

计划任务和时间同步

- 前言：

一个人就算是再尽职尽责，再怎么屹立不倒，也不能天天二十四小时盯着屏幕看，如果有什么事情能交给机器自己去处理，那该多好啊，这样就省去很多工作了。Linux下当然可以做到，这里就要用到计划任务的crontab了。至于时间同步，就是给服务器调整时间等，我们细细道来。

- 正文：

1、学习crontab的基本操作：

```
0 8 * * * df -h >> /tmp/df.log  
0 20 * * * df -h >> /tmp/df.log
```

话不多说，先使用crontab -e 命令进入一个计划任务的文档编辑界面，我输入了以上两条玩意儿，这就是要在何时执行什么命令的意思。

```
"/tmp/crontab.yi00g7" 2L, 63C written  
crontab: installing new crontab  
[root@MiWiFi-R4CM-srv ~]# date  
Tue Sep 14 19:53:49 EDT 2021  
[root@MiWiFi-R4CM-srv ~]# _
```

看一下本地时间啊，快到了，正好可以看看有没有效果，waiting~

```
[root@MiWiFi-R4CM-srv ~]# date  
Tue Sep 14 20:00:32 EDT 2021  
[root@MiWiFi-R4CM-srv ~]# cd /tmp/  
[root@MiWiFi-R4CM-srv tmp]# ls  
dff.log  
systemd-private-ac5cb6c4874f4096a2f42f4402890be5-chronyd.service-U6bh2W  
test_sgid  
yum_save_tx.2021-09-14.19-15.30q_mf.yumtx  
yum_save_tx.2021-09-14.19-18.iPbQso.yumtx  
[root@MiWiFi-R4CM-srv tmp]# cat dff.log  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        484M   0  484M   0% /dev  
tmpfs            496M   0  496M   0% /dev/shm  
tmpfs            496M  6.9M  489M   2% /run  
tmpfs            496M   0  496M   0% /sys/fs/cgroup  
/dev/mapper/centos-root 6.2G  1.9G  4.4G  30% /  
/dev/sda1       1014M  141M  874M  14% /boot  
/dev/mapper/mjvg01-mjlv01 2.0G  6.0M  1.8G   1% /opt1  
tmpfs            100M   0  100M   0% /run/user/0  
[root@MiWiFi-R4CM-srv tmp]# _
```

到点了，上号！可以看到真的完成了自动任务，crontab就是这么方便，那么我之前写的那两行什么意思呢？接下来我来逐个讲解。

①crontab -e 相当于用 vim 打开 /var/spool/cron/ 下的一个文件而已，并且不同用户只能定义自己的计划任务（除了root）。

②后半部分一看就是要执行的命令，没啥好看的，我们只看前面的0 20 * * * 五个字。

③0 指分，20 指时（0-24），* 指日，第二个* 指月，第三个* 指周（周一到周日）。* 代表的是任意数，那么这一条语句的意思就是——不管几月几号星期几，反正每天的20:00都检查一下硬盘，并把硬盘信息追加到dff.log里。

再举个例子，10 6 * 8 7 表示8月份时不管几号，只要是周日早上6:10一到，马上执行相应的命令。

2、学点crontab的扩展格式：

照上面说的，crontab的那一栏五个时间，表示的是什么时候做什么什么事，*号表示该位上所有的时间，那么问题来了，10 * * * *表示每个xx:10时间点执行一次，也就是一小时只执行一次，这一次在第10分，可是我想做到每隔10分执行一次，一个小时执行6次，怎么办呢？

```
*/2 * * * * df -h
*/5 10-14 * * * df -h
*/10 0-12/2 * * * df -h
```

这就是新的内容，一个/和一个-又可以玩出新花样了。/表示时间的间隔频率，-表示时间范围。第一行的意思是，所有时间点里每隔两分钟就df一次；第二行意思是每天的10-14点里每隔5分钟df一次；第三行的意思是每天的0-12小时里每隔2小时，再每隔10分钟df一次。

如果我想每隔两小时df一次，不要*/2 * * * *这样子，因为这是每隔两小时再每分钟执行，例如从10到12点0分时，就会开始每分钟执行一次，12:01、12:02.....12:59这样。所以可以在分位上加上一个特定时间，例如0、1之类的。

3、缩短crontab的时间间隔：

肯定有的小伙伴注意到了，crontab的最小时间单位是1min，那我要是想每隔几秒执行一次命令呢，那crontab岂不是用不了了？

当然，办法总比困难多，如果我们能让一组相同的命令在同一个时间段内的不同时间点执行，例如让一组df命令在一分钟内的不同秒依次执行，这样就可以造出一个每多少秒执行一次命令的假象了。

```
* * * * * date >> /root/sleep.log
* * * * * sleep 10 ; date >> /root/sleep.log
* * * * * sleep 20 ; date >> /root/sleep.log
* * * * * sleep 30 ; date >> /root/sleep.log
* * * * * sleep 40 ; date >> /root/sleep.log
* * * * * sleep 50 ; date >> /root/sleep.log
```

因此我们这里用到了sleep命令，后面接上数字表示让命令睡个10秒再执行后面的内容，而分号的作用就是隔开语句，表示这一行有多条命令要先后执行。如图，我们模拟出了每隔十秒打印一次时间。另外，crontab -l可以查看当前用户的全部计划任务。

```
[root@localhost ~]# ls
anaconda-ks.cfg  rpmbuild  sleep.log  test.txt  test.txt~
[root@localhost ~]# cat sleep.log
Tue Sep 14 23:28:02 EDT 2021
Tue Sep 14 23:28:12 EDT 2021
Tue Sep 14 23:28:22 EDT 2021
Tue Sep 14 23:28:32 EDT 2021
Tue Sep 14 23:28:42 EDT 2021
Tue Sep 14 23:28:52 EDT 2021
Tue Sep 14 23:29:01 EDT 2021
Tue Sep 14 23:29:11 EDT 2021
Tue Sep 14 23:29:21 EDT 2021
Tue Sep 14 23:29:31 EDT 2021
Tue Sep 14 23:29:41 EDT 2021
Tue Sep 14 23:29:51 EDT 2021
[root@localhost ~]#
```

看时间点，确实是每隔十秒打印一次时间，上述操作没有问题。我在想啊，要是每隔1s执行一次，岂不是要写上60行？不过到后面我们学了脚本编程后应该可以通过编程来把这些乏味的工作交给机器吧。

4、时间同步问题：

之前我开始用crontab展示时，不知道各位有没有留意到，我第一条写的时0 8 * * * df? 那时候其实时间是早上7点，但是我date查看Linux系统的时间发现写着19点，然后为了演示crontab的功能，我只能迎合机器，又写了一行0 20 * * * df。

给服务器时间同步是个很重要的问题，例如一些游戏里的跨区干群架，就是不同服务器的数据互通到一起，试想，一个服务器是早上7点的时间，一个服务器的时间是晚上7点，它们没有统一时间，如果策划要求干架服务器今晚8点开始，那就是别的区8点截止报名了，那么第一个区还有13个小时截止，第二区还有1个小时，岂不是全乱了，这时候如果干架区已经开了1个小时，那不是全完了❷?

```
[root@localhost ~]# date
Tue Sep 14 23:56:23 EDT 2021
[root@localhost ~]# timedatectl set-timezone Asia/Shanghai
[root@localhost ~]# date
Wed Sep 15 11:57:39 CST 2021
[root@localhost ~]# date -s "2021-09-15 11:59:00"
Wed Sep 15 11:59:00 CST 2021
[root@localhost ~]# ntpdate 0.asia.pool.ntp.org
15 Sep 11:59:36 ntpdate[3275]: no server suitable for synchronization found
[root@localhost ~]#
```

看图第一次date时，EDT表示时区，这是美国东部夏令时时间，我们要换成中国标准时间（CST），所以执行timedatectl set-timezone + 时区命令来更改时区，改成CST。但是有时候改了date后还是不标准，这是因为只是改了时区，系统把先前时区的时间换算成现在时区的时间，而曾经没同步过时间的话，时间还是错乱了（意思是旧时区上的时间也是错的），光换时区没用。当然我的图示是正确的。

①可以手动date -s + "详细时间"，来更改本地时间；

②也可ntpdate命令，来链接网络上的时间服务器的标准时间。

由于我是在图书馆的，没登学校账号没有分配到网络，所以提示没有服务器。后来我回到宿舍成功连到时间服务器校准了一波时间。

另外，一般可以ntpdate+crontab一起用，每过一段时间连一次服务器校准时间：0 */2 * * * ntpdate 0.asia.pool.ntp.org || ntpdate ntp1.aliyun.com。

- 后记：

这一节蛮简单但是也挺常用的，多用即可。