第11章 Linux实操篇 - 定时任务调度

在Linux系统运维工作中，定时任务调度是一项核心技能，它能够帮助运维人员实现任务的自动化执行，大大提升系统维护效率和稳定性。本章将详细介绍Linux系统中两种常见的定时任务调度工具：crond和at，包括它们的原理、语法、使用方法以及实际应用案例。

11.1 crond 任务调度

11.1.1 概述

任务调度是指系统在特定时间执行预先设定的命令或程序。在 Linux 环境下，任务调度主要分为两类：

- **系统工作**：一些关乎系统稳定和安全的重要任务，需要周期性地执行，如病毒扫描、系统资源监控等。
- **个别用户工作**：用户根据自身业务需求，希望系统在特定时间执行某些程序，例如对 MySQL 数据库的定期备份、日志文件的清理等。

可以通过一个简单的示意图来理解任务调度的流程：系统按照设定的时间规则，从任务队列中取出相应的任务并执行，就像一个有条不紊的管家，按时完成各项既定事务。

11.1.2 基本语法

crontab 命令用于设置和管理定时任务，其基本语法为：

```
crontab [选项]
```

11.1.3 常用选项

选项	说明
-e	编辑当前用户的 crontab 文件，用于添加、修改定时任务
-l	列出当前用户的所有定时任务
-r	删除当前用户的所有定时任务

11.1.4 快速入门

1. **设置任务调度文件**：系统级别的任务调度文件位于`/etc/crontab`，通常用于设置一些全局性的、对系统运行至关重要的定时任务。
2. **设置个人任务调度**：用户可以通过执行`crontab -e`命令来编辑自己的定时任务。在进入编辑界面后，按照特定的格式输入任务内容。例如：

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

这行配置表示每小时的每分钟都会执行`ls -l /etc/ > /tmp/to.txt`命令，即将`/etc`目录下的文件列表信息输出到`/tmp/to.txt`文件中。

参数细节说明

1. **5 个占位符的说明**：从左到右依次为分钟（0 - 59）、小时（0 - 23）、日期（1 - 31）、月份（1 - 12）、星期（0 - 7，0 和 7 都代表星期日）。每个占位符都用于指定任务执行的时间条件。
2. 特殊符号的说明
   - **星号（*）**：表示匹配该字段的所有可能值，如`* * * * *`表示每分钟都执行任务。
   - **逗号（,）**：用于分隔多个值，如`0 8,14,20 * * *`表示每天 8 点、14 点和 20 点执行任务。
   - **减号（-）**：表示一个范围，如`0 9-17 * * *`表示每天 9 点到 17 点之间，整点执行任务。
   - **斜杠（/）**：用于指定时间间隔，如`*/10 * * * *`表示每 10 分钟执行一次任务。
3. **特殊时间执行案例**：例如，`0 0 1 1 *`表示每年 1 月 1 日的 0 点执行任务；`0 2 * * 1`表示每周一的凌晨 2 点执行任务。

11.1.5 应用实例

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 * * * * date >> /tmp/mydate
```

1. 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。

   - 步骤 1：使用`vim /home/my.sh`命令创建并编辑脚本文件，写入内容`date >> /home/mycal`和`cal >> /home/mycal`。
   - 步骤 2：给`my.sh`增加执行权限，执行`chmod u+x /home/my.sh`。
   - 步骤 3：通过`crontab -e`命令增加定时任务`*/1 * * * * /home/my.sh`。
2. 案例 3

   ：每天凌晨 2:00 将 MySQL 数据库

   ```
   testdb
   ```

   ，备份到文件中。

   - 步骤 1：执行`crontab -e`命令进入编辑界面。
   - 步骤 2：添加定时任务`0 2 * * * mysqldump -u root -proot testdb > /home/db.bak`，注意这里的`-proot`是数据库密码，实际使用时请根据真实密码进行替换。

11.1.6 crond 相关指令

1. `crontab -r`：终止当前用户的任务调度，即删除所有已设置的定时任务。
2. `crontab -l`：列出当前用户所有已设置的任务调度，方便用户查看和管理。
3. `service crond restart`：重启任务调度服务，在修改了`crontab`文件或者调整了相关配置后，通常需要重启服务使更改生效。

11.2 at 定时任务

11.2.1 基本介绍

1. `at` 命令用于设置一次性定时计划任务，它依赖于 `atd` 守护进程，该进程以后台模式运行，不断检查作业队列，以确定是否有任务需要执行。
2. 默认情况下，`atd` 守护进程每 60 秒检查一次作业队列。当发现有作业时，会进一步检查作业的运行时间，若时间与当前时间匹配，则立即执行该作业。
3. 与 `crond` 不同，`at` 命令设置的任务是一次性的，执行完成后便不再重复执行。
4. 在使用 `at` 命令之前，务必确保 `atd` 进程已经启动。可以使用以下指令进行检查：

```
ps -ef | grep atd
```

通过一个简单的示意图可以更好地理解 `at` 命令的工作流程：用户通过 `at` 命令提交任务和执行时间，任务被加入作业队列，`atd` 守护进程定时检查队列，在合适的时间取出任务并执行。

11.2.2 at 命令格式

```
at [选项] [时间]
```

在输入完任务内容后，按下 `Ctrl + D` 组合键结束 `at` 命令的输入，此时会输出两次确认信息，表示任务已成功提交。

11.2.3 at 命令选项

选项	说明
-f	指定包含任务命令的文件，而不是在命令行中直接输入命令
-l	列出当前用户所有待执行的 `at` 任务，等同于 `atq` 命令
-d	删除指定编号的 `at` 任务，等同于 `atrm` 命令

11.2.4 at 时间定义

`at` 命令支持多种灵活的时间定义方式：

1. **指定具体时间**：接受当天的 `hh:mm`（小时：分钟）格式的时间指定。若该时间已过去，则任务会被安排在第二天执行。例如：`04:00`。
2. **模糊时间指定**：可以使用 `midnight`（深夜）、`noon`（中午）、`teatime`（饮茶时间，一般是下午 4 点）等模糊词语来指定时间。
3. **12 小时计时制**：采用 12 小时计时制，在时间后面加上 `AM`（上午）或 `PM`（下午）来明确时间是上午还是下午。例如：`12pm`。
4. **指定日期**：指定命令执行的具体日期，格式可以是 `month day`（月 日）、`mm/dd/yy`（月 / 日 / 年）或 `dd.mm.yy`（日。月. 年），日期必须跟在指定时间的后面。例如：`04:00 2021-03-1`。
5. **相对计时法**：指定格式为 `now + count time-units`，`now` 表示当前时间，`time-units` 是时间单位，可以是 `minutes`（分钟）、`hours`（小时）、`days`（天）、`weeks`（星期），`count` 是时间的数量。例如：`now + 5 minutes` 表示 5 分钟后执行任务。
6. **使用关键词**：直接使用 `today`（今天）、`tomorrow`（明天）来指定完成命令的时间。

11.2.5 应用实例

1. **案例 1**：2 天后的下午 5 点执行 `/bin/ls /home` 命令。

```
at 17:00 + 2 days
/bin/ls /home
Ctrl + D
```

1. **案例 2**：使用 `atq` 命令来查看系统中没有执行的工作任务。

```
atq
```

1. **案例 3**：明天 17 点钟，输出时间到指定文件 `/root/date100.log` 内。

```
at 17:00 tomorrow
date > /root/date100.log
Ctrl + D
```

1. **案例 4**：2 分钟后，输出时间到指定文件 `/root/date200.log` 内。

```
at now + 2 minutes
date > /root/date200.log
Ctrl + D
```

1. **案例 5**：删除已经设置的任务，使用 `atrm` 命令加上任务编号。例如，删除 `job` 队列中编号为 4 的任务：

```
atrm 4
```

通过对 `crond` 和 `at` 定时任务调度工具的学习，你已经掌握了在 Linux 系统中实现任务自动化执行的核心技能。在实际运维工作中，根据具体的业务需求和场景，灵活选择合适的工具和配置方式，能够极大地提高工作效率，确保系统的稳定运行。

课后练习

一、选择题

1. 以下哪个命令用于编辑当前用户的 crontab 文件？（）
   A. `crontab -l`
   B. `crontab -r`
   C. `crontab -e`
   D. `crontab -f`
2. 在 crontab 文件中，`0 2 * * 1` 表示什么含义？（）
   A. 每天凌晨 2 点执行任务
   B. 每周一凌晨 2 点执行任务
   C. 每月 1 号凌晨 2 点执行任务
   D. 每年 1 月 1 号凌晨 2 点执行任务

3. 以下关于 at 命令的说法，错误的是（ ）
   A. at 命令用于设置一次性定时计划任务
   B. atd 守护进程默认每 30 秒检查一次作业队列
   C. at 命令设置的任务执行完成后不再重复执行
   D. 可以使用 `atq` 命令查看系统中没有执行的工作任务
4. 要在每周一到周五的下午 3 点执行一个脚本 `/home/user/backup.sh`，crontab 文件中应添加的内容是（ ）
   A. `0 15 * * 1 - 5 /home/user/backup.sh`
   B. `0 15 1 - 5 * * /home/user/backup.sh`
   C. `0 15 * 1 - 5 * /home/user/backup.sh`
   D. `0 15 * * 1,2,3,4,5 /home/user/backup.sh`
5. 以下哪个不是 at 命令支持的时间定义方式？（ ）
   A. `now + 3 hours`
   B. `next week`
   C. `10:00 AM`
   D. `tomorrow`

二、填空题

1. 在 crontab 文件中，`*` 符号表示__。
   答案：匹配该字段的所有可能值
   解析：在 crontab 的时间设置中，`*` 是通配符，在分钟、小时、日期、月份、星期等字段使用时，表示该字段的所有可能取值。
2. 若要重启任务调度服务 crond，应执行的命令是__。
   答案：`service crond restart`（在一些系统中也可以使用 `systemctl restart crond`）
   解析：在使用 SysVinit 初始化系统的旧版本系统中，使用 `service crond restart` 来重启 `crond` 服务；在使用 Systemd 初始化系统的较新版本中，`systemctl restart crond` 是常用的重启命令。
3. at 命令中，用于指定包含任务命令的文件的选项是__。
4. 在 crontab 文件中，`0 0 1 1 *` 表示的含义是__。
5. 若要列出当前用户所有待执行的 at 任务，可使用__命令。

三、简答题

1. 简述 crond 和 at 定时任务调度的主要区别。
2. 假设你需要每天凌晨 3 点对数据库进行备份，备份命令为 `mysqldump -u root -p123456 mydb > /backup/mydb_backup.sql`，请写出在 crontab 中设置该定时任务的步骤和内容。
3. 假如你要在明天上午 10 点执行一个脚本 `/root/check_system.sh`，请使用 at 命令完成设置，并写出检查该任务是否设置成功的命令。
4. 解释 crontab 文件中 `30 23 * * 1 - 5 /usr/local/bin/log_clean.sh` 这行配置的含义。
5. 当使用 crontab 设置定时任务后，发现任务没有按照预期执行，可能有哪些原因？请至少列举 3 点。

四、实操题

1. 在 Linux 系统中，使用 crontab 设置一个定时任务，每 15 分钟将系统当前的内存使用情况（使用 `free -h` 命令获取）追加到 `/var/log/memory.log` 文件中。
2. 使用 at 命令设置一个任务，在 1 小时后执行 `echo "This is a test task." > /tmp/test.txt` 命令，并检查任务是否设置成功。