

敏行服务器系统维护手册

发布日期：2014 年 9 月

版本：V3.0

目 录

1 敏行服务器系统简介	2
1.1. 运行环境	2
1.2. 系统服务	2
2 系统维护	2
2.1. 常用命令	3
2.2. 启停服务	6
2.3. 备份数据	7
3 敏行后台维护系统	7
3.1. 系统功能	7
3.2. 详细说明	7

1 敏行服务器系统简介

敏行服务器系统是由一系列服务组成的高效可扩展的应用系统，运行于主流的 Linux 平台之上。

1.1. 运行环境

已证实敏行服务器系统可运行在以下平台之上：

- Debian 6.0.9
- Ubuntu 12.0.4
- Redhat Enterprise 6

为达到最佳运行效果，服务器至少需要 16GB 内存以及 30GB 硬盘空间。

1.2. 系统服务

敏行服务器系统包含以下主要服务程序：

- | | |
|-------------|------------|
| ● Monit | 监控服务 |
| ● MySQL | 数据库 |
| ● Redis | NoSQL |
| ● Memcached | NoSQL |
| ● Nginx | HTTP 代理 |
| ● Rainbows | HTTP 应用服务器 |
| ● Sidekiq | 消息队列服务 |
| ● Stunnel | TLS/SSL 网关 |
| ● Mqtt3d | 推送服务器 |

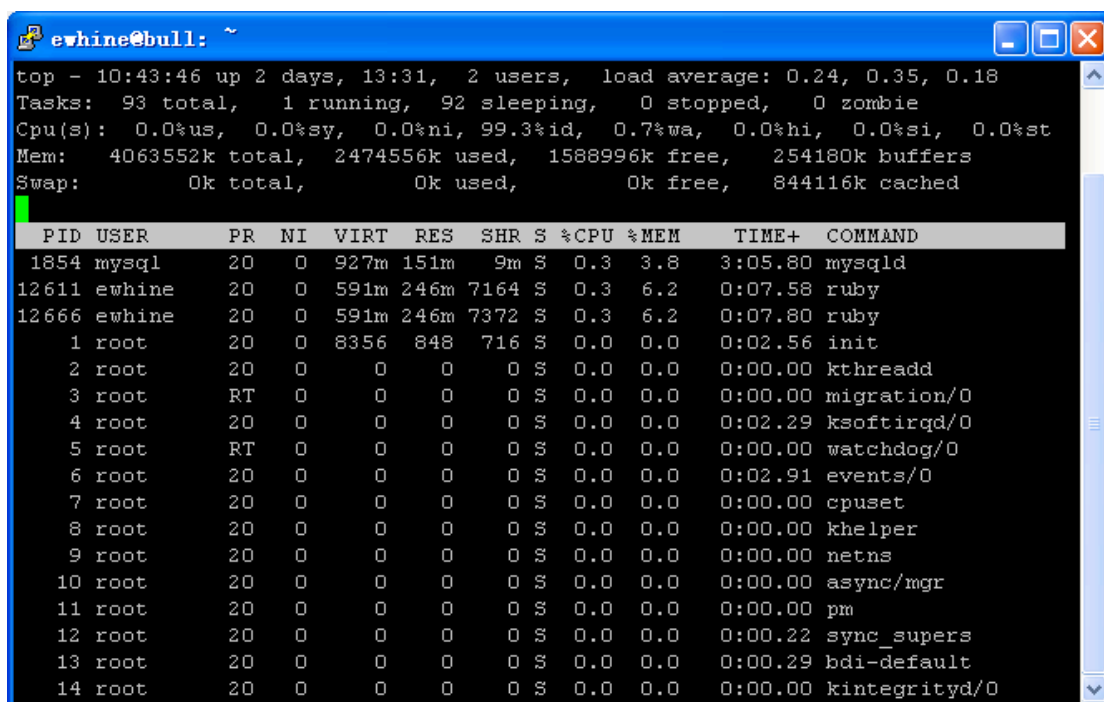
2 系统维护

敏行服务器系统运行于 Linux 平台之上，为保证系统的正常运行，需要进行日常维护。

2.1. 常用命令

Linux 系统提供了一些命令，可以用于监控系统，在此列举一些常用的命令。

top 是一个交互式的系统命令，用于显示系统当前状态(如系统负载、内存使用量，交换空间使用量、以及进程的运行状况)，top 命令的运行结果如图 2-1 所示。按“q”可退出 top 界面。



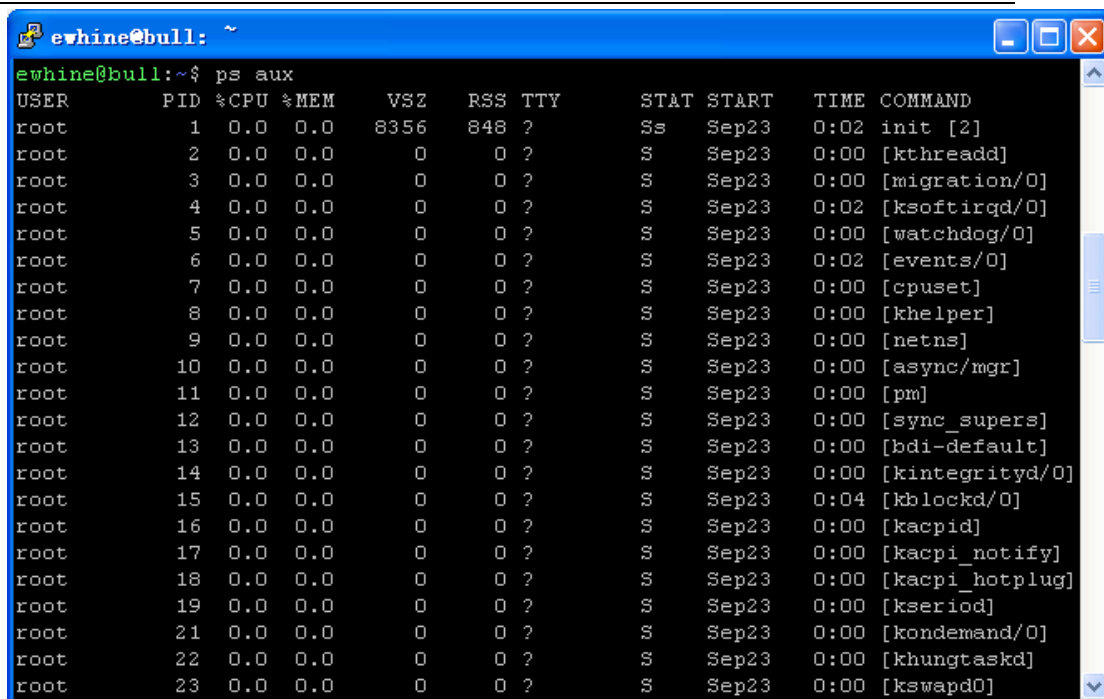
```

ewhine@bull: ~
top - 10:43:46 up 2 days, 13:31,  2 users,  load average: 0.24, 0.35, 0.18
Tasks:  93 total,   1 running, 92 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,   0.0%sy,   0.0%ni, 99.3%id,   0.7%wa,   0.0%hi,   0.0%si,   0.0%st
Mem:   4063552k total, 2474556k used, 1588996k free, 254180k buffers
Swap:      0k total,    0k used,    0k free, 844116k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 1854 mysql     20   0  927m 151m   9m S   0.3   3.8   3:05.80 mysqld
12611 ewhine    20   0  591m 246m 7164 S   0.3   6.2   0:07.58 ruby
12666 ewhine    20   0  591m 246m 7372 S   0.3   6.2   0:07.80 ruby
   1 root       20   0   8356  848   716 S  0.0   0.0   0:02.56 init
   2 root       20   0      0    0     0 S  0.0   0.0   0:00.00 kthreadd
   3 root       RT   0      0    0     0 S  0.0   0.0   0:00.00 migration/0
   4 root       20   0      0    0     0 S  0.0   0.0   0:02.29 ksoftirqd/0
   5 root       RT   0      0    0     0 S  0.0   0.0   0:00.00 watchdog/0
   6 root       20   0      0    0     0 S  0.0   0.0   0:02.91 events/0
   7 root       20   0      0    0     0 S  0.0   0.0   0:00.00 cpuset
   8 root       20   0      0    0     0 S  0.0   0.0   0:00.00 khelper
   9 root       20   0      0    0     0 S  0.0   0.0   0:00.00 netns
  10 root       20   0      0    0     0 S  0.0   0.0   0:00.00 async/mgr
  11 root       20   0      0    0     0 S  0.0   0.0   0:00.00 pm
  12 root       20   0      0    0     0 S  0.0   0.0   0:00.22 sync_supers
  13 root       20   0      0    0     0 S  0.0   0.0   0:00.29 bdi-default
  14 root       20   0      0    0     0 S  0.0   0.0   0:00.00 kintegrityd/0
  
```

图 2-1 top 命令

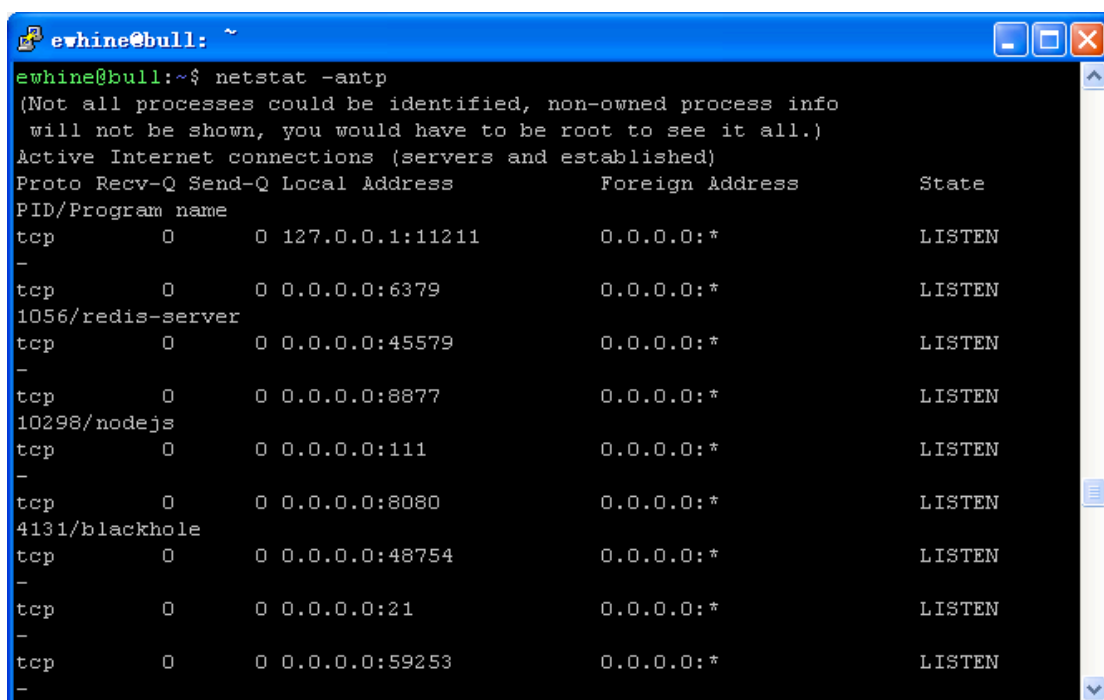
ps 是另一个查看系统中进程状态的命令，相比 top，ps 更加灵活。将 ps 和其他命令通过管道连接在一起，可以实现很多复杂的功能。图 2-2 是执行“ps aux”命令的运行结果。



```
ewhine@bull: ~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  8356   848 ?        Ss   Sep23   0:02 init [2]
root         2  0.0  0.0     0     0 ?        S    Sep23   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S    Sep23   0:00 [migration/0]
root         4  0.0  0.0     0     0 ?        S    Sep23   0:02 [ksoftirqd/0]
root         5  0.0  0.0     0     0 ?        S    Sep23   0:00 [watchdog/0]
root         6  0.0  0.0     0     0 ?        S    Sep23   0:02 [events/0]
root         7  0.0  0.0     0     0 ?        S    Sep23   0:00 [cpuset]
root         8  0.0  0.0     0     0 ?        S    Sep23   0:00 [khelper]
root         9  0.0  0.0     0     0 ?        S    Sep23   0:00 [netns]
root        10  0.0  0.0     0     0 ?        S    Sep23   0:00 [async/mgr]
root        11  0.0  0.0     0     0 ?        S    Sep23   0:00 [pm]
root        12  0.0  0.0     0     0 ?        S    Sep23   0:00 [sync_supers]
root        13  0.0  0.0     0     0 ?        S    Sep23   0:00 [bdi-default]
root        14  0.0  0.0     0     0 ?        S    Sep23   0:00 [kintegrityd/0]
root        15  0.0  0.0     0     0 ?        S    Sep23   0:04 [kblockd/0]
root        16  0.0  0.0     0     0 ?        S    Sep23   0:00 [kacpid]
root        17  0.0  0.0     0     0 ?        S    Sep23   0:00 [kacpi_notify]
root        18  0.0  0.0     0     0 ?        S    Sep23   0:00 [kacpi_hotplug]
root        19  0.0  0.0     0     0 ?        S    Sep23   0:00 [kseriod]
root        21  0.0  0.0     0     0 ?        S    Sep23   0:00 [kondemand/0]
root        22  0.0  0.0     0     0 ?        S    Sep23   0:00 [khungtaskd]
root        23  0.0  0.0     0     0 ?        S    Sep23   0:00 [kswapd0]
```

图 2-2 ps 命令

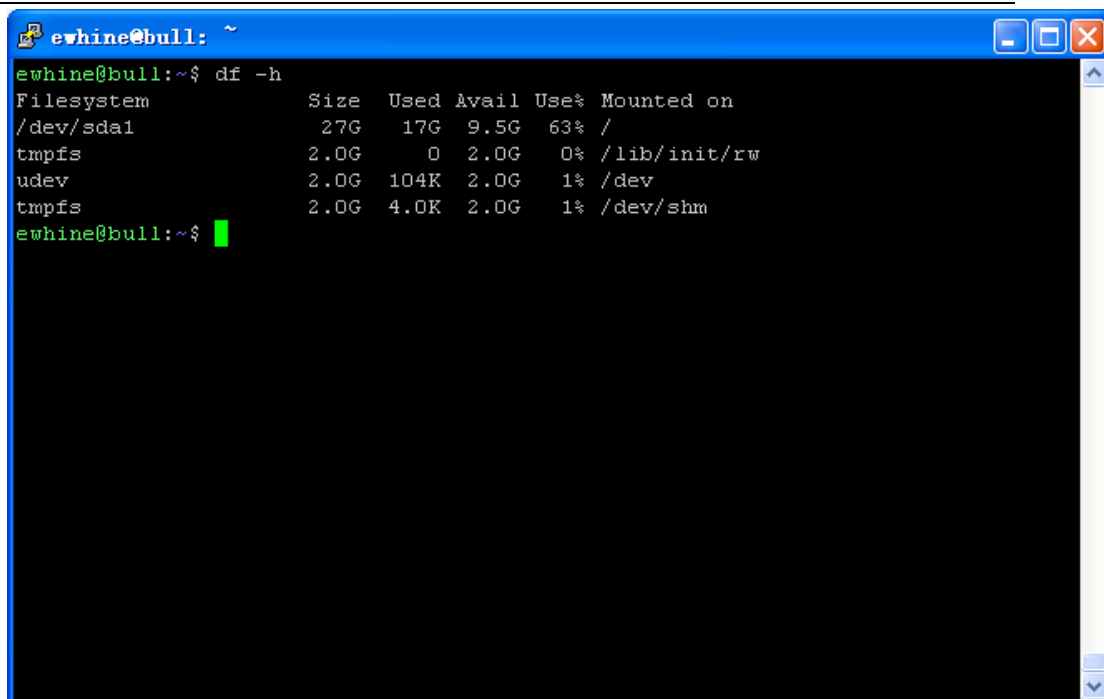
netstat 命令用于察看网络连接状况，在 Linux 平台上，通过 netstat -antp 可以查看进程和网络端口的对应关系，如图 2-3 所示。



```
ewhine@bull: ~$ netstat -antp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:11211            0.0.0.0:*                LISTEN
-
tcp        0      0 0.0.0.0:6379             0.0.0.0:*                LISTEN
1056/redis-server
tcp        0      0 0.0.0.0:45579            0.0.0.0:*                LISTEN
-
tcp        0      0 0.0.0.0:8877             0.0.0.0:*                LISTEN
10298/nodejs
tcp        0      0 0.0.0.0:111              0.0.0.0:*                LISTEN
-
tcp        0      0 0.0.0.0:8080             0.0.0.0:*                LISTEN
4131/blackhole
tcp        0      0 0.0.0.0:48754            0.0.0.0:*                LISTEN
-
tcp        0      0 0.0.0.0:21               0.0.0.0:*                LISTEN
-
tcp        0      0 0.0.0.0:59253            0.0.0.0:*                LISTEN
-
```

图 2-3 netstat 命令

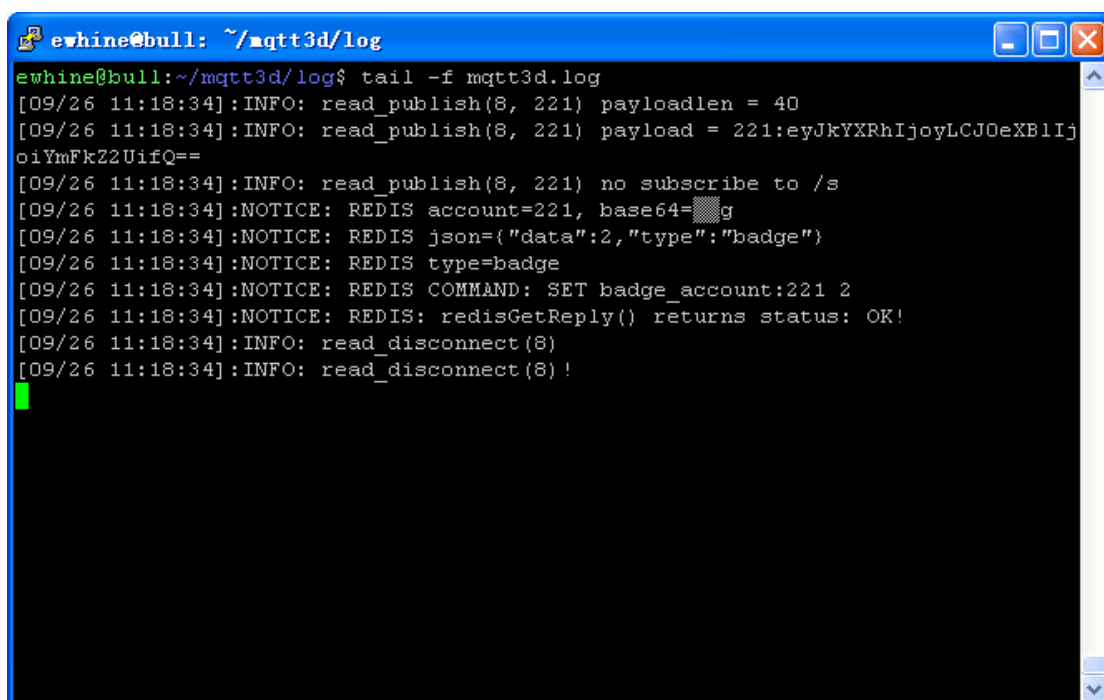
df 命令用于察看磁盘使用状况，如图 2-4 所示。-h 参数以人类可读的形式显示，否则将显示磁盘的 block 数，一般情况下，一个 block 为 512 字节或者 1024 字节。



```
ewhine@bull: ~  
ewhine@bull:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       27G   17G   9.5G   63% /  
tmpfs           2.0G    0   2.0G    0% /lib/init/rw  
udev            2.0G  104K   2.0G    1% /dev  
tmpfs           2.0G   4.0K   2.0G    1% /dev/shm  
ewhine@bull:~$
```

图 2-4 df 命令

tail -f 命令可以动态的查看日志文件，对于分析系统的问题很有帮助，图 2-5 所示为通过 tail -f 命令跟踪 mqtt 服务的日志。



```
ewhine@bull: ~/mqtt3d/log  
ewhine@bull:~/mqtt3d/log$ tail -f mqtt3d.log  
[09/26 11:18:34]:INFO: read_publish(8, 221) payloadlen = 40  
[09/26 11:18:34]:INFO: read_publish(8, 221) payload = 221:eyJkYXRhIjoyLCJ0eXB1IjoiYmFkZ2UifQ==  
[09/26 11:18:34]:INFO: read_publish(8, 221) no subscribe to /s  
[09/26 11:18:34]:NOTICE: REDIS account=221, base64= g  
[09/26 11:18:34]:NOTICE: REDIS json={"data":2,"type":"badge"}  
[09/26 11:18:34]:NOTICE: REDIS type=badge  
[09/26 11:18:34]:NOTICE: REDIS COMMAND: SET badge_account:221 2  
[09/26 11:18:34]:NOTICE: REDIS: redisGetReply() returns status: OK!  
[09/26 11:18:34]:INFO: read_disconnect(8)  
[09/26 11:18:34]:INFO: read_disconnect(8)!
```

图 2-5 通过 tail 命令跟踪日志文件

2.2. 启停服务

在大部分 Linux 发行版中(包括 Debian 和 Redhat), `/etc/init.d` 目录用于存放服务器程序的启动脚本。运行 `/etc/init.d/XXX start` 来启动服务; 运行 `/etc/init.d/XXX stop` 来停止服务; 运行 `/etc/init.d/XXX restart` 来重新启动服务; 运行 `/etc/init.d/XXX status` 察看服务器状态, 其中 XXX 代表服务脚本名称。

Redhat 还提供了一个叫做 `service` 的命令, 专门用于执行启动脚本, 在 Redhat 平台上可以使用 `service XXX start` 代替 `/etc/init.d/XXX start`。

■ 启停 monit

- ◆ 启动 `/etc/init.d/monit start`

- ◆ 停止 `/etc/init.d/monit stop`

■ 启停 MySQL

- ◆ 启动 `/etc/init.d/mysql start`

- ◆ 停止 `/etc/init.d/mysql stop`

■ 启停 Redis

- ◆ 启动 `/etc/init.d/redis start`

- ◆ 停止 `/etc/init.d/redis stop`

■ 启停 Memcached

- ◆ 启动 `/etc/init.d/memcached start`

- ◆ 停止 `/etc/init.d/memcached stop`

■ 启停 Nginx

- ◆ 启动 `/etc/init.d/nginx start`

- ◆ 停止 `/etc/init.d/nginx stop`

■ 启停 Rainbows

- ◆ 启动 `/etc/init.d/rainbows start`

- ◆ 停止 `/etc/init.d/rainbows stop`

■ 启停 Sidekiq

- ◆ 启动 `/etc/init.d/sidekiq start`

- ◆ 停止 `/etc/init.d/sidekiq stop`

- 启停 Stunnel
 - ◆ 启动 `/etc/init.d/stunnel start`
 - ◆ 停止 `/etc/init.d/stunnel stop`
- 启停 Mqtt3d
 - ◆ 启动 `/etc/init.d/mqtt3d start`
 - ◆ 停止 `/etc/init.d/mqtt3d stop`

2.3. 备份数据

备份数据之前请确保已经将 `monit`、`Rainbows` 以及 `sidekiq` 服务关闭，但是要确保 `redis` 和 `MySQL` 处于运行状态。

运行如下命令进行备份：

- `ruby /home/ewhine/minxing_ctl.rb backup`

3 敏行后台维护系统

敏行后台维护系统是一套基于 `web` 的服务器管理工具，可通过可视化的界面对服务器进行日常维护，降低维护的工作量。

3.1. 系统功能

敏行后台维护系统包括“许可证信息”、“许可证更新”、“服务状态”、“重启服务”、“启动服务”、“停止服务”、“备份数据”和“产品更新”功能。日后会陆续增加新功能。

3.2. 详细说明

为敏行后台维护系统，首先需要登陆。登陆界面如图 3-1 所示。



图 3-1 登陆界面

登陆成功之后即显示主界面。主界面的左侧是菜单项，罗列各项功能。

“许可证信息”的功能时显示许可证的版本、有效期和用户限制，如图 3-2 所示。



图 3-2 许可证信息

“许可证更新”功能用于更新到期或者作废的许可证。注意，在使用此功能前，请确保所有服务已经关闭。在“停止服务”页面可以关闭服务。如果不能确定服务是否关闭，可以在“服务状态”页面查看服务的状态。“许可证更新”如图 3-3 所示”。

在进行“许可证更新”操作时，需要将编码过的许可证信息粘贴到文本框中。



图 3-3 许可证更新

“服务状态”功能可显示主要服务的启动/停止信息。在进行启动服务、停止服务以及更新许可证之前，需要将服务关闭。通过“服务状态”功能，可以判断服务是否已经关闭。“服务状态”入图 3-4 所示。



图 3-4 服务状态

“重启服务”功能用于重新系统中的主要服务。在服务已经关闭的情况下执行“重启服务”相当于“启动服务”，但是系统会报告停止服务失败的信息。在进行“重启服务”之前可先通过“服务状态”功能察看服务是否处于启动状态。“重启服务”如图 3-5 所示。



图 3-5 重启服务

在进行"启动服务"操作前，请确保所有服务已经关闭。如果服务已经启动，执行"启动服务"操作将会返回未知的错误。“启动服务”如图 3-6 所示。



图 3-6 启动服务

在进行"停止服务"操作前，请确保所有服务已经启动。如果服务并未启动，

执行“停止服务”操作将会返回未知的错误。“停止服务”如图 3-7 所示。



图 3-7 停止服务

“备份数据”的主要目的是对数据库进行备份, 在执行此操作之前, 请确保已经关闭服务, 否则将导致数据一致性问题。“备份数据”如图 3-8 所示。



图 3-8 备份数据

敏行后台维护系统的最主要功能是“产品更新”, 通过此功能, 用户可以很

方便的升级敏行系统。在进行更新之前，请确保服务已经关闭，如果没有关闭服务就进行更新操作，将有可能遇到未知错误。进行产品更新时，需要上传敏行升级安装包，然后点击“更新”按钮一件更新。



图 3-9 产品更新