

安排周期性任务

内容提要

- 1. 了解周期性任务的安排方法
- 2. 学会使用 `crontab` 命令装载用户自己的 `cron` 任务
- 3. 理解系统中安排 `cron` 任务的方法
- 4. 理解系统中安排 `anacron` 任务的方法

安排周期性任务概述

系统中有两个守护进程 `cron` 和 `anacron` 用于自动执行周期性任务。`cron` 与 `anacron` 是完全不同的两个用于定期执行任务的守护进程。

- `cron` 假定服务器是 `24*7` 全天候运行的，当系统时间变化或有一段关机时间就会遗漏这一时间段应该执行的`cron`任务。
- `anacron` (`anachronistic cron`) 是 `cron` 的一个连续时间版本，它不会因为时间不连续而导致的任务不执行。

`anacron` 是针对非全天候运行而设计的，当 `anacron` 发现时间不连续时，也会执行这一时间段内该执行的任务，这样就不会遗漏计划任务的执行。

每个用户都可以安排自己的 `cron` 任务。超级用户可以管理系统的 `cron` 任务和 `anacron` 任务。

安排用户自己的周期性任务

`/var/spool/cron` 目录 `cron` 守护进程还将搜索 `/etc/crontab` 文件，这个文件是系统安装时设置好的自动安排的进程任务的 `crontab` 文件。

cron 简介

`cron` 守护进程启动以后，它将首先检查是否有用户设置了 `crontab` 文件，`cron` 守护进程首先会搜索 `/var/spool/cron` 目录，寻找以 `/etc/passwd` 文件中的用户名命名的 `crontab` 文件，被找到的这种文件将载入内存。例如一个用户名为 `osmond` 的用户，它所对应的 `crontab` 文件就应该是 `/var/spool/cron/osmond`。也就是说，以该用户命名的 `crontab` 文件存放在 `/var/spool/cron` 目录下。如果 `cron` 守护进程没有发现相应的 `crontab` 文件就转入"休眠"状态，释放系统资源，所以该后台进程占用资源极少。`cron` 守护进程每分钟唤醒一次，当 `crontab` 中的时间和日期与系统的当前时间和日期相同时，就执行相应的 `cron` 任务。`cron` 任务执行结束后，任何输出都将作为邮件发送给安排 `cron` 任务的所有者，或者是 `crontab` 中 `MAILTO` 环境变量中指定的用户。

`cron` 守护进程的执行不需要用户干涉；只需要用户安排 `crontab` 文件，在该文件中要执行的时间和命令序列，下面介绍 `crontab` 命令。

crontab 命令

每个用户都可以设置自己的 `crontab` 文件以便执行用户自己需要的自动运行的任务。用户自己的 `crontab` 文件位于 `/var/spool/cron/` 目录，但用户不能直接编辑这些文件，用户必须使用 `crontab` 命令编辑它。

`crontab` 命令用于安装、删除或者列出用于驱动 `cron` 后台进程的 `crontab` 任务。`crontab` 的命令格式如下：

格式1: `crontab [-u user] file`
格式2: `crontab [-u user] [-l|-r|-e]`

下表是 `crontab` 命令的选项说明。

选项	说明
<code>-u user</code>	指定具体哪个用户的 <code>crontab</code> 文件将被修改。如果不指定该选项， <code>crontab</code> 将默认认为是操作者本人的 <code>crontab</code> ，也就是执行该 <code>crontab</code> 命令的用户的 <code>crontab</code> 文件将被修改。当使用了 <code>su</code> 命令后执行 <code>crontab</code> 命令就应该指定此参数，以免出现混乱。
<code>file</code>	是一个已经编辑好的 <code>crontab</code> 文件。如果使用" <code>-</code> "号作为文件名，那就意味着使用标准输入。 用于将一个已经编辑好的 <code>crontab</code> 文件装载到 <code>/var/spool/cron/</code> 目录。
<code>-l</code>	该选项将使在标准输出上显示当前的 <code>crontab</code> 。
<code>-r</code>	删除当前的 <code>crontab</code> 任务。
<code>-e</code>	使用 <code>VISUAL</code> 或 <code>EDITOR</code> 环境变量指定的编辑器编辑当前的 <code>crontab</code> 文件。当结束编辑离开时，编辑后的文件将自动安装。

`crontab` 文件中的每一行格式为：

minute hour day-of-month month-of-year day-of-week [username] commands

每行中都由用空格间隔的七个字段组成。下表说明了各个字段的含义和取值范围。

字段	说明	取值范围
minute	一小时中的哪一分钟	0~59
hour	一天中的哪个小时	0~23
day-of-month	一月中的哪一天	1~31
month-of-year	一年中的哪一月	1~12
day-of-week	一周中的哪一天	0~7 (0和7均表示周日)
username	以指定的用户身份执行 <code>commands</code> ，省略此字段时表示以安排本任务的用户身份执行 <code>commands</code> 。	

commands	执行的命令（可以是多行命令或者是脚本调用）。	
----------	------------------------	--

下面重点说明一下前五个时间字段的语法：

- 不能为空，可以使用通配符*表示任何时间。
 - 可以指定多个值，它们之间用逗号间隔。例如：1,3,7。
 - 可以指定时间段，用减号间隔。例如：0-6。
 - 可以用/n表示步长。例如：8-18/2表示时间序列8,10,12,14,16,18。
1. 可以在crontab文件中定义并使用环境变量。
 2. 如果要执行多条命令可以将其写入一个脚本中，在commands字段中只需写脚本文件名即可。

自动安排进程任务举例

执行如下命令安排root用户的crontab任务：

```
# crontab -e
```

```
# 每月1日和15日凌晨1:30使系统进入维护状态，重新启动系统
30 1 1,15 * * shutdown -r +10 > /dev/null 2>&1
# 每天凌晨的2点删除/ftp/incoming/temp目录下的所有文件
00 02 * * * rm -rf /ftp/incoming/temp/*
# 每天凌晨的1点删除/tmp目录下7天没有被修改过的所有子目录
# 不包括/tmp当前目录和lost+found目录
00 01 * * * find /tmp ! -name . ! -name lost+found \
-type d -mtime +7 -exec /bin/rm -rf {} ';'
# 每天早上8点将 /var/log/secure 文件内容发送给 osmond@mydomain.com
0 8 * * * mail osmond@mydomain.com < /var/log/secure
# 每隔两小时将命令 netstat -a 的输出发送给 osmond@mydomain.com
0 */2 * * * netstat -a | mail osmond@mydomain.com
# 每天7-19点开放samba服务（其他服务也可以用这种方式设置）
0 7 * * * service smb start
0 19 * * * service smb stop
```

使用如下命令检查 crontab 任务：

```
# crontab -l
```

使用如下命令删除已经安排的 crontab 任务：

```
# crontab -r
```

控制安排 cron 任务的人员

是否每个用户都可以安排 cron 任务，需要受如下两个文件的限制：

- /etc/cron.allow
 - /etc/cron.deny
1. 当用户每次安排 cron 任务时，系统会先查找 /etc/cron.allow 文件，若该文件存在，则只有包含在此文件中的用户允许使用 cron。
 2. 若 /etc/cron.allow 文件不存在，系统继续查找 /etc/cron.deny 文件，若该文件存在，则只有包含在此文件中的用户禁止使用 cron。
 3. 有一个例外：无论 root 是否包含在 /etc/cron.allow 文件或 /etc/cron.deny 文件中，root 都可以使用 cron。

/etc/cron.allow 文件和 /etc/cron.deny 文件的格式很简单，每行只能包含一个用户名，且不能有空格字符。

安排系统的周期性任务

系统的 cron 任务

cron 守护进程在搜索 /var/spool/cron 目录下用户的 crontab 文件的同时，还将搜索 /etc/crontab 文件，这个文件是系统安装时设置好的自动安排的进程任务的 crontab 文件。这为系统管理员安排 cron 任务提供了方便。

CentOS 默认的 /etc/crontab 文件的内容为：

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

/etc/cron.daily、/etc/cron.monthly、/etc/cron.weekly 和 /etc/cron.hourly 是四个目录，分别放置系统每天、每个月、每周和每个小时要执行的任务的脚本

文件。

run-parts 命令的作用是运行指定目录下的所有可执行文件（包括程序或脚本）。

以第一行任务为例：每小时的第1分钟以root身份执行程序 **run-parts** 命令，命令参数是 **/etc/cron.hourly**，即执行该目录下的每一个脚本文件。因此，系统管理员只需要将每小时执行的任务编写一个脚本放在 **/etc/cron.hourly** 目录下即可。其他目录的使用类似，不再赘述。

- 通常系统管理员无需编辑 **/etc/crontab** 文件，只要直接编写 **/etc/cron.daily**、**/etc/cron.monthly**、**/etc/cron.weekly** 和 **/etc/cron.hourly** 目录下的脚本文件即可。
- 一旦编辑了 **/etc/crontab** 文件，为了使之立即生效，需要执行如下的命令：

```
# service crond restart
```

系统的 anacron 任务

anacron 在 **/var/spool/anacron** 中保留时间戳文件，记录作业运行的时间。当 **anacron** 运行时，它检查自作业上一次运行以来是否已经经过了所需的天数，如果需要，就运行作业。

anacron 的任务表存储在 **/etc/anacrontab** 文件中。CentOS 默认的 **/etc/anacrontab** 文件的内容为：

```
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# 若1天之内没有运行“日任务”，则 65 分钟之后运行之
1 65 cron.daily run-parts /etc/cron.daily
# 若7天之内没有运行“周任务”，则 70 分钟之后运行之
7 70 cron.weekly run-parts /etc/cron.weekly
# 若30天之内没有运行“月任务”，则 75 分钟之后运行之
30 75 cron.monthly run-parts /etc/cron.monthly
```

此处 **/etc/anacrontab** 的配置目的是确保每日、每周、和每月的 **cron** 任务都被运行。

/etc/anacrontab 文件格式与 **/etc/crontab** 略有不同。与 **/etc/crontab** 一样，**/etc/anacrontab** 可以包含环境设置。每个任务有四个字段：

```
period delay job-identifier command
周期 延迟 作业标识符 命令
```

说明：

- 周期（**period**）：命令执行的时间间隔（天数）
- 延迟（**delay**）：延迟是在作业符合运行条件之后，到实际启动它之前等待的时间（分钟）。可以使用这个设置防止在系统启动时集中执行作业。
- 作业标识符（**job-identifier**）：任务的描述，用在 **anacron** 的消息中，并作为作业时间戳文件的名称，只能包括非空白的字符（除斜线外）。
- 命令（**command**）：实际执行的任务。

对于每项任务，**anacron** 先判定该任务是否已在配置文件的周期字段中指定的期间内被执行了。如果它在给定周期内还没有被执行，**anacron** 会等待延迟字段中指定的分钟数，然后再次尝试执行命令字段中指定的命令。

当任务完成后，**anacron** 会将此日期记录在 **/var/spool/anacron** 目录的时间戳（**Timestamp**）文件中，默认的时间戳文件有三个：**cron.daily**、**cron.monthly** 和 **cron.weekly**。时间戳文件的记录内容都很简单，下面是作业标识符为 **cron.daily** 的时间戳文件示例：

```
# ls /var/spool/anacron/
cron.daily cron.monthly cron.weekly

# cat /var/spool/anacron/cron.daily
20071219
```

/etc/crontab 和 **/etc/anacrontab** 都通过直接编辑进行更新。不使用 **crontab** 命令更新这些文件或 **/etc/cron.d** 目录中的文件。

参考

- http://www.chinalinuxpub.com/doc/manual/9.0/rhl-cg-zh_CN-9/ch-autotasks.html [<http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jaGlueWxpbmV4cHVhLmNvbS9kb2MvbWFudWFsLzkuMC9yaGwtY2ctemhfQ04tO S9jaC1hdXRvdGFza3MuaHRtbA %3D%3D>]
- 显示源文件
- 登录