

一、计划任务概述

1.1 什么是计划任务？

- **定义：**在特定时间自动执行预先设定的命令或脚本。
- **核心用途：**
 - 延迟执行（如30秒后重启服务）。
 - 周期性任务（如每日备份数据库、每小时同步数据）。
 - 脱机任务（如无人值守服务器运维）。

1.2 Linux中的两大工具

工具	适用场景	时间粒度	特点
at	一次性/延迟任务	秒级精准	简单、轻量级，无需持久化守护进程
cron	周期性重复任务	分钟级调度	复杂时间表达式，需长期运行守护进程

```
[root@xnha ~]# systemctl status atd.service
● atd.service - Job spooling tools
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; vendor preset: enable)
   Active: active (running) since Tue 2025-03-11 20:55:52 EDT; 1 day 14h ago
 Main PID: 1203 (atd)
    Tasks: 1 (limit: 23821)
   Memory: 1.5M
    CGroup: /system.slice/atd.service
            └─1203 /usr/sbin/atd -f
```

```
[root@xnha ~]# systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enable)
   Active: active (running) since Tue 2025-03-11 20:55:52 EDT; 1 day 14h ago
 Main PID: 1206 (crond)
    Tasks: 1 (limit: 23821)
   Memory: 3.1M
    CGroup: /system.slice/crond.service
            └─1206 /usr/sbin/crond -n
```

二、at 命令详解

2.1 基本语法

```
at [选项] 时间表达式 [命令]
```

- **提交任务：**输入命令后按 `Ctrl+D` 提交。
- **查看任务：**`atq`（列出所有任务）。
- **删除任务：**`atrm <任务ID>`。

2.2 时间表达式

- 灵活时间格式:

```
now          # 立即执行
now + 5min   # 5分钟后
02:30 tomorrow/today # 次日2:30
2025-12-31 23:59 # 绝对时间
```

- 示例:

```
[root@xnha ~]# at now +1 minutes
warning: commands will be executed using /bin/sh
该警告意味着计划任务必须使用内置命令
```

- 显示加载

```
[root@xnha ~]# at now +1 minutes
warning: commands will be executed using /bin/sh
at> SHELL=/bin/bash
at> echo "当前shell: $(which bash)" >> /tmp/at-test.txt
at> <EOT>
job 1 at Thu Mar 13 10:14:00 2025

[root@xnha tmp]# cat at-test.txt
当前shell/usr/bin/bash
```

2.3 实战场景入门

场景1: 延迟执行高风险操作-删除

```
[root@xnha ~]# at now +1 minutes
warning: commands will be executed using /bin/sh
at> rm -rf /tmp/at-test.txt
at> <EOT>
job 2 at Thu Mar 13 10:29:00 2025
```

场景2: 固定时间任务-每天凌晨2点备份数据库

```
[root@xnha ~]# at 02:00 tomorrow
warning: commands will be executed using /bin/sh
at> tar -zcf /backup/db_$(date +%F).tar.gz /var/lib/mysql
at> <EOT>
job 3 at Fri Mar 14 02:00:00 2025
```

date命令解释:

`date` 的格式化符号以 `%` 开头, 后接特定字母表示不同时间单位。以下是常用符号及其含义:

符号	含义	示例值
%Y	四位年份 (如 2023)	2023
%m	两位月份 (如 07 表示七月)	07
%d	两位日期 (如 28 表示28日)	28
%H	24小时制小时 (如 14 表示下午2点)	14
%I	12小时制小时 (如 02 表示凌晨2点)	02
%M	两位分钟 (如 30 分钟)	30
%S	两位秒 (如 45 秒)	45
%a	星期几缩写 (如 Fri 表示周五)	Fri
%A	星期几全称 (如 Friday)	Friday
%b	月份缩写 (如 Jul 表示七月)	Jul
%B	月份全称 (如 July)	July
%F	ISO 8601日期格式 (年-月-日)	2023-07-28
%T	24小时制时间格式 (时:分:秒)	14:30:00
%R	12小时制时间格式 (带AM/PM)	02:30 PM
%Z	时区偏移 (如 UTC+8)	UTC+8
%s	自纪元起经过的秒数 (Linux时间戳)	1687923000

场景3：复杂时间格式

支持灵活时间描述：

- now：立即执行
- today / tomorrow：当天/次日
- YYYY-MM-DD HH:MM：绝对时间
- +N days / -N minutes：相对时间

示例：2025年12月25日23:59执行关机：

```
at 23:59 Dec 25 2025
shutdown -h now
Ctrl+D 提交
```

2.4 管理命令

命令	作用	示例
<code>atq</code>	查看所有待执行任务	<code>atq -l</code> (详细列表)
<code>atrm</code>	删除任务	<code>atrm 3</code> (删除ID=3的任务)
<code>at -c</code>	查看任务内容	<code>at -c 5</code> (查看ID=5的任务)

```
[root@xnha ~]# atq
3 Fri Mar 14 02:00:00 2025 a root
[root@xnha ~]# at now +5 minutes
warning: commands will be executed using /bin/sh
at> touch /tmp/5
at> <EOT>
job 4 at Thu Mar 13 10:45:00 2025
[root@xnha ~]# atq
3 Fri Mar 14 02:00:00 2025 a root
4 Thu Mar 13 10:45:00 2025 a root
[root@xnha ~]# at -c 4
```

三、cron命令详解

3.1 cron 核心功能

`cron` 是 Linux/Unix 系统中用于**周期性执行任务**的守护进程，适合以下场景：

- **定时备份**：每日/每周自动备份数据。
- **日志清理**：定时删除过期日志文件。
- **数据同步**：每小时同步服务器间数据。
- **监控报警**：检测服务状态并发送通知。

3.2 配置文件

- **系统级**： `/etc/crontab` 编辑模板（所有用户生效）。

```
[root@xnha etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# 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
```

- 语法格式：

```
* * * * * 用户名 命令
分钟 小时 日 月 星期
```

3.3 时间表达式

- 字段说明：

字段	允许值	特殊符号
分钟	0-59	<code>*</code> , <code>-</code> , <code>/</code> , <code>?</code>
小时	0-23	同上
日	1-31	<code>*</code> , <code>-</code> , <code>/</code> , <code>L</code>
月	1-12	<code>*</code> , <code>-</code> , <code>/</code>
星期	0-7	<code>*</code> , <code>-</code> , <code>L</code>

- 示例：

```
0 2 * * * root /usr/bin/systemctl restart nginx #每日凌晨2点重启Nginx
30 4 * * 1-5 user1 /home/user1/backup.sh # 工作日每天4:30备份
```

3.3 管理命令详解

`crontab` 是 `cron` 的配置工具，用于安装、删除、编辑用户的计划任务。以下是常用命令：

命令	作用	示例
<code>crontab -e</code>	编辑当前用户任务	<code>crontab -e</code>
<code>crontab -l</code>	列出当前用户任务	<code>crontab -l</code>
<code>crontab -r</code>	删除当前用户所有任务	<code>crontab -r</code>
<code>crontab -u username</code>	管理其他用户任务（需sudo）	<code>sudo crontab -e user1</code>
<code>crontab -i</code>	编辑时交互提示	<code>crontab -i</code>
<code>crontab -d</code>	删除用户任务（旧版用法）	<code>crontab -d</code>

1. `crontab -e`：编辑任务

- 功能：
打开当前用户的 `~/.crontab` 文件，支持 `vi` 或 `nano` 编辑器（取决于系统默认）。
- 使用场景：
添加、修改或删除个人定时任务。
- 示例：
bash

```
# 添加每日备份任务
crontab -e
```

cron

```
0 3 * * * touch /tmp/test2.txt
```

保存退出后，任务自动生效。

2. `crontab -l`：列出任务

- **功能：**
显示当前用户的所有计划任务（格式化输出）。
- **示例：**
bash

```
[root@xnha ~]# crontab -l
0 3 * * * touch /tmp/test2.txt
```

3. `crontab -r`：删除任务

- **功能：**
永久删除当前用户的所有计划任务。
- **示例：**
bash

```
crontab -r
```

4. `crontab -u username`：管理其他用户

- **功能：**
需要 `sudo` 权限，用于查看/编辑其他用户的任务。
- **示例：**
bash

```
[root@xnha ~]# crontab -u user777 -e
no crontab for user777 - using an empty one
crontab: installing new crontab
[root@xnha ~]# crontab -u user777 -l
0 3 * * * touch /tmp/user777
```

3.4 使用方法

场景1：每两个月的工作日的凌晨3点打包备份日志/var/log/secure

可以在/etc/crontab中直接编辑，也可以使用命令`crontab -e`进行编辑

```
# 编辑用户级crontab
crontab -e
# 添加任务（每天3:00执行）
0 3 * 1, 3, 5, 7, 9, 11 1-5 root tar -zcf /tmp/log.tar.gz /var/log/secure
```

```
#该命令执行后的效果有两个问题

1.tar打包命令最好使用相对路径
#修改
0 3 * 1, 3, 5, 7, 9, 11 1-5 root cd /var/log;tar -zcf /tmp/log.tar.gz secure
#使用';'分割两条命令

2.该命令会覆盖之前的压缩文件，最后只有11月的周五的日志
举个例子，接下来的这个小时，每分钟备份一次
[root@xnha ~]# date
2025年 03月 13日 星期四 11:34:32 EDT
[root@xnha ~]# crontab -e
* 11 13 3 * root cd /var/log;tar -zcf /tmp/log.tar.gz secure
[root@xnha ~]# cd /tmp
[root@xnha tmp]# ll |grep log.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:38 log.tar.gz
[root@xnha tmp]# ll |grep log.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:39 log.tar.gz

#修改
[root@xnha tmp]# touch logs`date +%H:%M`.txt
[root@xnha tmp]# ls |grep logs
logs11:41.txt
#同理,但是%在该配置文件中具有特殊含义，需要转义，此时错误的命令会发送邮件信息
* 11 13 3 * root cd /var/log;tar -zcf /tmp/log`date +\%H:\%M`.tar.gz secure
[root@xnha tmp]# ll | grep log
-rw-r--r--. 1 root root 2065 3月 3 21:50 anaconda.log
-rw-r--r--. 1 root root 2604 3月 3 21:50 dbus.log
-rw-r--r--. 1 root root 0 3月 3 21:50 ifcfg.log
-rw-r--r--. 1 root root 1678 3月 13 11:52 log11:52.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:53 log11:53.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:54 log11:54.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:55 log11:55.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:56 log11:56.tar.gz
-rw-r--r--. 1 root root 1678 3月 13 11:57 log11:57.tar.gz
```

四、at vs cron对比分析

特性	at	cron
适用场景	一次性/延迟任务	周期性重复任务
时间精度	秒级	分钟级
资源开销	无长期守护进程	需 <code>crond</code> 服务常驻
配置复杂度	简单时间描述	复杂时间表达式

五、总结与实战案例

5.1 关键总结

- `at`：适合一次性任务（如临时重启服务、延迟执行敏感操作）。
- `cron`：适合长期周期任务（如每日备份、定时监控）。

5.2 综合案例：自动化运维

需求：

- 每日凌晨2点备份网站数据到远程服务器。

```
0 2 * * * root ./linux.sh
```

- 每小时检查服务器磁盘使用率，超过90%时发邮件报警。

解决方案：github搜索脚本，按时运行！

1. 使用 `cron` 备份：

```
0 2 * * * rsync -avz /www/user123.com/ user@remote:/backup/
```

2. 使用 `at` 发送报警：

```
# 每小时检查磁盘
0 * * * * df -h | grep /dev/sda1 | awk '{print $5}' | while read -r usage;
do
    if [ "$usage" -gt 90 ]; then
        at now + 5 minutes -t Asia/Shanghai
        mail -s "Disk Alert!" admin@example.com <<< "Disk usage on /dev/sda1 is
$usage%"
    fi
done
```