GNU/Linux
分析和存储日志

GNU/Linux
分析和存储日志

进程和 kernel 在发生一些事件时需要能够将记录，这些记录将记录在指定或自定义的文件中。通过这些文件可以查看相关的信息，以控制和对问题的排故。

默认情况下 ,Linux 将日志的信息存储在 /var/log 目录下。

GNU/Linux
分析和存储日志

RHEL7 系统使用基于标准的日志协议完成日志记录。

大多数的程序都使用这个标准日志程序来记录自己所发生的信息。

RHEL7 中由两个服务来完成日志消息记录的。

1. systemd-journald
2. rsyslogd

GNU/Linux
分析和存储日志

systemd-journald 是一个改进的日志系统。可以良好的收集各种信息。

在 Linux 系统启动的初期 ,systemd-journald 即对各项守护进程中的启动、运行及相关信息、标准输出及各项错误等各项事件进行记录组织并按照相关要求标准、结构进行记录。默认情况下，systemd-journald 持续的记录。

GNU/Linux
分析和存储日志

systemd-journald 持续的记录。将不会错过其他程序产生信息而漏记的情况。

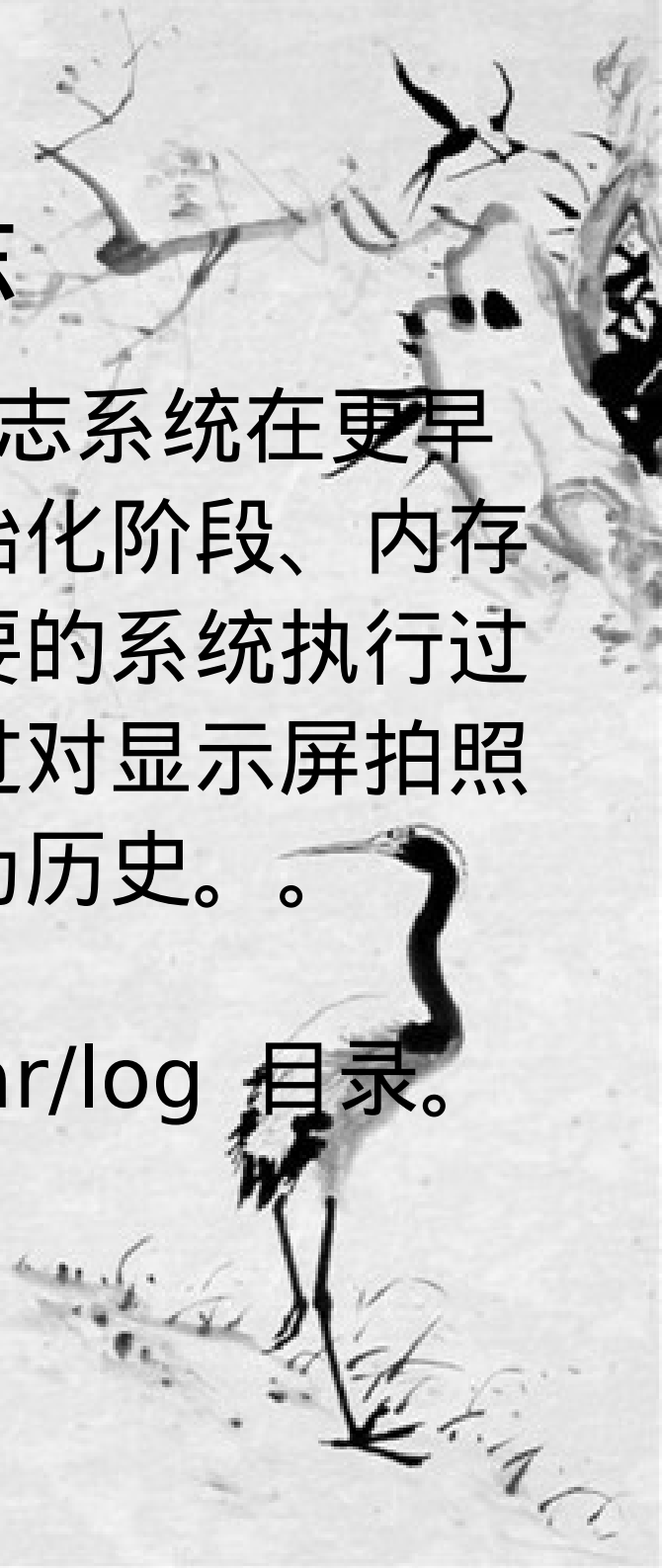
systemd-journald 会将所有的信息记录在一个集中的数据库中。

syslog 消息也通过 systemd-journald 传送到 rsyslog 上进行进一步处理

GNU/Linux
分析和存储日志

这就使得 systemd -journald 日志系统在更早的时候启动起来，可以记录内核初始化阶段、内存初始化阶段、前期启动步骤以及主要的系统执行过程的日志。所以，以前那种需要通过对显示屏拍照或者暂停系统来调试程序的日子成为历史。。

systemd 的日志文件都被放在 /var/log 目录。



GNU/Linux
分析和存储日志

rsyslog:

　　可以对服务进行排序，而后根据类型或设备、优先级等情况将信息写入至 /var/log 目录中

GNU/Linux
分析和存储日志

rsyslog:

日志信息是通过 logrotate 的相关配置及其命令，来实现周期性的检测信息并将这些信息写入至 /var/log 下。

每次轮询后，都将会将过去的 message 日志将会改为带有日期的 message 文件 . 如 messgae-20141030, 而新的记录则仍然记录在 message 文件中。

GNU/Linux
分析和存储日志

系统日志概述

1. /var/log/message

多数服务或进程、系统信息都被记录到此文件中，但不包含身份认证、电子邮件处理、计划任务和 debug 动作

message 格式为：

产生日志的时间 哪个主机所发送的日志

GNU/Linux
分析和存储日志

系统日志概述

1. /var/log/message

message 格式为：

(1) 产生日志的时间

(2) 哪个主机所发送的日志消息

(3) 哪个服务或进程所发送的日志消息

(4) 消息内容

GNU/Linux
分析和存储日志

系统日志概述

1. /var/log/message

当需要动态查看或实时监控 message 文件或其他文件时，都可以使用

#tail -f log_file

GNU/Linux
分析和存储日志

系统日志概述

2. /var/log/secure
   此日志记录与安全、认证相关的信息和错误

3. /var/log/maillog
   此文件记录了与邮件、邮件服务相关的信息

GNU/Linux
分析和存储日志

系统日志概述

4. /var/log/cron

   此文件记录了与计划任务相关的信息

5. /var/log/boot.log

   此文件记录了有关系统启动的消息记录

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 的配置文件位于 /etc/rsyslog.conf

其文件格式为

设施 . 优先级　　日志记录文件路径及文件名

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 设施类型 (facility)

设施名	代码	说明
kern	0	内核日志消息
user	1	随机的用户日志消息
mail	2	邮件系统日志消息
daemon	3	系统守护进程日志消息
auth	4	安全管理日志消息

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 设施类型 (facility)

设施名	代码	说明
syslog	5	syslogd 本身的日志消息
lpr	6	打印机日志消息
news	7	新闻服务日志消息
uucp	8	uucp(unix2unixcopy) 日志消息
cron	9	crond 日志消息

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 设施类型 (facility)

设施名	代码	说明
authpriv	10	授权相关的日志信息
ftp	11	ftp 守护进程的日志信息
	12-15	保留，由系统使用
local0~local 7	16-23	保留，本地使用（可自定义）

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 优先级 (priority)

优先级	代码	说明
emerg	0	系统不可用（最高紧急状态）
alert	1	必须马上采取行动（紧急状态）
cirt	2	关键事件（重要信息）
err	3	错误事件（错误）

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 优先级 (priority)

优先级	代码	说明
warning	4	警告时间（临界状态）
notice	5	普通但重要事件（出现不寻常的事件）
info	6	有用的信息（一般性信息）
debug	7	调试信息

GNU/Linux
分析和存储日志

Rsyslog

rsyslog 的优先级由高 (0) 到低 (7) ，来判定信息重要与否 .

一般低级别都包含高级别优先级

作为优先级中还有 1 个为” none”, 即不产生任何信息。

GNU/Linux
分析和存储日志

Rsyslog

Rsyslog 可以通过下面特殊字符对级别进行扩展操作：

=: 指定至使用这个级别，而不包含更高级别

!: 忽略所有级别

-: 表示使用异步的方式记录

GNU/Linux
分析和存储日志

使用 logger 发送日志消息

命令 :logger

功能： 发送一条日志消息

语法格式 :logger［ 选项 ］［ 消息 ］

GNU/Linux
分析和存储日志

使用 logger 发送日志消息

1. 发送一个 local3.info 的日志消息

   #logger -p local3.info “hello!”

   //*-p 指定优先级

2. 发送一个以 kern 设备优先级为 err 级别的消息

   #logger -t kern -p err “hello,wahahah.”

GNU/Linux
分析和存储日志

Rsyslog

1. 查看当前系统中 rsyslog 状态

#systemctl status rsyslog
#systemctl is-enabled rsyslog
#systemctl is-active rsyslog

2. 通过 ps 查看 rsyslog 进程

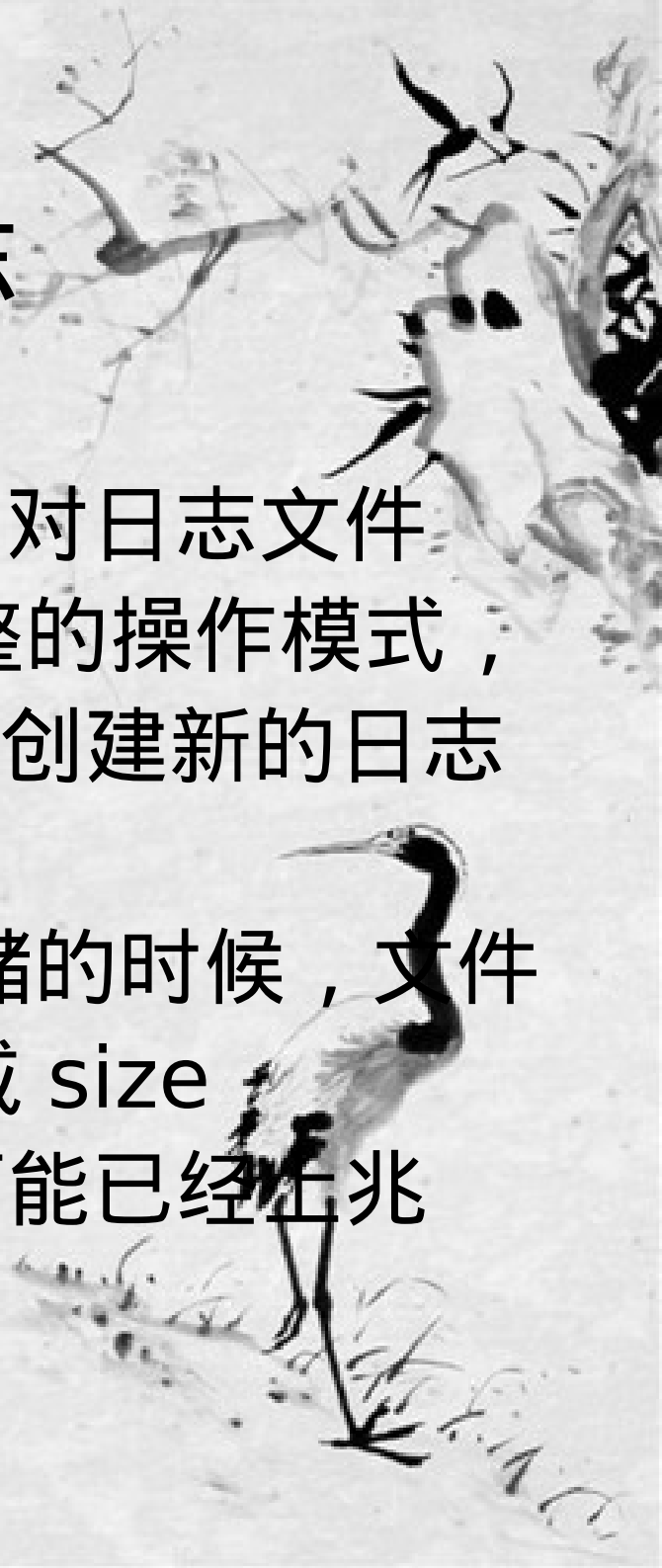
#ps -ef | grep rsyslogd | grep -v grep

GNU/Linux
分析和存储日志

logrotate

logrotate 是个强大的系统软件，它对日志文件（只要文件名固定的）有着一套完整的操作模式，譬如：转储（删除或到处旧文件，并创建新的日志文件）、邮件和压缩等等；

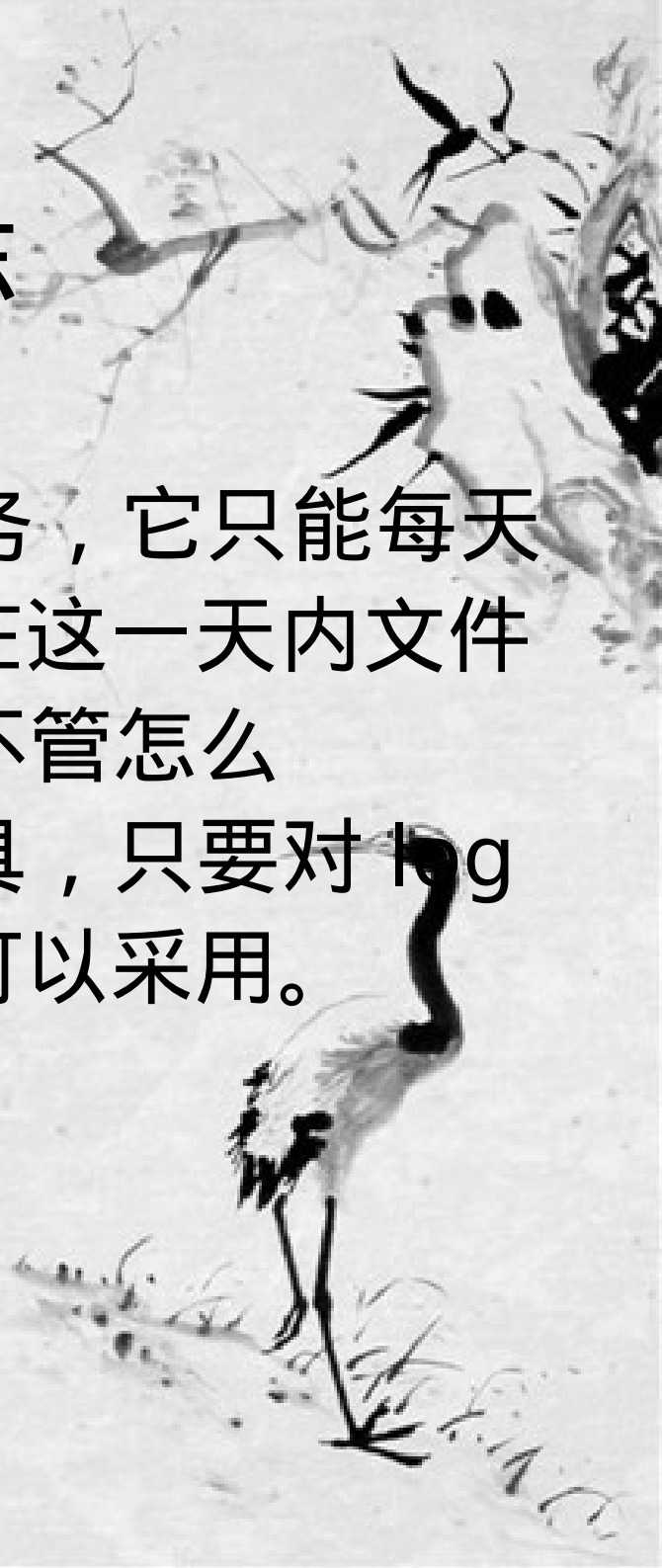
　但是它也有那么点瑕疵：就是转储的时候，文件大小经常不是预期的，譬如，设置成 size 1024 ，那么转储的时候文件大小可能已经上兆了。



GNU/Linux
分析和存储日志

logrotate

原因是 ,logrotate 是个 cron 任务，它只能每天（最小轮询单位）轮询一次，至于在这一天内文件大小超过 1024 ，它也无能为力。不管怎么说， logrotate 是个非常实用的工具，只要对 log 临时需要（测试）或要求不高，都可以采用。



GNU/Linux
分析和存储日志

命令 :logrotate

功能 :

为了简化对产生大量日志文件的各种系统管理， logrotate 从而被设计出来。它可以自动转储、压缩、删除和邮件这些日志。每个日志文件可以在每天，每星期，每月或太大了的时候被处理。

通常， logrotate 是一个每天执行一次的 cron 任务。除非被操作的日志文件是以大小处理的，并且在一天内 logrotate 被执行多次，或是使用了选项 -f 或 -force ，每天 logrotate 不会多次修改一个文件。

GNU/Linux
分析和存储日志

命令 :logrotate

功能 :

命令行可以包含任意数量的配置文件。后面的配置文件可能会覆盖前面的配置文件给出的选项，所以 logrotate 列出的配置文件顺序是很重要的。通常，如果需要包含其它配置文件信息，那么应该用一个单一的配置文件。根据下面更多的信息，通过使用 include 导向来完成这个。假如命令行上给出的是个目录，那么在这个目录下的文件都被看作是一个配置文件。

GNU/Linux
分析和存储日志

命令 :logrotate

功能： 如果命令行不包含任意选项，那么 logrotate 会打印出它的版本和版权信息，以及一段短小的用法摘要。

说明 : 循环处理、压缩、邮递系统日志

语法格式 :logrotate [选项] config_file

GNU/Linux
分析和存储日志

选项：

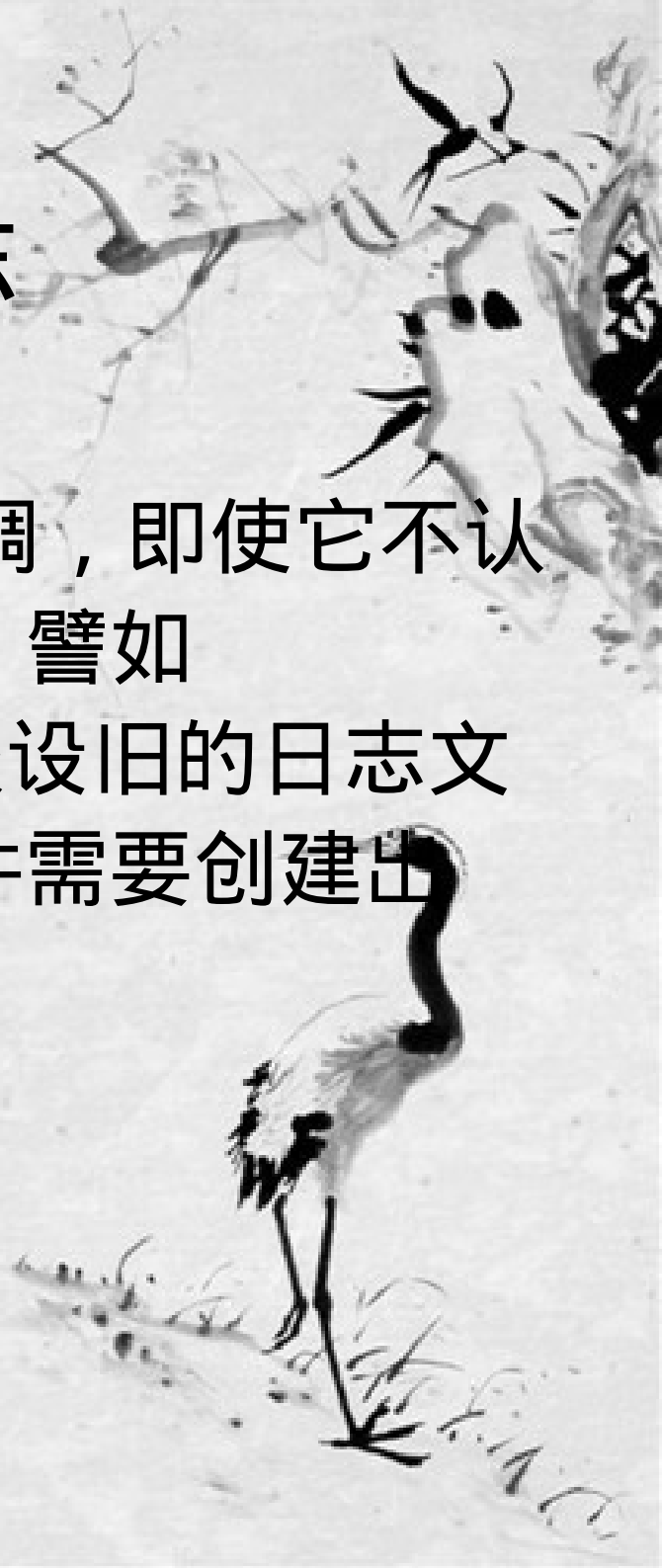
-v：打开详细模式。

-d: 打开调试模式，并且打开冗长模式 -v 。在调试模式下，日志文件和 log-rotate 状态文件不会有变化。

GNU/Linux
分析和存储日志

选项：

-f, --force: 告诉 logrotate 强制轮调，即使它不认为这是必要的。有时这是很有用的，譬如 logrotate 增加了新的入口；又如假设旧的日志文件被手动删除了，这样新的日志文件需要创建出来，保证正常写日志。



GNU/Linux
分析和存储日志

选项：

-m, --mail <command>: 告诉 logrotate 当邮件日志时调用 command。这个 command 需要包含 2 个参数：

1）消息的主题

2）接收者。这个 command 必须从标准输入上读出消息，然后发邮件给接收者。默认发邮件命令是 /bin/mail -s。

GNU/Linux
分析和存储日志

选项：

-s, --state <statefile>: 告诉 logrotate 使用备用的状态文件。当另一个用户正在用 logrotate 管理大量的日志文件时，这是非常有用的。默认的状态文件是 /var/lib/logrotate/status 。

--usage: 打印简要的用法信息。

GNU/Linux
分析和存储日志

logrotate 配置文件：

```
#vim  /etc/logrotate.conf
```

主要参数：

compress: 通过 gzip 压缩转储以后的日志

nocompress: 不需要压缩时，用这个参数

copytruncate: 用于还在打开中的日志文件，把当前日志备份并截断

nocopytruncate: 备份日志文件但是不截断

create mode owner group: 转储文件，使用指定的文件模式创建新的日志文件

GNU/Linux
分析和存储日志

主要参数
nocreate: 不建立新的日志文件
delaycompress 和 compress 一起使用：转储的日志文件到下一次转储时才压缩
nodelaycompress 大于 delaycompress 选项：转储同时压缩。
errors address: 专储时的错误信息发送到指定的 Email 地址
Ifempty: 即使是空文件也转储，这个是 logrotate 的缺省选项。

GNU/Linux
分析和存储日志

主要参数
notifempty: 如果是空文件的话，不转储
mail address: 把转储的日志文件发送到指定的 E-mail 地址
nomail: 转储时不发送日志文件
olddir directory: 转储后的日志文件放入指定的目录，必须和当前日志文件在同一个文件系统
noolddir 转储后的日志文件和当前日志文件放在同一个目录下

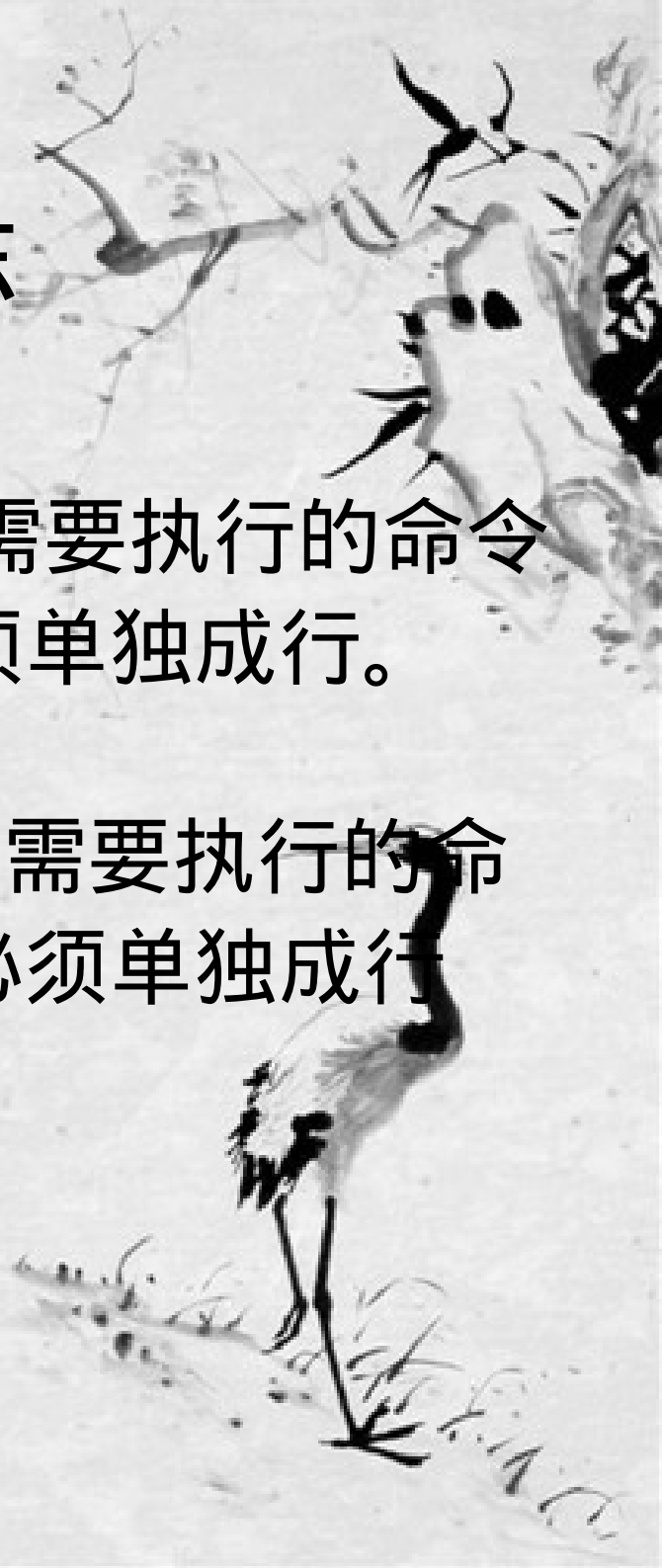
GNU/Linux
分析和存储日志

主要参数

prerotate/endscript: 在转储以前需要执行的命令可以放入这个对，这两个关键字必须单独成行。

postrotate/endscript: 在转储以后需要执行的命令可以放入这个对，这两个关键字必须单独成行

daily: 指定转储周期为每天
weekly: 指定转储周期为每周
monthly: 指定转储周期为每月



GNU/Linux
分析和存储日志

主要参数

rotate count: 指定日志文件删除之前转储的次数，0 指没有备份，5 指保留 5 个备份

tabootext [+] list: 让 logrotate 不转储指定扩展名的文件，缺省的扩展名是：.rpm-orig, .rpmsave, v, 和 ~

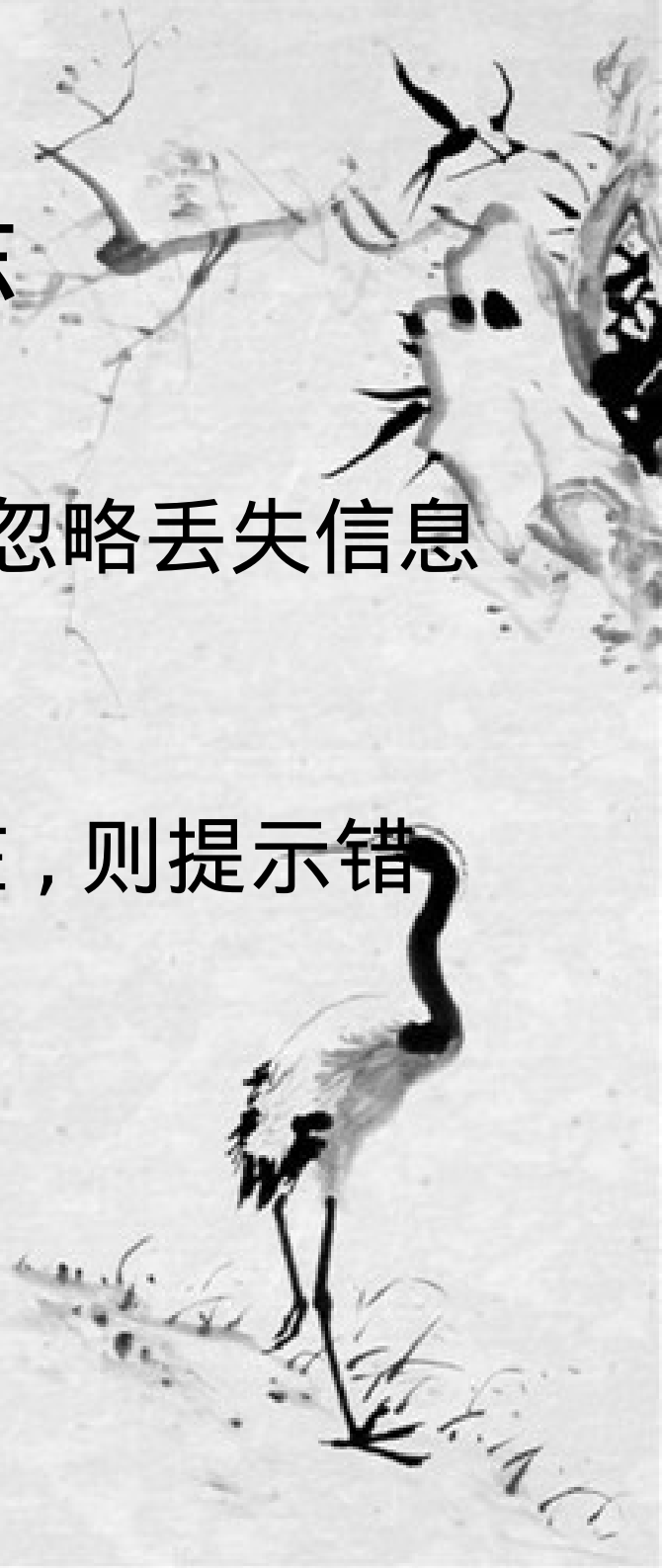
size size: 当日志文件到达指定的大小时才转储，Size 可以指定 bytes (缺省) 以及 KB (sizek) 或者 MB (sizem).

GNU/Linux
分析和存储日志

主要参数

missingok: 如果日志文件丢失，则忽略丢失信息的错误提示，继续工作

nomissingok: 如果日志文件不存在，则提示错误。（此属于默认选项）



GNU/Linux
分析和存储日志

缺省配置

weekly: 对所指定的日志每周转储 1 次

rotate 4: 指定转储文件保留 4 份

create: 自动建立新（空）的日志，并转储旧日志，权限，属组，属组

dateext: 以日志作为转储文件的后缀

#compress: 不压缩转储文件（默认为注释）

GNU/Linux
分析和存储日志

缺省配置

Include /etc/logrotate.d:logrotate 运行时同时运行在 /etc/logrotate.d/* 的子服务的日志配置

配置说明 :

/var/log/wtmp {

monthly ← 每月转储

create 0664 root utmp ← 创建 wtmp 文件 , 文件权限 0644, 属主 root, 属组 :utmp

minsize:1M ← 日志大于 1M 才将被转储

GNU/Linux
分析和存储日志

配置说明 :

举例 :

```
/var/log/wtmp {
    monthly ← 每月转储
    create 0664 root utmp ← 创建 wtmp 文件 ,
文件权限 0644, 属主 root, 属组 :utmp
    minsize:1M ← 日志大于 1M 才将被转储
    rotate 1 ← 转储文件保留 1 份
}
```

GNU/Linux
分析和存储日志

示例 1:

1. 编辑一个 test 的配置

```
#vim /etc/logrotate.d/test
/var/log/test.log {
    missingok
    rotate 5
    size 1k
    create 0640 root root
}
```

GNU/Linux
分析和存储日志

示例 1:

2. 创建 /var/log/test.log 文件，大小为 1000K

```
#dd if=/dev/zero of=/var/log/test.log
bs=1024 count=1000
```

3. 执行 logrotate 转储

```
#logrotate -v /etc/logrotate.conf
```

4. 查看转储

```
#ls -l /var/log/test.log*
```

GNU/Linux
分析和存储日志

journalctl 的使用

命令 :journalctl

功能 :systemd-journald 的日志分析工具

语法格式 :journalctl [选项] [标记]

GNU/Linux
分析和存储日志

1. 查看当前系统日志
   #journalctl

2. 显示最新的 5 条日志记录
   #journalctl -n 5

3. 仅列示出 err 的错误信息
   #journalctl -p err

GNU/Linux
分析和存储日志

4. 实时查看日志信息
   #journalctl -f

5. 查看今天的日志
   #journalctl --since today

6. 查看昨天的日志
   #journalctl --yesterday

GNU/Linux
分析和存储日志

7. 查看 2014-09-09 至 2014-09-15 的日志信息

#journalctl --since “2014-09-09” --until “2014-09-15”

8. 查看 2014-09-09 12:50 至 2014-09-15 12:00 的日志信息

#journalctl --since “2014-09-09 12:50:00” --until “2014-09-15 12:00:00”

注：计算机必须 uptime, 一旦 reboot, 将只能看到启动之后的日志。

GNU/Linux
分析和存储日志

9. 通过 journalctl 获得更多的各项服务 / 进程的信息

```
#journalctl -o verbose
```

10. 显示 10 个服务 / 进程的详细信息

```
#journalctl -o verbose -n
```

11. 显示 1 个服务 / 进程的详细信息

```
#journalctl -o verbose -n 1
```

GNU/Linux
分析和存储日志

12. 显示指定的单元类型的相关日志信息

#journalctl _SYSTEMD_UNIT=sshd.service

13. 显示指定的单元类型及其 PID 的相关日志信息

#journalctl _SYSTEMD_UNIT=sshd.service _PID=862

14. 如果想查看更多的指定单元的信息及查询方法可参阅帮助

#man systemd.journal-fields

GNU/Linux
分析和存储日志

12. 显示指定的单元类型的相关日志信息

#journalctl _SYSTEMD_UNIT=sshd.service

13. 显示指定的单元类型及其 PID 的相关日志信息

#journalctl _SYSTEMD_UNIT=sshd.service _PID=862

15. 显示指定的 UID 所运行的服务 / 进程

#journalctl _UID=0

GNU/Linux
分析和存储日志

16. 显示指定的 PID 信息

```
#journalctl _PID=1
```

17. 按当日指定时间查看

```
#journalctl --since 9:00 --until 18:00
```

18. 依照某个时间查看指定的单元信息

```
#journalctl --since 9:00
_SYSTEMD_UNIT=sshd.service
```

GNU/Linux
分析和存储日志

19. 查看从启动时开始的所有信息
    #journalctl -b

20. 显示上一次系统启动前产生的日志
    #journalctl -b -1

21. 显示由 kernel 产生的日志信息
    #journalctl -k

GNU/Linux
分析和存储日志

22. 如果想查看更多的指定单元的信息及查询方法可参阅帮助

```
#man  systemd.journal-fields
```

GNU/Linux
分析和存储日志

利用 systemd 分析系统启动过程

systemd 可以让你能更有效地分析和优化你的系统启动过程：

1. 显示本次启动系统过程中用户、 initrd 和 kernel 所花费的时间

#systemd-analyze

2. 显示每个启动项所花费的时间明细

#systemd-analyze blame

GNU/Linux
分析和存储日志

3. 按时间顺序你打印 UNIT 树
   #systemd-analyze critical-chain

4. 产生开机启动过程的时间图标
   #systemd-analyze plot > bootplot.svg

GNU/Linux
分析和存储日志

5. 为开机启动过程生成向量图

#systemd-analyze dot | dot -Tsvg > systemd.svg

//* 需要 graphviz 软件包

//* 颜色标识 :

黑色 (black): 需要启动相关关联

深蓝 (dark blue): 必须

深灰 (black grey): 需求

红色 (Red): 冲突

绿色 (green):after(之后)

GNU/Linux
分析和存储日志

6. 将 systemd journal 写入至硬盘

1) 建立存放路径并设置相关权限

#mkdir -v /var/log/journal

#chown root.systemd-journal /var/log/journal

#chmod 2755 /var/log/journal

2) 重启你的系统或执行

#killall -USR1 systemd-journald

//*USR1 用户自定义信号，本实验用来产生日志消息

GNU/Linux
分析和存储日志

6. 将 systemd journal 写入至硬盘

3) 确认目录下内容

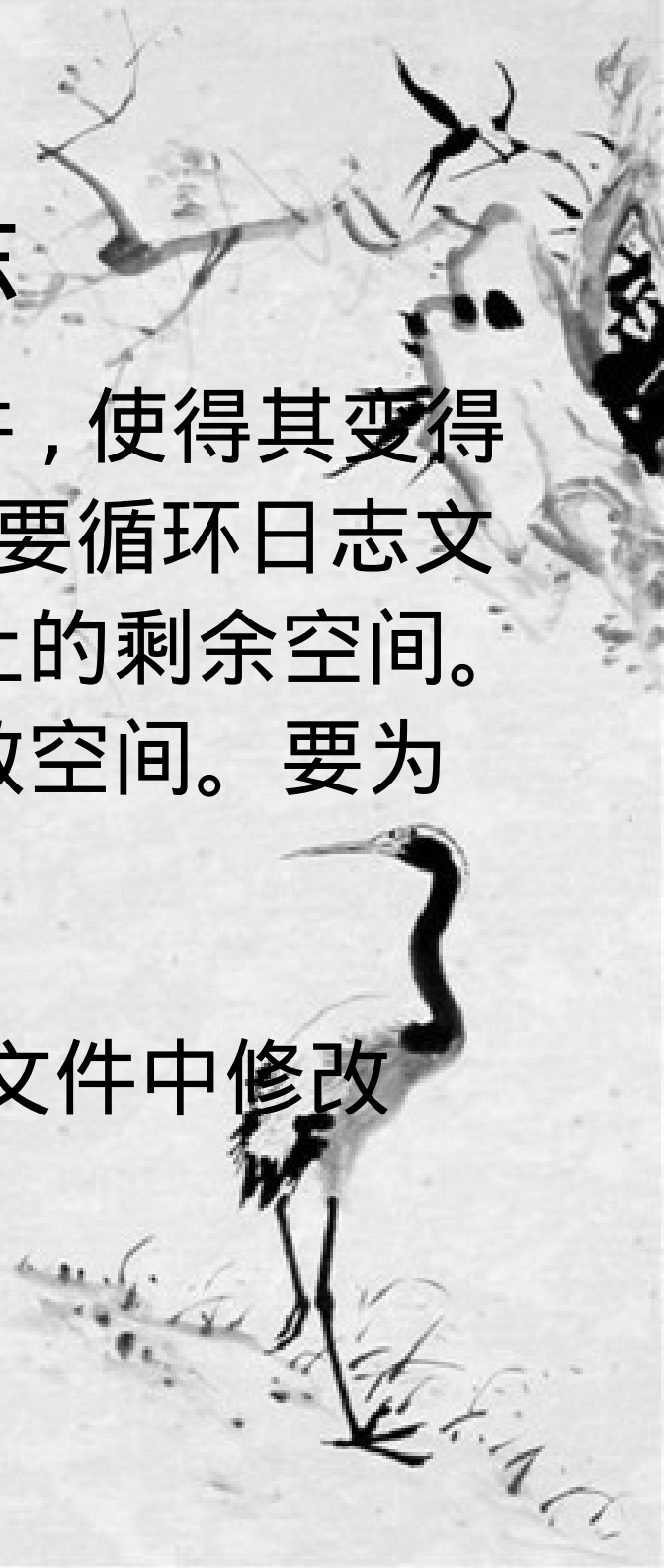
```
#cd  /var/log/journal
#cd $ID ← 进入你机器的编号
#ls
system.journal
```

GNU/Linux
分析和存储日志

logrotate 的关闭与归档日志文件，使得其变得日益巨大。在 journald 上，没有必要循环日志文件。它构建的目的在于监控存储卷上的剩余空间。如果卷快满了，就删除旧有记录释放空间。要为 journald 日志设置一个最大尺寸。

在 /etc/systemd/journal.conf 文件中修改

SystemMaxUse= 参数即可
//* 其单位为 (K,M,G,T,P,E)



GNU/Linux
分析和存储日志

Journald 没有将来自其他服务器或设备的日志文件进行替代的选项。

同时，也没有指定哪台日志服务器的日志事件可以转发的选项。

如果要 journald 在其他地方存放日志信息，最佳做法是将信息转发给 rsyslog 服务，让 rsyslog 完成日志的处理集中。

GNU/Linux
分析和存储日志

查看 systemd.journal 字段信息

#man systemd.journal-f