



定时的调度我们的脚本或者代码=》机制
crontab

定时调用

脚本/代码
完成某个任务
比如备份mysql数据

Linux

定时的调度我们的脚本或者代码=》机制
crontab

定时调用

mytask1.sh
date >> /tmp/mydate

```
*/1 * * * * ls -l /etc >>  
/tmp/to.txt
```

```
*/1 * * * *  
/home/mytask1.sh
```

1. 如果只是简单的任务，可以不用写脚本，直接在crontab中加入任务即可。
2. 对于比较复杂的任务，需要写脚本（Shell编程）

11.3 基本语法

crontab [选项]

11.3.1 常用选项

| | |
|----|--------------------|
| -e | 编辑crontab定时任务 |
| -l | 查询crontab任务 |
| -r | 删除当前用户所有的crontab任务 |

11.4快速入门

11.4.1 任务的要求

设置任务调度文件：/etc/crontab

设置个人任务调度。执行 crontab -e 命令。

接着输入任务到调度文件

如：*/1 * * * * ls -l /etc/ > /tmp/to.txt

意思说每小时的每分钟执行 ls -l /etc/ > /tmp/to.txt 命令

11.4.2 步骤如下

- 1) cron -e
- 2) */1 * * * * ls -l /etc >> /tmp/to.txt
- 3) 当保存退出后就程序。
- 4) 在每一分钟都会自动的调用 ls -l /etc >> /tmp/to.txt

11.4.3 参数细节说明

• 5个占位符的说明

I

| 项目 | 含义 | 范围 |
|------------------------|------------|----------------|
| 第一个“ <code>%M</code> ” | 一小时当中的第几分钟 | 0-59 |
| 第二个“ <code>%H</code> ” | 一天当中的第几小时 | 0-23 |
| 第三个“ <code>%D</code> ” | 一个月当中的第几天 | 1-31 |
| 第四个“ <code>%m</code> ” | 一年当中的第几个月 | 1-12 |
| 第五个“ <code>%w</code> ” | 一周当中的星期几 | 0-7（0和7都代表星期日） |

• 特殊符号的说明

| 特殊符号 | 含义 |
|------|--|
| * | 代表任何时间。比如第一个“*”就代表一小时中每分钟都执行一次的意思。 |
| , | 代表不连续的时间。比如“0 8,12,16 * * * 命令”，就代表在每天的8点0分，12点0分，16点0分都执行一次命令 |
| - | 代表连续的时间范围。比如“0 5 * * 1-6 命令”，代表在周一到周六的凌晨5点0分执行命令 |
| */n | 代表每隔多久执行一次。比如“*/10 * * * * 命令”，代表每隔10分钟就执行一遍命令 |

• 特定时间执行任务案例

| 时间 | 含义 |
|-----------------|--|
| 45 22 * * * 命令 | 在22点45分执行命令 |
| 0 17 * * 1 命令 | 每周1的17点0分执行命令 |
| 0 5 1,15 * * 命令 | 每月1号和15号的凌晨5点0分执行命令 |
| 40 4 * * 1-5 命令 | 每周一到周五的凌晨4点40分执行命令 |
| */10 4 * * * 命令 | 每天的凌晨4点，每隔10分钟执行一次命令 |
| 0 0 1,15 * 1 命令 | 每月1号和15号，每周1的0点0分都会执行命令。注意：星期几和几号最好不要同时出现，因为他们定义的都是天。非常容易让管理员混乱。 |

11.5.1 案例 1：每隔 1 分钟，就将当前的日期信息，追加到 /tmp/mydate 文件中

- 1) 先编写一个文件 /home/mytask1.sh

```
date >> /tmp/mydate
```
- 2) 给 mytask1.sh 一个可以执行权限

```
chmod 744 /home/mytask1.sh
```
- 3) crontab -e
- 4) */1 * * * * /home/mytask1.sh
- 5) 成功

11.5.2 案例 2：每隔 1 分钟，将当前日期和日历都追加到 /home/mycal 文件中

- 1) 先编写一个文件 /home/mytask2.sh

```
date >> /tmp/mycal  
cal >> /tmp/mycal
```
- 2) 给 mytask1.sh 一个可以执行权限

```
chmod 744 /home/mytask2.sh
```
- 3) crontab -e
- 4) */1 * * * * /home/mytask2.sh
- 5) 成功

11.5.3 案例 3：每天凌晨 2:00 将 mysql 数据库 testdb，备份到文件中 mydb.bak。

- 1) 先编写一个文件 /home/mytask3.sh

```
/usr/local/mysql/bin/mysqldump -u root -proot testdb > /tmp/mydb.bak
```
- 2) 给 mytask3.sh 一个可以执行权限

-
- ```
chmod 744 /home/mytask3.sh
```
- 3) crontab -e
  - 4) 0 2 \* \* \* /home/mytask3.sh
  - 5) 成功

### 11.6 crond 相关指令：

- 1) conrtab -r：终止任务调度。
- 2) crontab -l：列出当前有那些任务调度
- 3) service crond restart [重启任务调度]

`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 //查看服务状态
```