

# 1、简介

## 1.1、来源

**crontab** 命令常见于 Unix 和类 Unix 的操作系统之中，用于设置周期性被执行的指令。该命令从标准输入设备读取指令，并将其存放于“crontab”文件中，以供之后读取和执行。该词来源于希腊语 chronos (χρόνος)，原意是时间。<sup>[3][4]</sup>

通常，**crontab** 储存的指令被守护进程激活，**crond** 常常在后台运行，[每一分钟检查是否有预定的作业需要执行](#)。这类作业一般称为 cron jobs。<sup>[3][4]</sup>

## 1.2、什么是 crontab

**crontab** 是 **crond** 服务提供的使用命令。

**crontab** 命令被用来[提交和管理](#)用户和需要[周期性执行的任务](#)，与 windows 下的计划任务类似，当安装完成操作系统后，默认会安装此服务工具，并且会自动启动 **crond** 进程，**crond** 进程[每分钟会定期检查是否有要执行的任务](#)，如果有要执行的任务，则自动执行该任务。<sup>[1]</sup>

linux 系统上面原本就有非常多的计划性工作，是由 **crond** 这个系统服务来控制的，因此这个系统服务是[默认启动的](#)。<sup>[2][5]</sup>

# 2、语法

```
crontab [options] file
crontab [options]
crontab -n [hostname]
```

# 3、参数选项

- **-e** : 编辑当前用户的计时器工作内容;
- **-i** : 删除操作前打印确认提示 yes/no;
- **-l** : 列出当前用户的计时器工作内容;
- **-r** : 删除当前用户的所有计时器工作内容，如果仅需要移除一项，请用 **-e** 去编辑;
- **-u [UserName]**: 帮助指定用户[新建/移除](#)计时器工作任务，该选项一般是 root 管理员用于为其他用户设置定时任务，不写时默认是 root 用户。

## 4、 crontab 工作内容配置含义

用户所建立的 crontab 文件中，每一行都代表一项任务，每行的每个字段代表一项设置，它的格式共分为六个字段，前五段是时间设定段，第六段是要执行的命令段，格式如下：

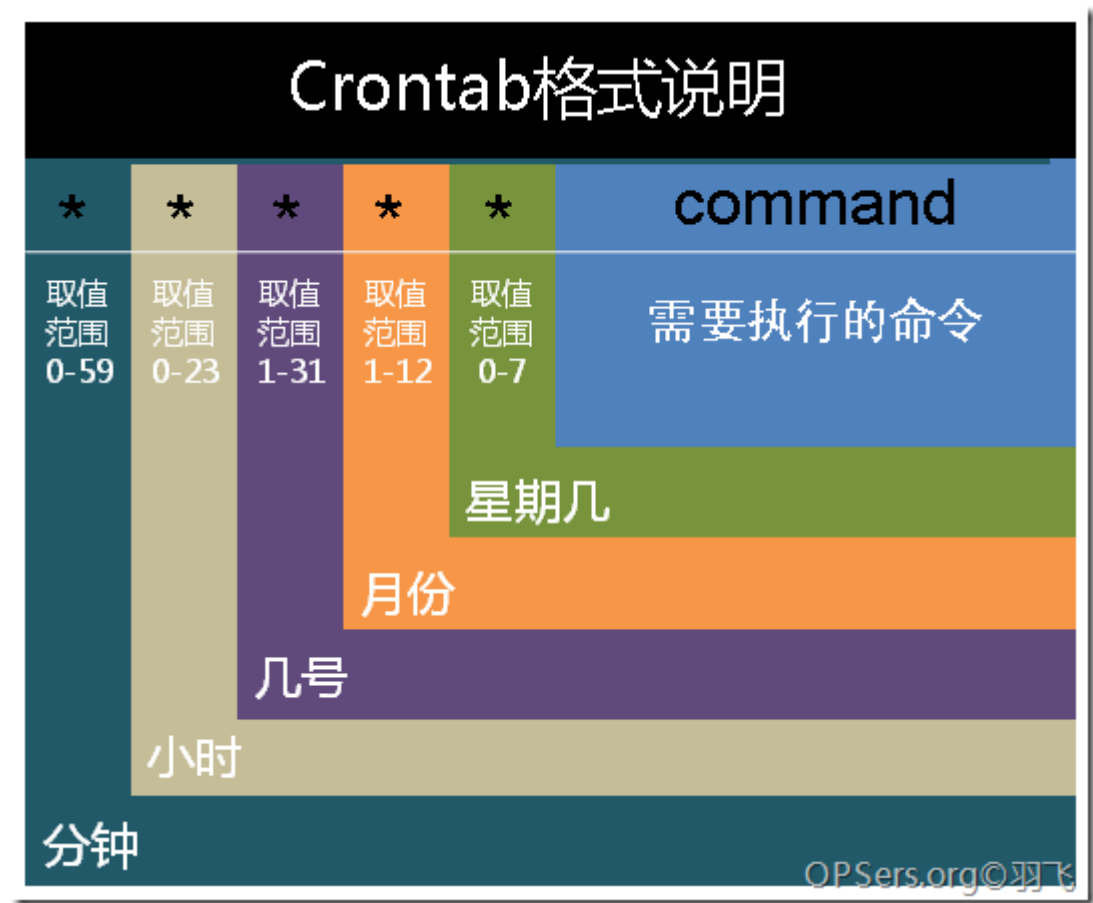


图 4-1 crontab 格式说明<sup>[2]</sup>

上述格式中，各项内容的具体含义如下：

表 4-1 crontab 格式详细说明

格式内容	具体含义
分钟 minute	表示分钟，可以是 0 到 59 之间的任何整数。
小时 hour	表示小时，可以是 0 到 23 之间的任何整数。
几号 day	表示日期，可以是 1 到 31 之间的任何整数。
月份 month	表示月份，可以是 1 到 12 之间的任何整数。
星期几 week	表示星期几，可以是 0 到 7 之间的任何整数， <a href="#">这里的 0 或 7 都代表星期日</a> 。
需要执行的命令 command	要执行的命令，可以是系统命令，也可以是自己编写的脚本文件。

在以上各个字段中，除了命令字段，还可以使用以下特殊字符：

表 4-2 特殊字符详细说明

特殊字符	具体含义
* (星号)	代表所有可能的值，例如 month 字段如果是星号，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（此时等价于 1-12）。
, (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”。
- (减号)	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”。
/n (斜线)	可以用正斜线指定时间的间隔频率，那个数字 n，代表“每隔 n 单位间隔”的意思。  例如“0-23/2”表示从 0 小时开始，每间隔等待 2 小时就会执行一次任务。同时正斜线可以和星号一起使用，例如*/10，如果用在 minute 字段，表示每十分钟执行一次。

## 5、 实例<sup>[1]</sup>

### 5.1、 每 1 分钟执行一次 command

```
* * * * * command
```

### 5.2、 每小时的第 3 和第 15 分钟执行

```
3,15 * * * * command
```

### 5.3、 在上午 8 点到 11 点的第 3 和第 15 分钟执行

```
3,15 8-11 * * * command
```

### 5.4、 每隔两天的上午 8 点到 11 点的第 3 和第 15 分钟执行

```
3,15 8-11 */2 * * command
```

### 5.5、 每个星期一的上午 8 点到 11 点的第 3 和第 15 分钟 执行

```
3,15 8-11 * * 1 command
```

### 5.6、 每晚的 21:30 重启 smb

```
30 21 * * * /etc/init.d/smb restart
```

### 5.7、 每月 1、10、22 日的 4 : 45 重启 smb

```
45 4 1,10,22 * * /etc/init.d/smb restart
```

### 5.8、 每周六、周日的 1:10 重启 smb

```
10 1 * * 6,0 /etc/init.d/smb restart
```

### 5.9、 每天 18 : 00 至 23 : 00 之间每隔 30 分钟重启 smb

```
0,30 18-23 * * * /etc/init.d/smb restart
```

### 5.10、 每星期六的晚上 11:00 pm 重启 smb

```
0 23 * * 6 /etc/init.d/smb restart
```

### 5.11、 每一小时重启 smb

```
* */1 * * * /etc/init.d/smb restart
```

### 5.12、 晚上 11 点到早上 7 点之间，每隔一小时重启 smb

```
* 23-7/1 * * * /etc/init.d/smb restart
```

### 5.13、 每月的 4 号与每周一到周三的 11 点重启 smb

```
0 11 4 * mon-wed /etc/init.d/smb restart
```

### 5.14、 一月一号的 4 点重启 smb

```
0 4 1 jan * /etc/init.d/smb restart
```

### 5.15、 每小时执行/etc/cron.hourly 目录内的脚本

```
01 * * * * root run-parts /etc/cron.hourly
```

## 6、 安装 crontab

如果系统中尚未安装 **crontab**，可以使用下面的命令手动安装，：

```
yum install -y vixie-cron  
yum install -y crontabs
```

说明：<sup>[11]</sup>

**vixie-cron** 软件包是 cron 的主程序；

**crontabs** 软件包是用来安装、卸装、或列举用来驱动 cron 守护进程的表格的程序。

## 7、 crontab 启动/关闭/查看详情

```
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 // 查看 crontab 服务状态
```

## 8、 设置 crontab 开机启动

查看 crontab 服务是否已设置为开机启动，执行命令：`ntsysv`，随后执行如下命令：

```
# 加入开机自动启动  
chkconfig --level 35 crond on
```

启动等级详解：

等级 0 表示：表示关机  
等级 1 表示：单用户模式  
等级 2 表示：无网络连接的多用户命令行模式  
等级 3 表示：有网络连接的多用户命令行模式  
等级 4 表示：不可用  
等级 5 表示：带图形界面的多用户模式  
等级 6 表示：重新启动

## 9、Linux 下任务调度

Linux 下的任务调度分为两类：系统任务调度和用户任务调度。<sup>[1]</sup>

### 9.1、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` 目录下就有一个 `crontab` 文件，这个是系统任务调度的配置文件。

初始时 `/etc/crontab` 文件包含下面几行：

```
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
```

图 9-1 默认 cron 系统配置文件

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。

不过，由于 root 并无法在客户端以 POP3 之类的软件收信，因此，通常推荐将 email 设置成自己的邮箱账号，好随时了解系统的状况！例如：

```
MAILTO=bluesking@my.host.name[5];  
# 第四行的 HOME 变量指定了在执行命令或脚本时使用的主目录。
```

## 9.2、 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。所有用户定义的 `crontab` 文件都被保存在 `/var/spool/cron` 目录中。

举例来说，`dmtsai` 使用 `crontab` 后，他的工作会被纪录到 `/var/spool/cron/dmtsai` 里头去！但请注意，不要使用 `vi` 直接编辑该文件，因为可能由于输入语法错误（尚未验证），会导致无法执行 `cron` 喔！另外，`cron` 执行的每一项工作都会被纪录到 `/var/log/cron` 这个登录档中，所以啰，如果你的 Linux 不知道有否被植入木马时，也可以搜寻一下 `/var/log/cron` 这个登录档呢！<sup>[5]</sup>

## 10、 查看 crontab 执行日志

通过日志文件 `/var/log/cron` 查看 `crontab` 执行日志信息。

```
tail -f /var/log/cron
```

```
Sep 12 14:38:56 Xdata-Base-01 crontab[9295]: (hadoop) BEGIN EDIT (hadoop) 开始修改crontab任务  
Sep 12 14:38:59 Xdata-Base-01 crontab[9295]: (hadoop) REPLACE (hadoop) 替换修改  
Sep 12 14:38:59 Xdata-Base-01 crontab[9295]: (hadoop) END EDIT (hadoop) 结束修改
```

图 10-1 crond 日志 1

```
(●●●): /etc >tail -f /var/log/cron  
May 6 03:43:02 VM_238_202_centos run-parts(/etc/cron.daily)[27676]: finished man-db.cron  
May 6 03:43:02 VM_238_202_centos run-parts(/etc/cron.daily)[27646]: starting mlocate  
May 6 03:43:04 VM_238_202_centos run-parts(/etc/cron.daily)[27687]: finished mlocate  
May 6 03:43:04 VM_238_202_centos anacron[23984]: Job 'cron.daily' terminated  
May 6 03:43:04 VM_238_202_centos anacron[23984]: Normal exit (1 job run)  
May 6 03:44:01 VM_238_202_centos CROND[27765]: (root) CMD (/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>&1 &)  
May 6 03:45:01 VM_238_202_centos CROND[27856]: (root) CMD (/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>&1 &)  
May 6 03:46:01 VM_238_202_centos CROND[27937]: (root) CMD (/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>&1 &)  
May 6 03:47:01 VM_238_202_centos CROND[28022]: (root) CMD (/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>&1 &)  
May 6 03:48:01 VM_238_202_centos CROND[28109]: (root) CMD (/usr/local/qcloud/stargate/admin/start.sh > /dev/null 2>&1 &)  
执行用户 执行命令
```

图 10-2 crond 日志 2

## 11、 使用者的权限设定

出于安全考虑，linux 允许限制 `crontab` 的使用者账号，配置使用限制的方式有：

- `/etc/cron.allow`  
将可以使用 `crontab` 的账号写入其中，若不在这个文件内的使用者则不可使用 `crontab`；

- `/etc/cron.deny`

将不可以使用 `crontab` 的账号写入其中，若未记录到这个文件当中的使用者，就可以使用 `crontab`。

以优先级来说，`/etc/cron.allow` 比 `/etc/cron.deny` 要优先，而判断上面，这两个文件只选择一个来限制而已，因此，建议你只要保留一个即可，免得影响自己在设定上面的判断！一般来说，**系统默认是保留 `/etc/cron.deny`**，你可以将不想让他执行 `crontab` 的那个使用者写入 `/etc/cron.deny` 当中，一个账号一行！<sup>[5]</sup>

详细的 `cron` 使用权限判断规则如下：<sup>[6][7]</sup>

- 若 `cron.allow` 文件和 `cron.deny` 文件不存在，只允许 `root` 用户使用 `cron`；
- 若只有 `cron.allow` 文件存在，仅允许 `cron.allow` 文件中**罗列**的用户使用 `cron`；
- 若只有 `cron.deny` 文件存在，仅允许 `cron.deny` 文件中**不存在**用户使用 `cron`；
- 如果 `cron.allow` 和 `cron.deny` 文件同时存在，并存在相悖的配置时，`cron.allow` 文件优先生效。

## 12、 `crond` 服务读取配置文件的位置

一般来说，`crond` 预设有三个地方会有执行脚本配置文件，他们分别是：<sup>[5]</sup>

- `/etc/crontab`
- `/etc/cron.d/*`
- `/var/spool/cron/*`

这三个地方中，跟系统的运作比较有关关系的两个配置文件是放在 `/etc/crontab` 文件内以及 `/etc/cron.d*` 目录内的文件，另外一个跟用户自己的工作比较有关的配置文件，就是放在 `/var/spool/cron/` 里面的文件群。现在我们已经知道了 `/var/spool/cron` 以及 `/etc/crontab` 的内容，那现在来瞧瞧 `/etc/cron.d` 里面的东西吧！<sup>[5]</sup>



```
(●'●'●): ~ > ll /etc/cron.d
total 4
-rw-r--r--. 1 root root 128 Aug  3 2017 0hourly
我的服务器中，/etc/cron.d目录下只有一个默认系统定时任务配置文件。
(●'●'●): ~ > cat /etc/cron.d/0hourly
# Run the hourly jobs
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
01 * * * * root run-parts /etc/cron.hourly
(●'●'●): ~ >
```

这个配置文件中的配置和我们经常使用的 `crontab -e` 相比，多出了三行环境设置，和 `/etc/crontab` 一样。

图 12-1 `/etc/cron.d` 下系统配置



当我们有一系列不限定特定用户的定时任务配置设计好了以后，可以在 `/etc/cron.d/` 目录下新添加一个定时任务文件，这样子比在 `/etc/crontab` 或者 `crontab -e` 去一行一行配置定时任务要方便，更改的时候可以直接覆盖文件。

而如果一些列相同执行时间的任务，可以使用 `run-parts` 命令，去设置定时执行指定目录下的定时任务。

## 13、 一些注意事项

### 13.1、 资源分配不均的问题

当同一时刻大量触发 `crontab` 定时任务的时候，系统性能或多或少都会被拖慢，所以使用定时任务的时候，尽量避免多个任务同一时刻触发。

举个例子，如果出现了同时设置了多个任务每隔 5 分钟执行，如何错开可能同时触发的任务呢？可以如下图，通过细粒化设置任务执行时间，错开任务。

```
[root@study ~]# vim /etc/crontab
1,6,11,16,21,26,31,36,41,46,51,56 * * * * root CMD1
2,7,12,17,22,27,32,37,42,47,52,57 * * * * root CMD2
3,8,13,18,23,28,33,38,43,48,53,58 * * * * root CMD3
4,9,14,19,24,29,34,39,44,49,54,59 * * * * root CMD4
```

图 13-1 错开四个每 5 分钟执行一次的任务 1<sup>[5]</sup>

经测试后发现，修改 `crond` 配置文件后，下一分钟的 01 秒，`crond` 服务就会重新加载（RELOAD）配置文件。`*/5 * * * * CMD` 配置等价于 `0-59/5 * * * * CMD`，其含义是，在 0 分钟执行一次命令 `CMD`，然后每停顿（间隔）5 分钟，执行一次命令 `CMD`，直到 59 分为止。在 0-59 分钟范围内，任务执行的时期为下图：

0	执行	15	执行	30	执行	45	执行
1		16		31		46	
2		17		32		47	
3		18		33		48	
4		19		34		49	
5	执行	20	执行	35	执行	50	执行
6		21		36		51	
7		22		37		52	
8		23		38		53	
9		24		39		54	
10	执行	25	执行	40	执行	55	执行
11		26		41		56	
12		27		42		57	
13		28		43		58	
14		29		44		59	

图 13-2 每隔 5 分钟任务执行周期

而 1-59/5 \* \* \* \* CMD 的含义为，在 1 分钟执行一次命令 CMD，然后每停顿（间隔）5 分钟，执行一次命令 CMD，直到 59 分为止。2-59/5 \* \* \* \* CMD 配置同理类推。由此可得，图 13-1 中的配置等价于下图配置：

```
0-59/5 * * * * echo 'CMD1'
1-59/5 * * * * echo 'CMD2'
2-59/5 * * * * echo 'CMD3'
3-59/5 * * * * echo 'CMD4'
4-59/5 * * * * echo 'CMD5'
```

图 13-3 错开四个每 5 分钟执行一次的任务 2

5 个任务执行周期排开，如下图所示：

0	CMD1	15	CMD1	30	CMD1	45	CMD1
1	CMD2	16	CMD2	31	CMD2	46	CMD2
2	CMD3	17	CMD3	32	CMD3	47	CMD3
3	CMD4	18	CMD4	33	CMD4	48	CMD4
4	CMD5	19	CMD5	34	CMD5	49	CMD5
5	CMD1	20	CMD1	35	CMD1	50	CMD1
6	CMD2	21	CMD2	36	CMD2	51	CMD2
7	CMD3	22	CMD3	37	CMD3	52	CMD3
8	CMD4	23	CMD4	38	CMD4	53	CMD4
9	CMD5	24	CMD5	39	CMD5	54	CMD5
10	CMD1	25	CMD1	40	CMD1	55	CMD1
11	CMD2	26	CMD2	41	CMD2	56	CMD2
12	CMD3	27	CMD3	42	CMD3	57	CMD3
13	CMD4	28	CMD4	43	CMD4	58	CMD4
14	CMD5	29	CMD5	44	CMD5	59	CMD5

图 13-4 5 个每隔 5 分钟任务执行周期

## 13.2、 取消不必要输出

定时任务需要执行的命令，不必要的输出可以重定向到 `/dev/null` 中。

## 13.3、 安全的检验

很多时候被植入木马都是以例行命令的方式植入的，所以可以藉由检查 `/var/log/cron` 的内容来视察是否有“非您设定的 cron 被执行了？”这个时候就需要小心一点喽！<sup>[5]</sup>

## 13.4、 周与日月不可同时并存

当同时配置了某月某日执行定时任务和星期几执行定时任务，可能会出现配置重复的情况，即既满足某月某日执行条件，又满足特定星期执行条件的情况。这时候，满足任意一个条件，都会触发定时任务。<sup>[8][9]</sup>由于使用上存在歧义性，故而使用时不推荐同时配置两种条件限制。

# 14、 Windows 定时任务

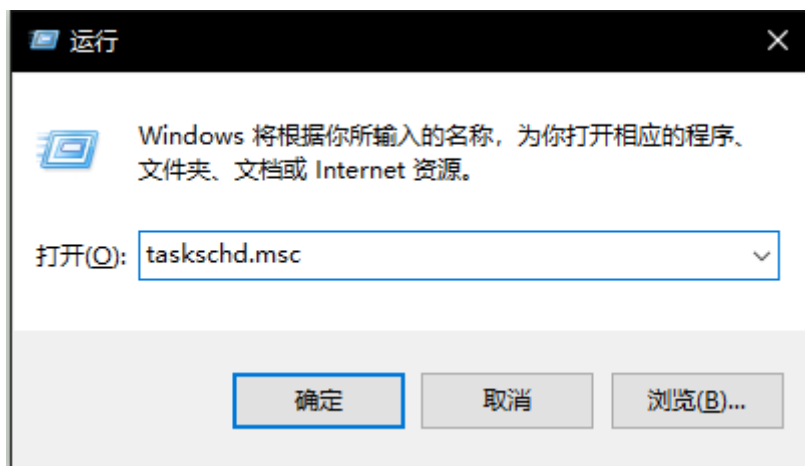


图 14-1 命令进入任务计划程序

win + r 输入 `taskschd.msc`，回车进入 **【任务计划程序】**。

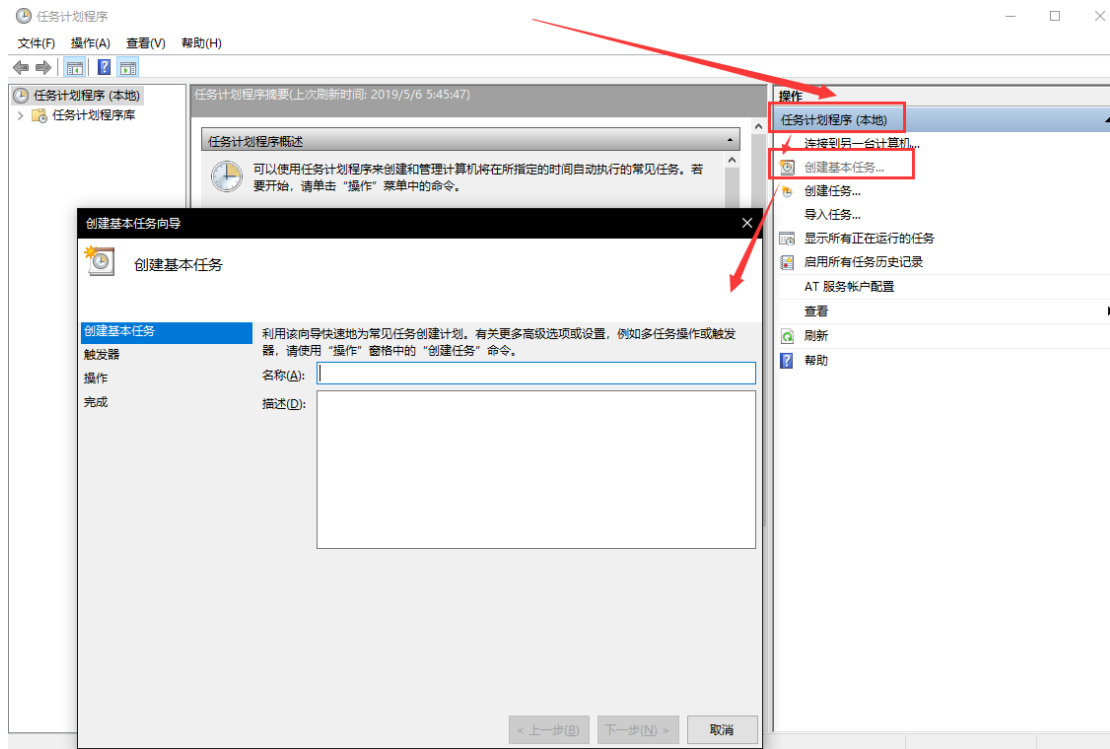


图 14-2 创建定时任务方式 1

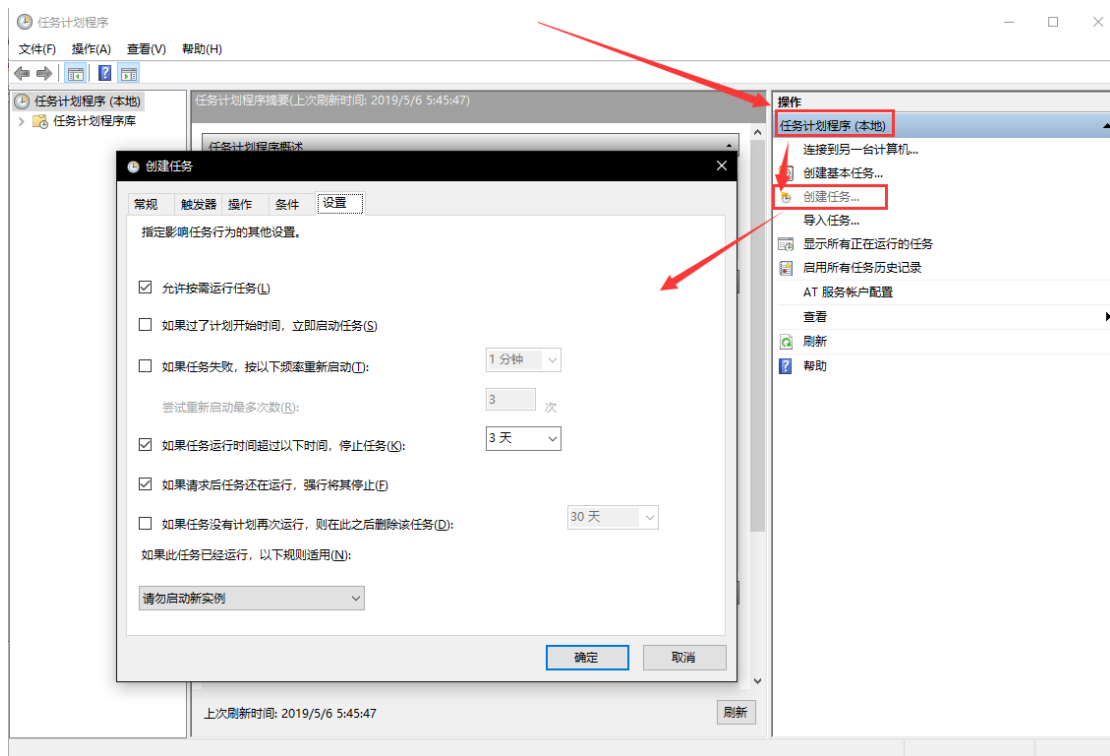


图 14-3 创建定时任务方式 2

在任务计划程序中，可以新建定时任务，这些任务除了设置执行时间周期，还有很多细节设置。相比较而言 `cron` 就显得简陋很多了。

## 15、 参考资料

- [1] crontab 命令 [2019-05-06]  
<http://man.linuxde.net/crontab>
- [2] Linux 定时任务 Crontab 命令详解 [2016-08-12]  
<https://www.cnblogs.com/intval/p/5763929.html>
- [3] 维基百科-Cron [2019-03-01]  
<https://zh.wikipedia.org/wiki/Cron>
- [4] 百度百科-crontab-w\_ou [2018-08-12]  
<https://baike.baidu.com/item/crontab/8819388?fr=aladdin>
- [5] 《鸟哥的 Linux 私房菜 基础学习篇 第四版》 [2018-10]
- [6] 赋 crontab -e 权限-warden2010 [2011-01-26]  
<https://blog.csdn.net/warden2010/article/details/6164531>
- [7] Linux crontab command-Computer Hope [2019-05-04]  
<https://www.computerhope.com/unix/ucrontab.htm>
- [8] linux 中的 crontab 中日期和星期条件-北方的忍冬草 [2018-02-08]  
<https://blog.csdn.net/dccmxj/article/details/79291792>
- [9] crontab -- tables for driving cron [2018-06-06]  
<https://www.freebsd.org/cgi/man.cgi?query=crontab&sektion=5>
- [10] Centos7 利用 crontab 定时执行任务及配置方法-从网络安全到运维的屌丝!  
[2018-08-15]  
<http://www.cnblogs.com/p0st/p/9482167.html>
- [11] Linux Crontab 安装使用详细说明-斌-逸风 [2016-02-22]  
<https://www.cnblogs.com/binblogs/p/5206789.html>