

# GNU/Linux 计划任务



# GNU/Linux- 计划任务

进程是由程序执行后才产生的。程序的执行可以通过两种方法之一可以实现

1. 手动执行

2. 利用调度任务，依据一定的条件自动执行。

就像 windows 中的计划任务

如果打算让程序自动执行，您可以使用下面的命令来得已实现：

- 1.at

- 2.cron



# GNU/Linux- 计划任务

at 由 atd 服务负责并管理的。属于 at 的守护进程。可通过如下指令查看、启动、关闭 atd。

```
#systemctl stop atd
```

```
#systemctl start atd
```

```
#systemctl restart atd
```

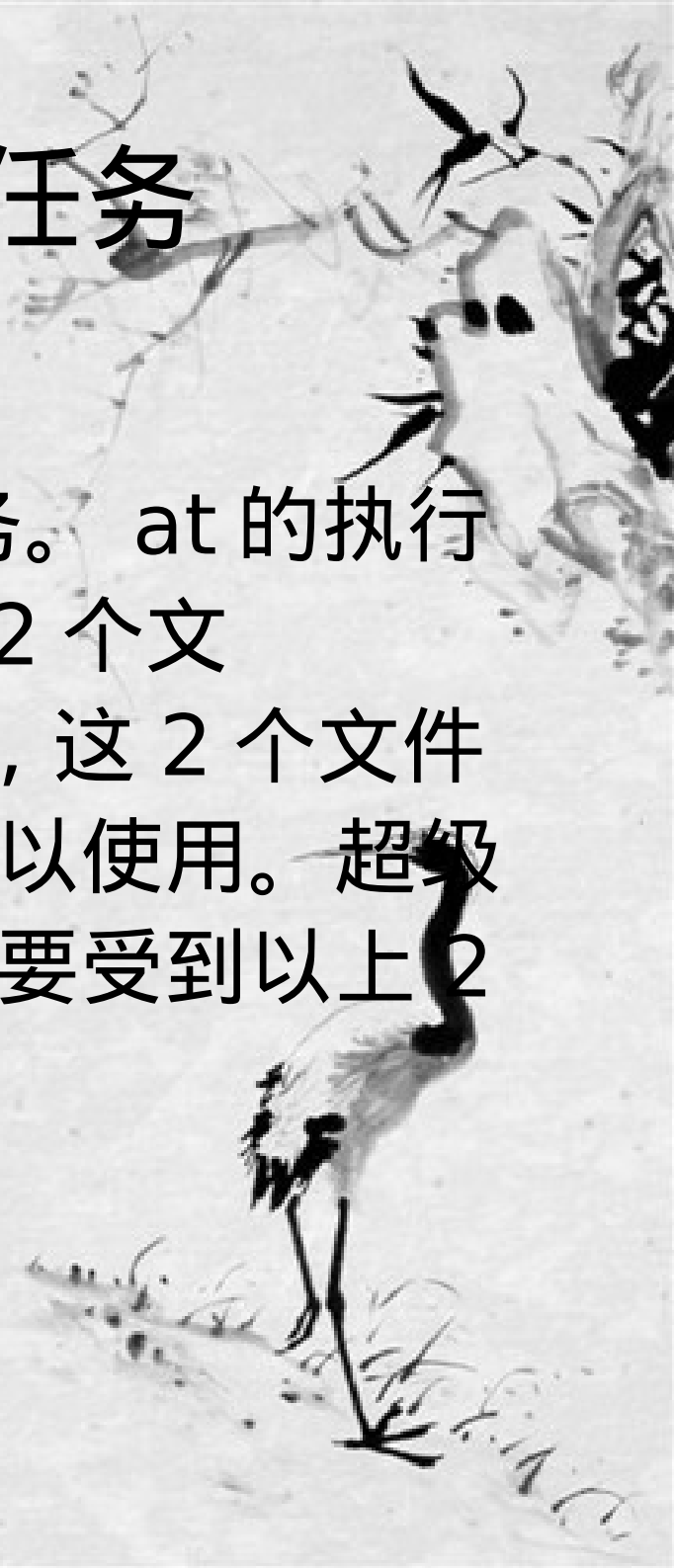
```
#systemctl status atd
```



# GNU/Linux- 计划任务

at

at ，这是一个一次性的计划任务。 at 的执行主要依靠系统服务 atd 。系统中有 2 个文件 /etc/at.allow 和 /etc/at.deny ，这 2 个文件规定了系统谁可以使用 at ，谁不可以使用。超级用户随时可以使用 at 。其他用户则要受到以上 2 个文件的限制。



# GNU/Linux- 计划任务

at

总结如下：

如果 2 个文件都存在，只检查 /etc/at.allow，只有在 /etc/at.allow 中规定的用户才可以使用。

如果只有 /etc/at.deny 存在，每个在该文件中不存在的用户都可以使用。

如果 2 个文件都不存在，只有超级用户可以使用。

# GNU/Linux- 计划任务

at 可接受的时间格式

日期

MM/DD/YY

MMDDYY

DD.MM.YY

时间

HH:MM

HH:MMam

HH:MMpm



# GNU/Linux- 计划任务

at 可接受的时间格式

特殊日期

midnight(0am)

noon(0pm)

teatime(4pm)

now( 现在 )

now + N(minutes, hours, days, weeks)



# GNU/Linux- 计划任务

命令 :at

语法 :at [ 选项 ] [time]

功能：依据预先定义好的时间，执行指定进程





# GNU/Linux- 计划任务

示例：

```
#at 23:00
```

```
at>reboot
```

```
at><EOT> ← 按 ^D 得到此符号，意味结束
```

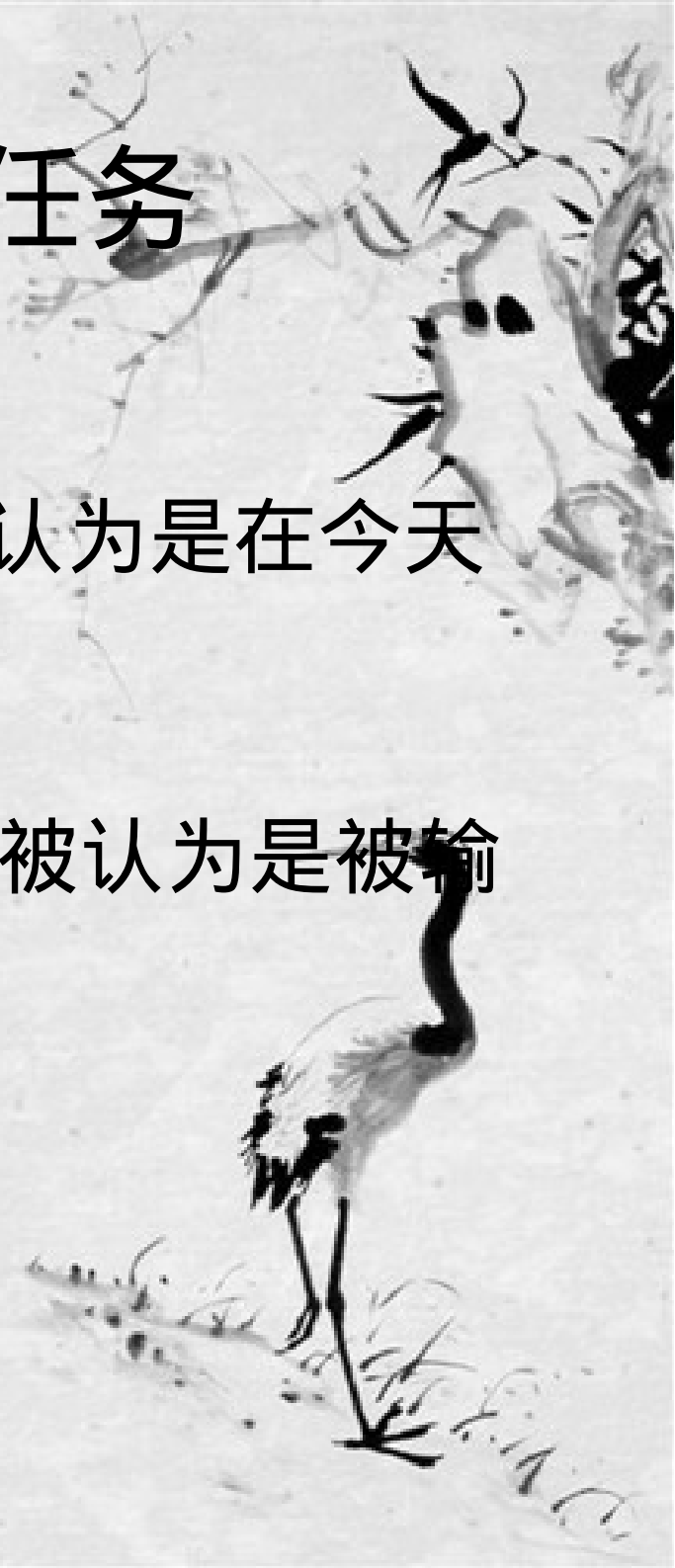
```
at
```



# GNU/Linux- 计划任务

at 时间输入规范：

1. 只输入时间，不输入日期，会被认为是在今天的某个时刻。
2. 如果只输入日期，不输入时间，被认为是被输入日期的现在这个时刻。
3. 支持输入一个完整的时间。



# GNU/Linux- 计划任务

at 时间时间格式范例：

#at 23:00

#at 12/31/13

#at 23:00 12/31/13

#at noon



# GNU/Linux- 计划任务

at 时间时间格式范例：

#at 0:00pm 12/31/13

#at teatime + 30 minutes

#at 4:00pm + 4 days

#at now + 4 weeks



# GNU/Linux- 计划任务

## at 队列管理

at 在编辑后，将进入队列中。由 atd 服务扫描任务并等待任务的执行

查看 at 队列可使用下面命令  
命令 :atq

功能：查看 at 队列

结构语法 :atq [ 选项 ]



# GNU/Linux- 计划任务

## at 队列管理

at 在编辑后，如打算查看某个任务队列的信息及准备执行的命令，可按如下命令执行：

```
#at -c at 队列 ID
```



# GNU/Linux- 计划任务

## at 队列管理

at 编辑后，队列等待执行。如取消预定的 at 任务，可以使用 atrm 完成取消动作。

命令 :atrm

功能：删除 at 队列中的指定任务

结构语法 :atrm [ 选项 ] [ 队列 ID]



# GNU/Linux- 计划任务

cron

cron 与 at 最大区别，在于 cron 属于依据指定周期执行预定任务。cron 的守护进程为 crond。

```
#systemctl stop crond
```

```
#systemctl start crond
```

```
#systemctl restart crond
```

```
#systemctl status crond
```





# GNU/Linux- 计划任务

cron 使用

命令 :crontab

功能 : 实现周期性计划任务的执行

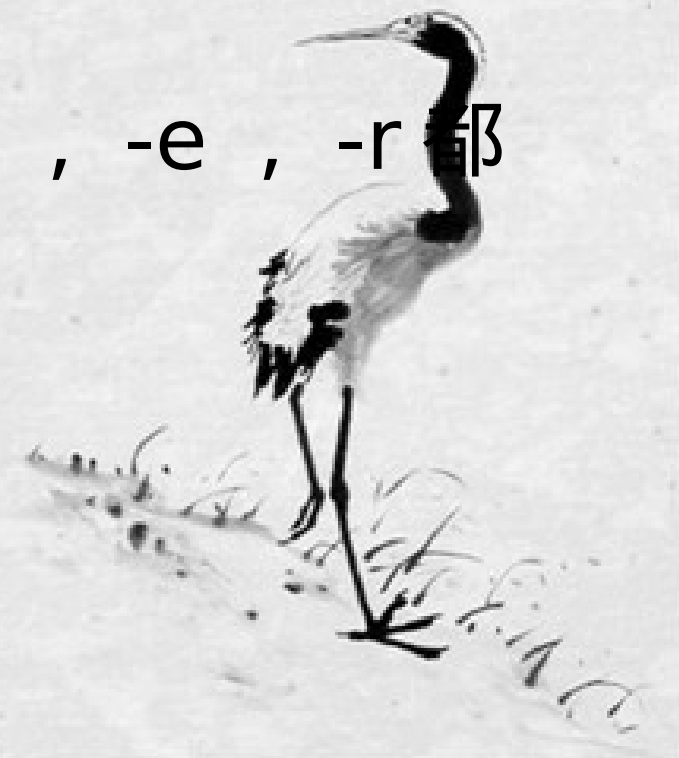
语法格式 :crontab [ 选项 ]



# GNU/Linux- 计划任务

选项：

- l 查看 cron 计划任务列表。
- e 以默认文本编辑器，编辑 cron 计划任务列表。
- r 删除 cron 计划任务列表。
- u 指定用户。如果不指定用户，-l，-e，-r 都认为是对当前用户操作。



# GNU/Linux- 计划任务

cron 编辑的格式：

1)shell 及 mail 格式定义

`SHELL=/bin/bash`

SHELL 指定以哪个 shell 来处理计划任务的命令。

`MAILTO=root`

MAILTO 指定运行完计划任务或者是运行是遇到障碍，都会发邮件给某个用户，给以提示。



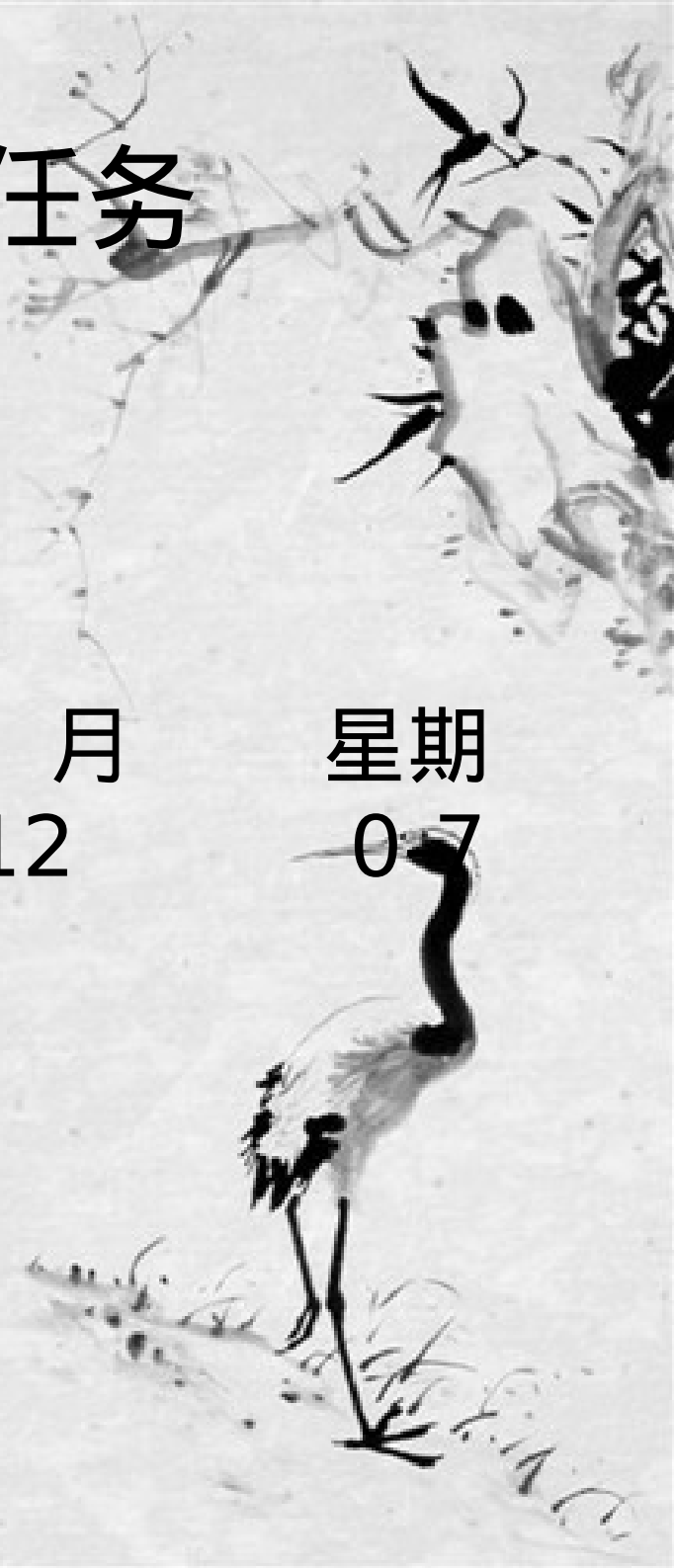
# GNU/Linux- 计划任务

cron 编辑的格式：

## 2) 时间格式定义

| 分钟   | 小时   | 每月中的第几天 | 月    | 星期  |
|------|------|---------|------|-----|
| 0-59 | 0-23 | 1-31    | 1-12 | 0-7 |

//\* 星期中 0,7 同样表示周日



# GNU/Linux- 计划任务

cron 编辑的格式：

2) 时间格式定义

“\*” 在第一列，能够代表每一分钟。

“\*” 在第二列，能够代表每一小时。

“\*” 在第三列，能够代表一个月中的每一天。

“\*” 在第四列，能够代表一年中的每个月。

“\*” 在第五列，能够代表一个星期中的每一天。



# GNU/Linux- 计划任务

cron 编辑的格式：

3) 特殊时间格式定义

连续的情况，以小时举例

1,2,3,4,5,8,9,10 可以这样写 1-5,8-10。

周期间隔

如每个 5 分钟可在第一列写成：

`*/5`



# GNU/Linux- 计划任务

cron 编辑的格式：

4)cron 格式

\* \* \* \* \* 所需要执行的命令及路径

如

0 0 1 1 \* echo "Happy New Yera"



# GNU/Linux- 计划任务

cron 编辑的格式：

示例

```
0 */2 13 * 5 /usr/bin/wall be careful,dark  
friday!
```

代表黑色星期五，每隔 2 小时，向用户发出提示。

```
0 20 * * 1-5 /sbin/shutdown -h now
```

从周 1 到周 5，每天晚上 8 点，自动关机。



# GNU/Linux- 计划任务

## cron 编写任务

1. 以当前账户创建 cron 计划任务

```
#crontab -e
```

2. 以指定用户创建其 cron 计划任务 ( 仅 root)

```
#crontab -e -u snow
```



# GNU/Linux- 计划任务

## cron 编写任务

### 3. 以指定文件作为计划任务的要求

```
#vim newcron
```

```
*/5 * * * * wall hello
```

```
#crontab newcron
```

或

```
#crontab -u snow newcron
```



# GNU/Linux- 计划任务

## cron 编写任务

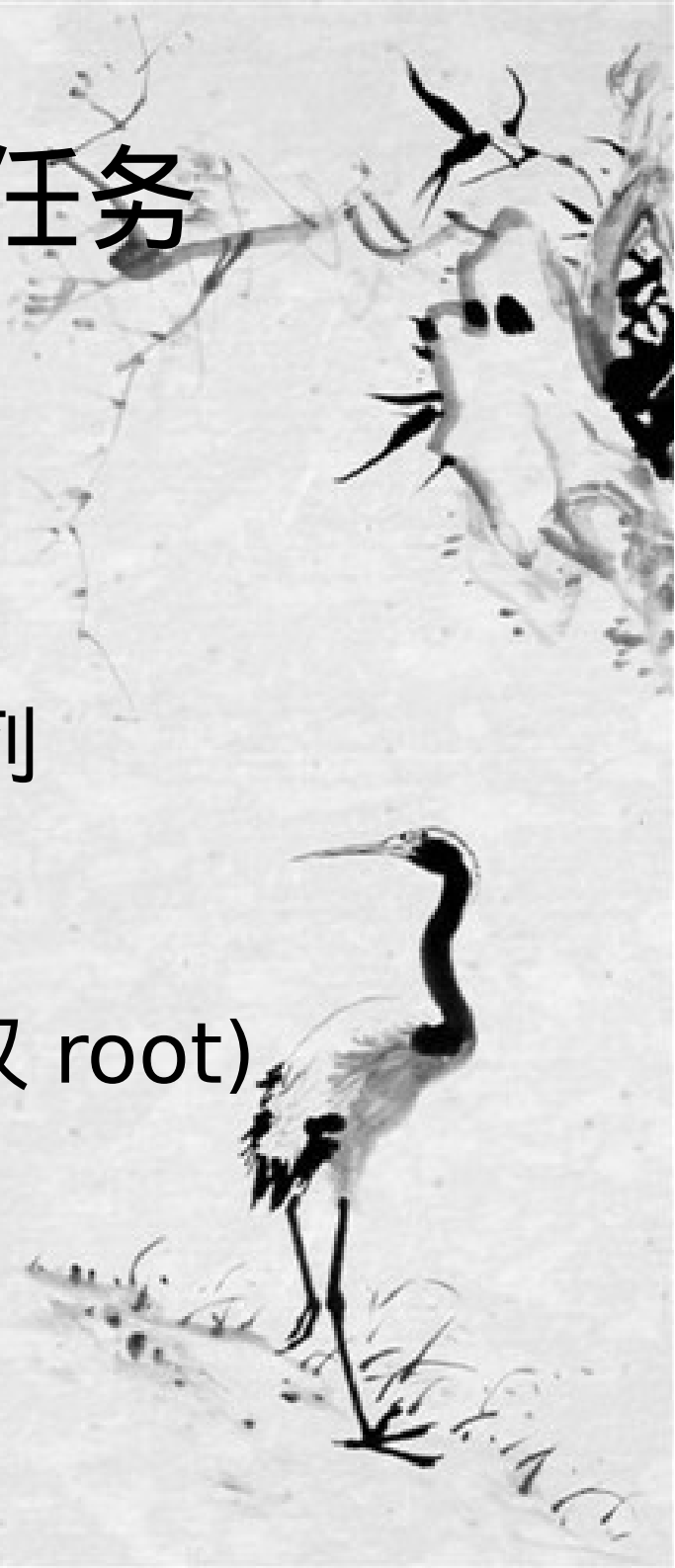
### 4. 查看 cron 任务队列

1) 查看当前使用账户的 cron 队列

```
#crontab -l
```

2) 查看其他账户的 cron 队列 ( 仅 root)

```
#crontab -l -u snow
```



# GNU/Linux- 计划任务

## cron 编写任务

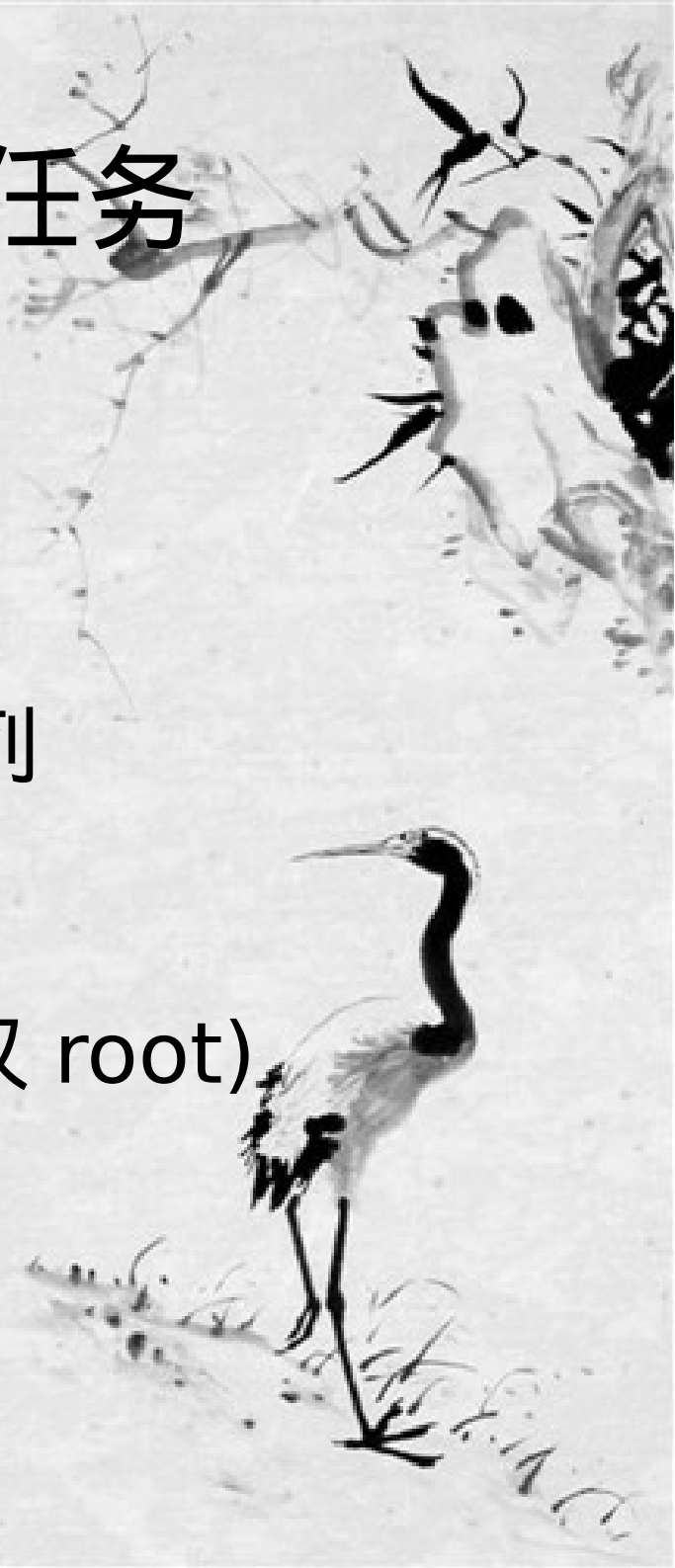
### 5. 清除 cron 任务队列

1) 清除当前使用账户的 cron 队列

```
#crontab -r
```

2) 清除其他账户的 cron 队列 ( 仅 root)

```
#crontab -r -u snow
```



# GNU/Linux- 计划任务

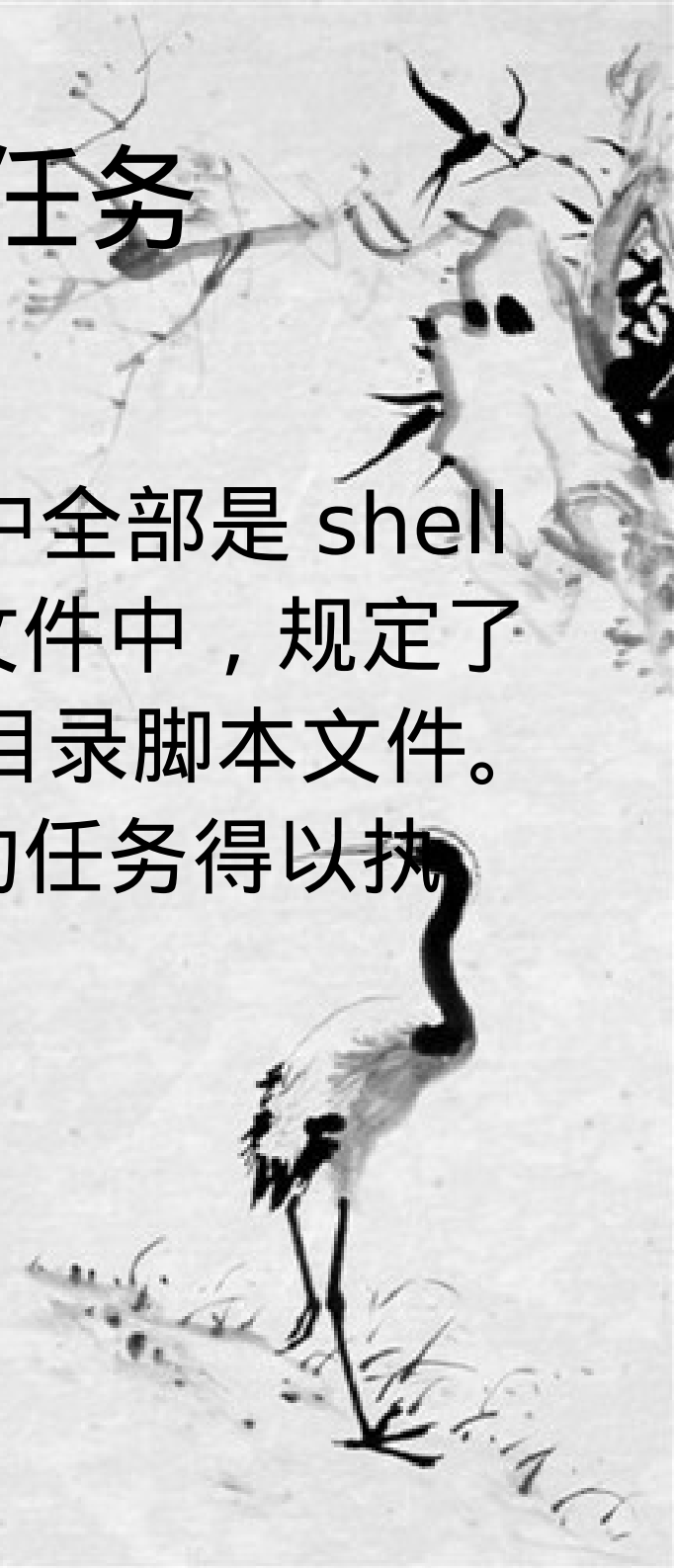
## cron 编写任务

|                   |                 |
|-------------------|-----------------|
| /etc/crontab      | 系统计划任务表及说明。     |
| /etc/cron.d       | 特定的系统计划任务表      |
| /etc/cron.hourly  | 系统每小时要运行的计划任务。  |
| /etc/cron.daily   | 系统每天要运行的计划任务。   |
| /etc/cron.weekly  | 系统每个星期要运行的计划任务。 |
| /etc/cron.monthly | 系统每个月要运行的计划任务。  |

# GNU/Linux- 计划任务

## cron

虽然 `/etc/cron.hourly` 等目录中全部是 shell 脚本，但在 `/etc/cron.d/0hourly` 文件中，规定了使用 `run-parts` 命令来执行指定的目录脚本文件。这就使得这些上述 shell 脚本目录的任务得以执行。



# GNU/Linux- 计划任务

## anacron

cron 的计划任务执行必须保持计算机处于开启状态。如果关机或在指定的时间没有执行，将只能等待下一个周期。

而 anacron 不同，如果在指定的时间没有成功执行，则会在一段时间之内与所指定的重试时间再次执行



# GNU/Linux- 计划任务

anacron 配置文件

```
#vim /etc/anacrontab
```

格式

时间间隔（默认为天，@mohtnly 为 1 个月内执行）

等待间隔（默认为分钟）

任务标识

命令





# GNU/Linux- 计划任务

anacron 配置文件

说明：

时间间隔：执行任务的时间间隔

等待间隔：时间间隔到期后，任务没有执行，则使用此处设置时间再次尝试执行

任务标识：任务说明

命令：实际所执行的任务

