oldboylinux ~]# ls -l /usr/bin/crontab

-rw**s**r-xr-x. 1 root root 51784 Mar 30 2015 /usr/bin/crontab

[root@oldboylinux ~]# ls -l `which crontab`

-rw**s**r-xr-x. 1 root root 51784 Mar 30 2015 /usr/bin/crontab

[root@oldboylinux ~]#

***提示：如果不想让普通用户使用crontab命令，可以suid去掉权限。***

### 2．2指令说明

通过crontab我们可以在固定的间隔时间执行指定的系统指令或script脚本。时间间隔的单位可以是**分钟，小时，日，月，周**及以上的任意组合（注意日和周不要组合）。**crond服务通过crontab命令**可以很容易的实现周期性的日志分析或数据备份等企业运维场景工作。

### 2．3使用者权限及定时任务文件

|  |  |
| --- | --- |
| **文件** | **说明** |
| /etc/cron.deny | 该文件中所列用户不允许使用crontab命令 |
| /etc/cron.allow | 该文件中所列用户允许使用crontab命令/etc/cron.deny |
| /var/spool/cron | 所有用户crontab配置文件默认都存放在此目录，文件名以用户名命名**\***  [root@oldboylinux ~]# cat /var/spool/cron/root  #time sync by oldboy at 2010-2-1  \*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1  [root@oldboylinux ~]# crontab -l  #time sync by oldboy at 2010-2-1  \*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1  通过crontab查看或编辑的定时任务规则默认都会放到/var/spool/cron/，  以当前用户命名的文件。  [root@oldboylinux logrotate.d]# crontab -e === vi/vim /var/spool/cron/root  [root@oldboylinux logrotate.d]# crontab -l === vi/vim /var/spool/cron/root |

### 2．4指令选项说明含义表

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 指定示例 |
| -1（字母） | 查看crontab文件内容，提示：l可理解为list的缩写。 | crontab –l **\*\*\*** |
| -e | 编辑crontab文件内容，提示：e可理解为edit的缩写。 | crontab –e **\*\*\*** |
| -i | 删除crontab文件内容，删除前会提示确认。用的很少。 | crontab –ri |
| -r | 删除crontab文件内容。用的很少。 | crontab -r |
| -u user | 指定使用的用户执行任务。 | crontab –u boy –l |
| 特别强调：-i，-r参数在生产中很少用，没什么需求必须要用-e进去编辑即可。  **补充：crontab｛-l|-e｝实际上就是操作/var/spool/cron/当前用户这样的文件。**  系统帮助：  SYNOPSIS  crontab [-u user] file  crontab [-u user] [-l | -r | -e] [-i] [-s]  (default operation is replace, per 1003.2)  **-e (edit user's crontab)**  **-l (list user's crontab)**  -r (delete user's crontab)  -i (prompt before deleting user's crontab)  -s (selinux context) | | |

### 2．5定时任务指令使用格式

默认情况下，当用户建立定时任务规则后，该规则记录对应的配置文件会存在于/var/spool/cron中，其crontab配置文件对应的文件名与登录的用户名一致，如：root用户的定时任务配置文件为/var/spool/cron/root。

Crontab定时任务的书写格式很简单，**用户的定时任务**规则一般分为6个段（每个段通过空格来分隔，系统的定时任务为/etc/crontab，分为8段，空格来分隔），前五段为时间的设定段，第六段为所要执行的命令或脚本任务段。

**基本格式如下：**

01 \* \* \* \* cmd

02 4 \* \* \* cmd

22 4 \* \* 0 cmd

42 4 1 \* \* cmd

提示：

1，cmd为要执行的命令或脚本，例如/bin/sh /server/scripts/oldboy.sh

2，每个列之间必须要有一个空格。多个空格可以吗？

[root@oldboylinux ~]# cat /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

HOME=/

# For details see man 4 crontabs

# Example of job definition:

# .---------------- minute (0 - 59)

# | .------------- hour (0 - 23)

# | | .---------- day of month (1 - 31)

# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...

# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat

# | | | | |

# \* \* \* \* \* user-name command to be executed

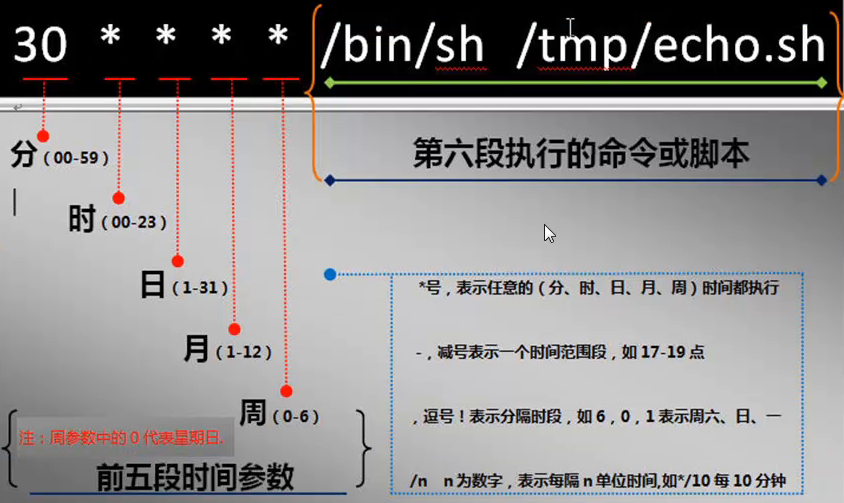
#### 2．5．1 crontab语法格式中时间段的含义如下表

|  |  |  |
| --- | --- | --- |
| **段** | **含义** | **取值范围（整数）** |
| 第一段 | 代表分钟 | 00-59（00也可以是0） |
| 第二段 | 代表小时 | 00-23 |
| 第三段 | 代表日，天 | 01-31 |
| 第四段 | 代表月份 | 01-12 |
| 第五段 | 代表星期，周几 | 0-7 （0和7都代表星期日） |
| **提示：时间记忆口诀：分时日月周。取值范围记忆：正常日期时间范围。** | | |

#### 2．5．2 crontab语法格式中特殊符号含义如下表

|  |  |
| --- | --- |
| **特殊符号** | **含义** |
| \* | **\*号，表示任意时间都，实际就是“每”时间的意思**。举例：如**00 23 \* \* \* cmd**  表示每月每周每日的23：00都执行cmd任务。需要注意的是：每个时间位上的  \*表示每，如果位上是\*就是该位上时间的取值范围，例如：小时上的\*等价于00-23  技巧：定时任务规则如果到小时，就最多提每天。  分位上的\*就等价于0-59。表示每分。 |
| - | 减号，表示分隔符，表示一个时间范围，区间段，如**17-19**点，每天的17，1，19  的00分执行任务。**00 17-19 \* \* \* cmd**。就是17，18，19点整点分别执行的意思。 |
| ， | 逗号，表示分隔时段的意思。30 **17，18，19** \* \* \* /bin/sh /scripts/oldboy.sh表示  每天17，18和19点的半点时刻执行/scripts/oldboy.sh脚本。**也可以和“-”结合使用，**  **例如：30 3-5，17-19 \* \* \* /bin/sh/scripts/oldboy.sh** |
| /n | N代表数字，即“每隔n单位时间”，例如：每10分钟执行一次可以  写成**\*/10 \* \* \* \* cmd,**其中，\*/10，\*的范围是0-59，因此，也可以写成0-59/10。  [root@oldboylinux ~]# crontab -l  #time sync by oldboy at 2010-2-1  \*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1 |

#### 2．5．3 学习画图来表达说明



[**http://oldboy.blog.51cto.com/2561410/1180894**](http://oldboy.blog.51cto.com/2561410/1180894)

#### 2．5．4 crontab编辑的定时任务依赖的服务

[root@oldboylinux ~]# chkconfig --list crond

crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@oldboylinux ~]# /etc/init.d/crond status

crond (pid 1267) is running...

[root@oldboylinux ~]# ps -ef |grep crond |grep -v grep

root 1267 1 0 Oct20 ? 00:00:00 crond

[root@oldboylinux ~]# /etc/init.d/crond restart

Stopping crond: [ OK ]

Starting crond: [ OK ]

[root@oldboylinux ~]#

## 3 定时任务crontab实例说明

**范例1：**

[root@oldboylinux ~]# crontab -l

\*/1 \* \* \* \* /bin/sh /scripts/data.sh

或者

\* \* \* \* \* /bin/sh /scripts/data.sh

在上例中除了数字与命令脚本外，还使用到了符号“\*”,其中\*号意思每一个单位。

第一列为分钟，特殊符号“/”表示每隔的意思，即每隔一分钟执行/bin/sh /scripts/data.sh程序。

**范例2：**每五分钟同步一次系统时间

[root@oldboylinux ~]# crontab -l

#time sync by oldboy at 2010-2-1

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

[root@oldboylinux ~]#

＃---->这个命令其实就是写一个定时任务，相当于执行crontab　–e然后加入内容：

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1　保存退出。

|  |  |  |
| --- | --- | --- |
| crontab　–e | /var/spool/cron/root | crontab –e 编辑会检查语法，而直接echo,vi编辑配置不会。 |
| visudo | /etc/sudoers | visudo编辑会检查语法，而直接echo,vi编辑配置不会。 |

下面我们再看几个例子：

1. 30 3，12 \* \* \* /bin/sh /scripts/oldboy.sh

在本例中，第一列为30，表示30分钟，第二列为3，12，这代表3点及12点，此定时任务的意思是**每天凌晨3点和中午12点的半点**时刻（或描述为每天凌晨3：30和中午12：30）执行/scripts/oldboy.sh脚本

1. 30 \*/6 \* \* \* /bin/sh /scripts/oldboy.sh

在本例中，第一列为30，表示30分钟，第二列\*/6代表每6个小时，也相当于6，12，18，24的作用。此定时任务的意思是每隔6个小时的半点时刻执行/scripts/oldboy.sh脚本任务。

1. 30 8-18/2 \* \* \* /bin/sh /scripts/oldboy.sh

在本例中，第一列为30，表示30分钟，第二列8-18/2代表在早餐8点到下午18点之间每隔2小时，也相当于8，10，12，14，16，18单独列出的作用。

那么，此定时任务的意思就是早餐8点到下午18点之间每隔2小时的30分钟时刻执行/scripts/oldboy.sh脚本任务。

1. 30 21 \* \* \* /application/apache/bin/apachect1 graceful

上面的例子表示每晚的21：30重启apache。

1. 45 4 1，10，22 \* \* /application/apache/bin/apachect1 graceful

上面的例子表示每月1，10，22日的凌晨4：45分重启apache。

1. 10 1\* \* 6，0 \* \* /application/apache/bin/apachect1 graceful

上面的例子表示每周六，日的凌晨1：10分重启apache。

1. 0，30 18-23 \* \* \* /application/apache/bin/apachect1 graceful

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。

提示：最后一次执行任务是23：30分

1. 00 \*/1 \* \* \* /application/apache/bin/apachect1 graceful

上面的例子表示每隔一小时整点重启apache

1. **\* 23，00-07/1 \* \* \* /application/apache/bin/apachect1 graceful**

上面的例子并不表示晚上23点和早上0-7点之间每隔一小时重启apache

以上的定时任务的第一列为\*，表示每分都执行任务即晚上23点和早上0-7点之间每隔一小时的每分都重启apache。

1. 00 11 \* 4 1-3 /application/apache/bin/apachect1 graceful

上面的例子表示4月的每周一到周三的上午11点整重启apache

**强调：周和日尽量不要同时用，否则可能达不到想要的效果。**

## 4生产环境crontab专业实例

### 4．1书写crontab定时任务多个基本要领

#### 4．1.1规范定时任务两例

1，定时任务要添加注释

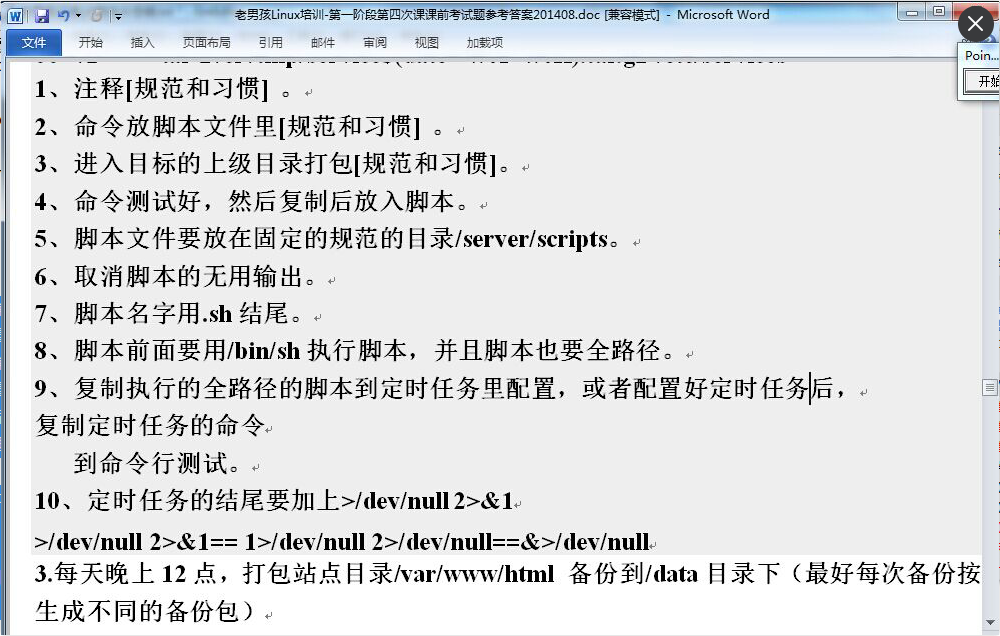
2，结尾不要有>/dev/null 2&>1(>/dev/null 2>&1== 1>/dev/null 2>/dev/null==&>/dev/null)

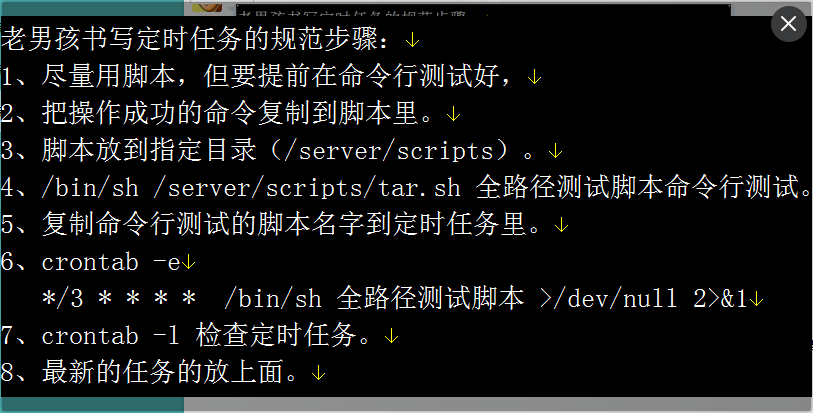
3，/server/log目录必须要存在才能出结果

4，定时任务中的路径一定要绝对路径

5，crond服务必须开启

6，习惯查看定时任务日志tail /var/log/cron





1、在每周6的凌晨3:15执行/home/shell/collect.pl，并将标准输出和标准错误输出到/dev/null设备，请写出crontab中的语句。

15 3 \* \* 06 /usr/bin/perl /home/shell/collect.pl >/dev/null 2>&1

chmod +x /home/shell/collect.pl

15 3 \* \* 06 /home/shell/collect.pl >/dev/null 2>&1

2、crontab在11月份内，每天的早上6点到12点中，每个2小时执行一次/usr/bin/httpd.sh怎么实现。

00 06-12/2 \* 11 \* /bin/sh /usr/bin/httpd.sh >/dev/null 2>&1

\* \* \* \* \*

\*==0-59

\*==0-23

\*==1-31

\*==1-12

\*==0-7

#### 4．1.2试题：

1. 每天晚上00:30点，打包站点目录/etc/rc.local,/var/spool/cron,/etc/hosts,备份到/backup目录下,每次备份按时间生成不同的备份包,删除3天以前的备份包。

[root@oldboylinux scripts]# vim backup.sh

tar zcf /backup/bak\_$(date +%m-%d-%T).tar.gz /etc/rc.local /etc/hosts /var/spool/cron

[root@oldboylinux scripts]# find /backup/ -type f -name "bak\*.tar.gz"

/backup/bak\_10-27-13:32:11.tar.gz

/backup/bak\_10-27-13:35:43.tar.gz

[root@oldboylinux scripts]# vim backup.sh

#!/bin/sh

tar zcf /backup/bak\_$(date +%m-%d-%T).tar.gz /etc/rc.local /etc/hosts /var/spool/cron

find /backup/ -type f -name "bak\*.tar.gz" -mtime +3 |xargs rm –rf

[root@oldboylinux scripts]# crontab -l |tail -2

#####back and rm for test by oldboy 2015-10-27

30 00 \* \* \* /bin/sh /server/scripts/backup.sh >dev/null 2>&1

-h, --dereference：打包软链接

--hard-dereference：打包硬链接

1.创建备份目录

[root@student ~]# mkdir /backup

2.创建脚本目录

[root@student ~]# mkdir /server/scripts

3.将测试结果写入脚本(已在本地测试ok)

[root@student scripts]# cat tar.sh

#!/bin/sh

cd / &&\

/bin/tar zcf /backup/config\_$(date +%F-%T).tar.gz ./etc/rc.local ./var/spool/cron ./etc/hosts &&\

/bin/find /backup/ -type f -name "\*.tar.gz" -mtime +3|xargs rm -f

4.写入定时任务(注意规范写入操作)

[root@student scripts]# tail -2 /var/spool/cron/root

#####00:30 tar config to backupdir####

30 00 \* \* \* /bin/sh /server/scripts/tar.sh >/dev/null 2>&1

5.查看crond服务是否成功(如没启动则会执行失败 使用/etc/init.d/crond start启动)

[root@student scripts]# /etc/init.d/crond status

crond (pid 1078) is running.

## 5．工作中调试定时任务的方法

1. 增加执行任务频率调试任务

每分钟或者系统时间之后5分钟执行，看看结果对不对

某些任务不能用于生产环境没有机会测试

代码、配置变更发布流程：个人开发环境---办公测试环境--->IDC机房测试环境-->IDC正式环境(分组，灰度发布）。

1. 调整系统时间调试任务（不能直接用于生产环境），保存5分钟
2. 通过脚本日志输出调试定时任务。
3. 注意一些任务命令带来的问题

例如：

>/dev/null 2&>1(>/dev/null 2>&1== 1>/dev/null 2>/dev/null==&>/dev/null)

\* \*/2 \* \* \* /bin/tar zcvf /tmp/services\_`(date +**%T**)`.tar.gz /etc/services >/dev/null 2>&1

1. 注意：环境变量导致的定时任务故障（java）
2. 通过crond定时任务服务日志调试定时任务（**/var/log/cron**）
3. 其他稀奇古怪的问题调试的方法