crontab：定时任务的守护进程，精确到分，设计秒的我们一般写脚本  -->相当于闹钟  
     日志文件:  ll /var/log/cron\*  
     编辑文件：系统任务： vim /etc/crontab       用户任务：在/var/spool/cron目录中。其文件名与用户名一致  
        进程：ps -ef | grep crond  ==> /etc/init.d/crond restart  
       定时备份：实时备份crontab -l > $HOME/mycron

**常见命令参数**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | 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) |

**定时任务分类**

Linux下的任务调度分为两类，系统任务调度和用户任务调度。  
    系统任务调度：系统周期性所要执行的工作，比如写缓存数据到硬盘、日志清理等。  
        在/etc/crontab文件，这个就是系统任务调度的配置文件。  
    用户任务调度：用户定期要执行的工作，比如用户数据备份、定时邮件提醒等。  
        用户可以使用 crontab 工具来定制自己的计划任务。  
        在crontab 文件都被保存在/var/spool/cron目录中。其文件名与用户名一致

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | 1.系统定时任务：例如清理系统日志，清理系统缓存   -->不过多的关注      查询系统定时处理任务的路径：          路径1：              cd /etc/logrotate.d/        -->可以写定时任务              less syslog          路径2：              cat /etc/crontab   -->不推荐使用，但是可以看格式          路径3：               ls /etc/ | grep cron\*                   anacrontab                   cron.d                 -->同路径2 ，可以写定时任务                   cron.daily                   cron.deny              -->控制普通用户使用定时任务crontab                   cron.hourly                   cron.monthly                   crontab                   cron.weekly  2.用户的定时任务      -->关注重点 |

**crontab文件内容分析**

cat /etc/crontab

[](https://images2018.cnblogs.com/blog/519608/201807/519608-20180707143445382-412660140.png)

前四行是用来配置crond任务运行的环境变量

第一行SHELL变量指定了系统要使用哪个shell，这里是bash

第二行PATH变量指定了系统执行命令的路径

第三行MAILTO变量指定了crond的任务执行信息将通过电子邮件发送给root用户

如果MAILTO变量的值为空，则表示不发送任务执行信息给用户

第四行的HOME变量指定了在执行命令或者脚本时使用的主目录。

星号（\*）：代表所有可能的值，如month字段为星号，则表示在满足其它字段的制约条件后每月都执行该命令操作。

逗号（,）：可以用逗号隔开的值指定一个列表范围，例如，“1,2,5,7,8,9”

中杠（-）：可以用整数之间的中杠表示一个整数范围，例如“2-6”表示“2,3,4,5,6”

正斜线（/）：可以用正斜线指定时间的间隔频率，例如“0-23/2”表示每两小时执行一次。

小 结：  
    数字的表示最好用2为阿拉伯数字显示  
    周和日最好不要同时用  
    定时任务要加注解  
    可以定向到日志文件或者空文件  
    定时任务一定是绝对路径，且目录必须存在才能出结果  
    crontab 服务一定要开启运行

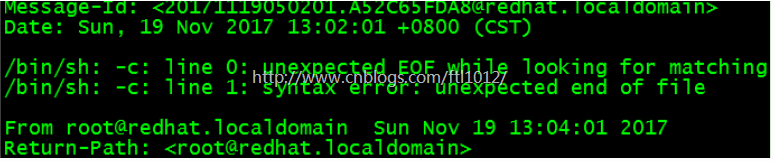
**crontab日志路径**

|  |  |
| --- | --- |
| 1  2  3 | # 【日志是按照天排列的】  ll /var/log/cron\*  **/var/log/cron**只会记录是否执行了某些计划的脚本【查看定时器是否执行和一些特征】 |



具体执行是否正确以及脚本执行过程中的一些信息则linux会每次都发邮件到该用户下。

|  |  |
| --- | --- |
| 1 | less /var/spool/mail/root |



**常用的命令展示**

安装crontab：这是linux系统自带的一般不需要安装

|  |  |
| --- | --- |
| 1 | yum install crontabs |

crontab服务操作说明：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | /sbin/service crond start //启动服务    /sbin/service crond stop //关闭服务    /sbin/service crond restart //重启服务    /sbin/service crond reload //重新载入配置 |

查看crontab服务状态：

|  |  |
| --- | --- |
| 1 | service crond status |

手动启动crontab服务：

|  |  |
| --- | --- |
| 1 | service crond status |

查看crontab服务是否已设置为开机启动，执行命令：

|  |  |
| --- | --- |
| 1  2 | 方法一： 界面启动      ntsysv  方法二： 加入开机自动启动：   chkconfig –level 35 crond on |

crontab -l查看定时任务列表

编辑定时任务

|  |  |  |
| --- | --- | --- |
| 1 | | **crontab –e //等同下** |
| 1 | **==》vim /var/spool/cron/root** | |

每隔2分钟输出时间到文件

|  |  |
| --- | --- |
| 1 | \*/2 \* \* \* \* echo `date` >> $HOME>test.txt |

crontab -r 是删除定时任务删除整个文件 //如果时多个任务打开文件一个一个删除即可

|  |  |
| --- | --- |
| 1  2  3 | ==> 从/var/spool/cron目录中删除用户的crontab文件  ==> 如果不指定用户，则默认删除当前用户的crontab文件   crontab –i  在删除用户的crontab文件时给确认提示 |

备份crontab文件

|  |  |
| --- | --- |
| 1 | crontab -l > $HOME/mycron |

恢复丢失的crontab文件

|  |  |
| --- | --- |
| 1  2  3 | 如果不小心误删了crontab文件，假设你在自己的$ H O M E目录下还有一个备份，那么可以将其拷贝到/var/spool/cron/<username>，其中<username>是用户名。如果由于权限问题无法完成拷贝，可以用：  crontab <filename> 其中，<filename>是你在$ H O M E目录中副本的文件名。  有些crontab的变体有些怪异，所以在使用crontab命令时要格外小心。如果遗漏了任何选项，crontab可能会打开一个空文件，或者看起来像是个空文件。这时敲delete键退出，不要按<Ctrl-D>，否则你将丢失crontab文件。 |

每小时执行/etc/cron.hourly目录内的脚本

|  |  |
| --- | --- |
| 1 | 0 \* \* \* \* root run-parts /etc/cron.hourly |

每隔2分钟同步一次互联网时间

|  |  |
| --- | --- |
| 1 | echo "\*/2 \* \* \* \* /usr/bin/ntpstat time.windows.com >/dev/null 2>&1" >> /var/spool/cron/root |

每天3-5,17-20每隔30分钟执行一次脚本

|  |  |
| --- | --- |
| 1 | echo "\*/30 [3-5],[17-20] \* \* \* /bin/sh /home/omc/h.sh >/dev/null 2>&1" >> /var/spool/cron/root |

当天23点，第二天0点到凌晨7点 每隔1分钟执行一次脚本

|  |  |
| --- | --- |
| 1 | \* 23,00-07/1 \* \* \* /bin/sh /home/omc.h.sh |

每年的4月份每周的周一到周三的11点执行脚本

00 11 \*  4 1-3 /bin/sh /home/omc/h.sh

|  |  |
| --- | --- |
|  | **其它例子：** |
|  | 每一分钟执行一次 /bin/ls：  \* \* \* \* \* /bin/ls  在 12 月内, 每天的早上 6 点到 12 点，每隔 3 个小时 0 分钟执行一次 /usr/bin/backup：  0 6-12/3 \* 12 \* /usr/bin/backup  周一到周五每天下午 5:00 寄一封信给 alex@domain.name：  0 17 \* \* 1-5 mail -s "hi" alex@domain.name < /tmp/maildata  每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"：  20 0-23/2 \* \* \* echo "haha"  下面再看看几个具体的例子：  0 \*/2 \* \* \* /sbin/service httpd restart 意思是每两个小时重启一次apache  50 7 \* \* \* /sbin/service sshd start 意思是每天7：50开启ssh服务  50 22 \* \* \* /sbin/service sshd stop 意思是每天22：50关闭ssh服务  0 0 1,15 \* \* fsck /home 每月1号和15号检查/home 磁盘  1 \* \* \* \* /home/bruce/backup 每小时的第一分执行 /home/bruce/backup这个文件  00 03 \* \* 1-5 find /home "\*.xxx" -mtime +4 -exec rm {} \; 每周一至周五3点钟，在目录/home中，查找文件名为\*.xxx的文件，并删除4天前的文件。  30 6 \*/10 \* \* ls 意思是每月的1、11、21、31日是的6：30执行一次ls命令 |

**crontab的注意事项**

1. 注意环境变量问题

在crontab文件中定义多个调度任务时，需要特别注意的一个问题就是环境变量的设置

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | # 脚本中涉及文件路径时写全局路径；  # 脚本执行要用到java或其他环境变量时，通过source命令引入环境变量，如：  cat start\_cbp.sh  #!/bin/sh  source /etc/profile  export RUN\_CONF=/home/d139/conf/platform/cbp/cbp\_jboss.conf  /usr/local/jboss-4.0.5/bin/run.sh -c mev &    # 当手动执行脚本OK，但是crontab死活不执行时。可以尝试在crontab中直接引入环境变量解决问题。  0 \* \* \* \* . /etc/profile;/bin/sh /var/www/java/audit\_no\_count/bin/restart\_audit.sh |

2. 系统级任务调度与用户级任务调度

|  |  |
| --- | --- |
| 1 | root用户的任务调度操作可以通过“crontab –uroot –e”来设置，也可以将调度任务直接写入/etc/crontab文件，需要注意的是，如果要定义一个定时重启系统的任务，就必须将任务放到/etc/crontab文件，即使在root用户下创建一个定时重启系统的任务也是无效的。 |

3.其他注意事项

|  |  |
| --- | --- |
| 1  2  3 | 当crontab突然失效时，可以尝试/etc/init.d/crond restart解决问题。或者查看日志看某个job有没有执行/报错tail -f /var/log/cron。  千万别乱运行crontab -r。它从Crontab目录（/var/spool/cron）中删除用户的Crontab文件。删除了该用户的所有crontab都没了。  在crontab中%是有特殊含义的，表示换行的意思。如果要用的话必须进行转义\%，如经常用的date ‘+%Y%m%d’在crontab里是不会执行的，应该换成date ‘+\%Y\%m\%d’ |

4. 生产调试定时任务

|  |  |
| --- | --- |
| 1  2  3  4 | 1.增加执行任务的频率调试  2.调整系统时间调试任务，提前5分钟   -->不用于生产环境  3.通过脚本日志输出调试定时 任务  4.注意一些任务命令带来的问题        -->确保命令的正确性 |

5.crontab箴言

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | 1.环境变量问题，例如crontab不能识别Java的环境变量      crontab执行shell时，只能识别为数不多的环境变量，普通的环境变量是无法识别的，所以在编写shell时，最好使用export重新声明变量，确保脚本执行。  2.命令的执行最好用脚本  3.脚本权限加/bin/sh，规范路径/server/scripts  4.时间变量用反斜线转义，最好用脚本  5.定时任务添加注释  6.>/dev/null 2>&1   ==>&>/dev/null,别随意打印日志文件  7.定时任务里面的程序脚本尽量用全路径  8.避免不必要的程序以及命令输出  9.定时任务之前添加注释  10.打包到文件目录的上一级 |