

[http://www.itshanghai.net/technology/2009/0209/article\\_238.html](http://www.itshanghai.net/technology/2009/0209/article_238.html)

另外,可以参考这

个[http://www.centos.org/docs/5/html/5.1/Deployment\\_Guide/ch-proc.html](http://www.centos.org/docs/5/html/5.1/Deployment_Guide/ch-proc.html)

- 内容摘要: Linux系统上的/proc目录是一种文件系统,即proc文件系统。

Linux系统上的/proc目录是一种文件系统,即proc文件系统。与其它常见的文件系统不同的是, /proc是一种伪文件系统(也即虚拟文件系统),存储的是当前内核运行状态的一系列特殊文件,用户可以通过这些文件查看有关系统硬件及当前正在运行进程的信息,甚至可以通过更改其中某些文件来改变内核的运行状态。

基于/proc文件系统如上所述的特殊性,其内的文件也常被称作虚拟文件,并具有一些独特的特点。例如,其中有些文件虽然使用查看命令查看时会返回大量信息,但文件本身的大小却会显示为0字节。此外,这些特殊文件中大多数文件的时间及日期属性通常为当前系统时间和日期,这跟它们随时会被刷新(存储于RAM中)有关。

为了查看及使用上的方便,这些文件通常会按照相关性进行分类存储于不同的目录甚至子目录中,如/proc/scsi目录中存储的就是当前系统上所有SCSI设备的相关信息, /proc/N中存储的则是系统当前正在运行的进程的相关信息,其中N为正在运行的进程(可以想象得到,在某进程结束后其相关目录则会消失)。

大多数虚拟文件可以使用文件查看命令如cat、more或者less进行查看,有些文件信息表述的内容可以一目了然,但也有文件的信息却不怎么具有可读性。不过,这些可读性较差的文件在使用一些命令如apm、free、lspci或top查看时却可以有着不错的表现。

## 一、 进程目录中的常见文件介绍

/proc目录中包含许多以数字命名的子目录,这些数字表示系统当前正在运行进程的进程号,里面包含对应进程相关的多个信息文件。

```
[root@rhe15 ~]# ll /proc
total 0
dr-xr-xr-x  5 root    root      0 Feb  8 17:08 1
dr-xr-xr-x  5 root    root      0 Feb  8 17:08 10
dr-xr-xr-x  5 root    root      0 Feb  8 17:08 11
dr-xr-xr-x  5 root    root      0 Feb  8 17:08
1156
dr-xr-xr-x  5 root    root      0 Feb  8 17:08
139
dr-xr-xr-x  5 root    root      0 Feb  8 17:08
140
dr-xr-xr-x  5 root    root      0 Feb  8 17:08
141
dr-xr-xr-x  5 root    root      0 Feb  8 17:09
1417
dr-xr-xr-x  5 root    root      0 Feb  8 17:09
1418
```

上面列出的是/proc目录中一些进程相关的目录,每个目录中是当程本身相关信息的文件。下面是作者系统(RHEL5.3)上运行的一个PID为2674的进程saslauthd的相关文件,其中有些文件是每个进程都会具有的,后文会对这些常见文件做出说明。

```
[root@rhe15 ~]# ll /proc/2674
total 0
dr-xr-xr-x  2 root    root      0 Feb  8 17:15 attr
-r-----  1 root    root      0 Feb  8 17:14 auxv
```

导航

[博客园](#) [首页](#) [联系](#) [管理](#)

2011年4月						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

公告

昵称:[zhezhelein](#)

园龄:[5年2个月](#)

粉丝:[94](#)

关注:[4](#)

[+加关注](#)

统计

随笔 - 856 文章 - 0 评论 - 33

搜索

  

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

我的标签

[QThread\(6\)](#)

[BFS求最短路\(2\)](#)

[event\(2\)](#)

[DFS求连通块\(1\)](#)

[QTimer\(1\)](#)

[signal slot\(1\)](#)

随笔分类

[android\(67\)](#)

[bash\(2\)](#)

[bfs\(2\)](#)

[BFS求最短路\(5\)](#)

[c++\(11\)](#)

[dfs\(7\)](#)

[dot\(1\)](#)

[javascript\(5\)](#)

[kruskal\(1\)](#)

[linux驱动\(12\)](#)

[makefile\(1\)](#)

[qml\(25\)](#)

[Qt\(79\)](#)

[queue\(1\)](#)

[rockchip\(1\)](#)

[stl\(2\)](#)

[svn\(1\)](#)

[TODC\(1\)](#)

```

-r--r--r-- 1 root root 0 Feb 8 17:09 cmdline
-rw-r--r-- 1 root root 0 Feb 8 17:14 coredump_filter
-r--r--r-- 1 root root 0 Feb 8 17:14 cpuset
lrwxrwxrwx 1 root root 0 Feb 8 17:14 cwd ->
/var/run/saslauthd
-r----- 1 root root 0 Feb 8 17:14 environ
lrwxrwxrwx 1 root root 0 Feb 8 17:09 exe ->
/usr/sbin/saslauthd
dr-x----- 2 root root 0 Feb 8 17:15 fd
-r----- 1 root root 0 Feb 8 17:14 limits
-rw-r--r-- 1 root root 0 Feb 8 17:14 loginuid
-r--r--r-- 1 root root 0 Feb 8 17:14 maps
-rw----- 1 root root 0 Feb 8 17:14 mem
-r--r--r-- 1 root root 0 Feb 8 17:14 mounts
-r----- 1 root root 0 Feb 8 17:14 mountstats
-rw-r--r-- 1 root root 0 Feb 8 17:14 oom_adj
-r--r--r-- 1 root root 0 Feb 8 17:14 oom_score
lrwxrwxrwx 1 root root 0 Feb 8 17:14 root -> /
-r--r--r-- 1 root root 0 Feb 8 17:14 schedstat
-r----- 1 root root 0 Feb 8 17:14 smaps
-r--r--r-- 1 root root 0 Feb 8 17:09 stat
-r--r--r-- 1 root root 0 Feb 8 17:14 statm
-r--r--r-- 1 root root 0 Feb 8 17:10 status
dr-xr-xr-x 3 root root 0 Feb 8 17:15 task
-r--r--r-- 1 root root 0 Feb 8 17:14 wchan

```

1.1、cmdline — 启动当前进程的完整命令，但僵尸进程目录中的此文件不包含任何信息；

```

[root@rhel5 ~]# more /proc/2674/cmdline
/usr/sbin/saslauthd

```

1.2、cwd — 指向当前进程运行目录的一个符号链接；

1.3、environ — 当前进程的环境变量列表，彼此间用空字符(NULL)隔开；变量用大写字母表示，其值用小写字母表示；

```

[root@rhel5 ~]# more /proc/2674/environ
TERM=linuxauthd

```

1.4、exe — 指向启动当前进程的可执行文件(完整路径)的符号链接，通过/proc/N/exe可以启动当前进程的一个拷贝；

1.5、fd — 这是个目录，包含当前进程打开的每一个文件的文件描述符(file descriptor)，这些文件描述符是指向实际文件的一个符号链接；

```

[root@rhel5 ~]# ll /proc/2674/fd
total 0
lrwx----- 1 root root 64 Feb 8 17:17 0 -> /dev/null
lrwx----- 1 root root 64 Feb 8 17:17 1 -> /dev/null
lrwx----- 1 root root 64 Feb 8 17:17 2 -> /dev/null
lrwx----- 1 root root 64 Feb 8 17:17 3 -> socket:[7990]
lrwx----- 1 root root 64 Feb 8 17:17 4 ->
/var/run/saslauthd/saslauthd.pid
lrwx----- 1 root root 64 Feb 8 17:17 5 -> socket:[7991]
lrwx----- 1 root root 64 Feb 8 17:17 6 ->
/var/run/saslauthd/mux.accept

```

1.6、limits — 当前进程所使用的每一个受限资源的软限制、硬限制和管理单元；此文件仅可由实际启动当前进程的UID用户读取；(2.6.24以后的内核版本支持此功能)；

1.7、maps — 当前进程关联到的每个可执行文件和库文件在内存中的映射区域及其访问权限所组成的列表；

```

[root@rhel5 ~]# cat /proc/2674/maps
00110000-00239000 r-xp 00000000 08:02 130647
/lib/libcrypto.so.0.9.8e
00239000-0024c000 rwxp 00129000 08:02 130647

```

[1000\(1\)](#)  
[ubuntu\(12\)](#)  
[编译原理\(1\)](#)  
[并查集\(12\)](#)  
[二叉树\(6\)](#)  
[二分\(2\)](#)  
[分类讨论\(1\)](#)  
[浮点\(1\)](#)  
[回溯\(9\)](#)  
[计数排序\(2\)](#)  
[经验\(4\)](#)  
[枚举\(6\)](#)  
[逆序对\(2\)](#)  
[排列树\(2\)](#)  
[排序\(6\)](#)  
[嵌入式linux\(1\)](#)  
[树\(1\)](#)  
[四叉树\(1\)](#)  
[算法导论\(1\)](#)  
[算法竞赛入门经典\(47\)](#)  
[随机\(1\)](#)  
[贪心\(1\)](#)  
[挑战编程\(6\)](#)  
[挑战程序设计竞赛\(1\)](#)  
[图遍历\(3\)](#)  
[栈\(4\)](#)  
[子集树\(1\)](#)  
[组合\(2\)](#)

#### 随笔档案

[2015年12月 \(12\)](#)  
[2015年11月 \(19\)](#)  
[2015年10月 \(29\)](#)  
[2015年9月 \(20\)](#)  
[2015年8月 \(14\)](#)  
[2015年7月 \(14\)](#)  
[2015年4月 \(2\)](#)  
[2015年3月 \(2\)](#)  
[2015年1月 \(1\)](#)  
[2014年12月 \(9\)](#)  
[2014年11月 \(28\)](#)  
[2014年10月 \(6\)](#)  
[2014年9月 \(3\)](#)  
[2014年8月 \(17\)](#)  
[2014年7月 \(8\)](#)  
[2014年6月 \(9\)](#)  
[2014年5月 \(3\)](#)  
[2014年4月 \(22\)](#)  
[2014年3月 \(14\)](#)  
[2013年11月 \(1\)](#)  
[2013年10月 \(7\)](#)  
[2013年2月 \(1\)](#)  
[2012年10月 \(1\)](#)  
[2012年9月 \(23\)](#)  
[2012年8月 \(8\)](#)  
[2012年7月 \(14\)](#)  
[2012年6月 \(20\)](#)  
[2012年5月 \(27\)](#)

```
/lib/libcrypto.so.0.9.8e
0024c000-00250000 rwxp 0024c000 00:00 0
00250000-00252000 r-xp 00000000 08:02 130462
/lib/libdl-2.5.so
00252000-00253000 r-xp 00001000 08:02 130462
/lib/libdl-2.5.so
```

1.8、mem — 当前进程所占用的内存空间，由open、read和lseek等系统调用使用，不能被用户读取；

1.9、root — 指向当前进程运行根目录的符号链接；在Unix和Linux系统上，通常采用chroot命令使每个进程运行于独立的根目录；

1.10、stat — 当前进程的状态信息，包含一系统格式化后的数据列，可读性差，通常由ps命令使用；

1.11、statm — 当前进程占用内存的状态信息，通常以“页面”(page)表示；

1.12、status — 与stat所提供信息类似，但可读性较好，如下所示，每行表示一个属性信息；其详细介绍请参见 proc的man手册页；

```
[root@rhel5 ~]# more /proc/2674/status
Name:   saslauthd
State:  S (sleeping)
SleepAVG:    0%
Tgid:    2674
Pid:     2674
PPid:    1
TracerPid:  0
Uid:    0      0      0      0
Gid:    0      0      0      0
FDSize: 32
Groups:
VmPeak:   5576 kB
VmSize:   5572 kB
VmLck:    0 kB
VmHWM:    696 kB
VmRSS:    696 kB
■ ■ ■ ■
```

1.13、task — 目录文件，包含由当前进程所运行的每一个线程的相关信息，每个线程的相关信息文件均保存在一个由线程号(tid)命名的目录中，这类似于其内容类似于每个进程目录中的内容；(内核2.6版本以后支持此功能)

## 二、/proc目录下常见的文件介绍

### 2.1、/proc/apm

高级电源管理(APM)版本信息及电池相关状态信息，通常由apm命令使用；

### 2.2、/proc/buddyinfo

用于诊断内存碎片问题的相关信息文件；

### 2.3、/proc/cmdline

在启动时传递至内核的相关参数信息，这些信息通常由lilo或grub等启动管理工具进行传递；

```
[root@rhel5 ~]# more /proc/cmdline
ro root=/dev/VolGroup00/LogVol00 rhgb quiet
```

### 2.4、/proc/cpuinfo

处理器的相关信息的文件；

### 2.5、/proc/crypto

系统上已安装的内核使用的密码算法及每个算法的详细信息列表；

[2012年12月\(167\)](#)

[2012年4月\(10\)](#)

[2012年3月\(8\)](#)

[2012年2月\(7\)](#)

[2012年1月\(2\)](#)

[2011年12月\(19\)](#)

[2011年11月\(32\)](#)

[2011年10月\(28\)](#)

[2011年9月\(50\)](#)

[2011年8月\(56\)](#)

[2011年7月\(61\)](#)

[2011年6月\(56\)](#)

[2011年5月\(53\)](#)

[2011年4月\(74\)](#)

[2011年3月\(47\)](#)

[2011年2月\(19\)](#)

## 文章分类

[programming-challenges](#)

## 最新评论

[1. Re:确定Windows XP到底是UCS-2的还是UTF-16的](#)

作者你好,为什么用vs2010测试,根本无法显示这3个字符(3个方框)

项目已经设为unicode字符集了

--叶子如花

[2. Re:Ubuntu 14.04 配置 LAMP+phpMyAdmin PHP开发环境!](#)

博主您好,我也是刚学linux在配置php环境上出现了一点小问题,可以帮我一下吗?这是我的邮箱

1798041060@qq.com

--学沫沫

[3. Re:正规式->最小化DFA说明](#)

简洁,清晰,精准!好文章!

--IceDream61

[4. Re:解决ubuntu下无线网卡不能启用](#)

我的电脑用UEFI模式装ubuntu就会有这种问题:第一次开机, rfkill, pyh0 hard blocked, wifi开关快捷键还改不了它;用Legacy模式装就正常,第一次是开的,快捷键也能正常.....

--xusiwei

[5. Re:解决ubuntu下无线网卡不能启用](#)

mark

我的笔记本:Acer 4750G,

网卡:Atheros AR9287

--xusiwei

## 阅读排行榜

[1. Linux下Crontab的使用\(34852\)](#)

[2. 组播MAC地址\(23466\)](#)

[3. 深入理解linux系统下proc文件系统内容\(16824\)](#)

[4. 解决ubuntu下无线网卡不能启用\(15471\)](#)

[5. Win7+Ubuntu11.10\(EasyBCD硬盘安装\)\(14682\)](#)

[6. linux shell 中判断字符串为空的正确方法\(14430\)](#)

[7. diff和patch使用指南\(12534\)](#)

[8. TortoiseGit使用入门\(12520\)](#)

[9. Linux 指定运行时动态库路径\(12135\)](#)

[10. android studio 新建项目 里面一直卡在](#)

Web2PDF

converted by Web2PDFConvert.com

```
[root@rhe15 ~]# more /proc/crypto
name       : crc32c
driver     : crc32c-generic
module     : kernel
priority   : 0
type       : digest
blocksize  : 32
digestsize : 4
■■■■
```

## 2.6、/proc/devices

系统已经加载的所有块设备和字符设备的信息, 包含主设备号和设备组(与主设备号对应的设备类型)名;

```
[root@rhe15 ~]# more /proc/devices
Character devices:
 1 mem
 4 /dev/uc/0
 4 tty
 4 ttyS
■■■■

Block devices:
 1 ramdisk
 2 fd
 8 sd
■■■■
```

## 2.7、/proc/diskstats

每块磁盘设备的磁盘I/O统计信息列表;(内核2.5.69以后的版本支持此功能)

## 2.8、/proc/dma

每个正在使用且注册的ISA DMA通道的信息列表;

```
[root@rhe15 ~]# more /proc/dma
2: floppy
4: cascade
```

## 2.9、/proc/execdomains

内核当前支持的执行域(每种操作系统独特“个性”)信息列表;

```
[root@rhe15 ~]# more /proc/execdomains
0-0      Linux      [kernel]
```

## 2.10、/proc/fb

帧缓冲设备列表文件, 包含帧缓冲设备的设备号和相关驱动信息;

## 2.11、/proc/filesystems

当前被内核支持的文件系统类型列表文件, 被标示为nodev的文件系统表示不需要块设备的支持;通常mount一个设备时, 如果没有指定文件系统类型将通过此文件来决定其所需文件系统的类型;

```
[root@rhe15 ~]# more /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    proc
          iso9660
          ext3
■■■■
■■■■
```

## 2.12、/proc/interrupts

[10. android studio 新建项目 介面二程序区 \["building `项目名` gradle project info"\] \(9920\)](#)  
[11. I2C驱动\(9020\)](#)  
[12. nsswitch.conf文件详解\(8462\)](#)  
[13. 如何使用QScrollArea\(8202\)](#)  
[14. sd卡驱动\(8166\)](#)  
[15. Linux下 解包/打包 Android 映像文件 system.img, boot.img, ramdisk.img, userdata.img.\(7976\)](#)

### 评论排行榜

[1. 深入理解linux系统下proc文件系统内容\(5\)](#)  
[2. 八度音阶和频率的关\(3\)](#)  
[3. 如何在Qt中使用自定义数据类型\(2\)](#)  
[4. 解决ubuntu下无线网卡不能启用\(2\)](#)  
[5. Ubuntu+OpenGL程序设计\(1\)](#)

### 推荐排行榜

[1. 正规式->最小化DFA说明\(3\)](#)  
[2. TortoiseGit使用入门\(2\)](#)  
[3. Win7+Ubuntu11.10\(EasyBCD硬盘安装\)\(2\)](#)  
[4. buildroot 编译udev 173 出错 'SOCK\\_NONBLOCK' undeclared\(1\)](#)  
[5. Qt Creator and Clang\(1\)](#)  
[6. 一些距离测算方法\(1\)](#)  
[7. makefile 自动化变量\(1\)](#)  
[8. 如何在Ubuntu 14.10 上安装WordPress?\(1\)](#)  
[9. mount --bind 的妙用\(1\)](#)  
[10. Linux 指定运行时动态库路径\(1\)](#)  
[11. qt 工程中包含另一个工程\(1\)](#)  
[12. 如何在Qt中使用自定义数据类型\(1\)](#)  
[13. 网卡工作原理\(1\)](#)  
[14. 片选\(1\)](#)  
[15. PHY芯片\(1\)](#)

Powered by:

[博客园](#)

Copyright © zhezheLin

X86或X86\_64体系架构系统上每个IRQ相关的中断号列表;多路处理器平台上每个CPU对于每个I/O设备均有自己的中断号;

```
[root@rhe15 ~]# more /proc/interrupts
CPU0
 0:      1305421      IO-APIC-edge  timer
 1:         61      IO-APIC-edge  i8042
185:      1068      IO-APIC-level  eth0
■■■■
```

### 2.13. /proc/iomem

每个物理设备上的记忆体(RAM或者ROM)在系统内存中的映射信息;

```
[root@rhe15 ~]# more /proc/iomem
00000000-0009f7ff : System RAM
0009f800-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
■■■■
```

#### 3.2.12. /proc/iomem

[http://www.centos.org/docs/5/html/5.1/Deployment\\_Guide/s2-proc-iomem.html](http://www.centos.org/docs/5/html/5.1/Deployment_Guide/s2-proc-iomem.html)

This file shows you the current map of the system's memory for each physical device:

```
00000000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-07ffffff : System RAM
00100000-00291ba8 : Kernel code
00291ba9-002e09cb : Kernel data
e0000000-e3ffffff : VIA Technologies, Inc. VT82C597 [Apollo VP3]
e4000000-e7ffffff : PCI Bus #01
e4000000-e4003fff : Matrox Graphics, Inc. MGA G200 AGP
e5000000-e57ffffff : Matrox Graphics, Inc. MGA G200 AGP
e8000000-e8ffffff : PCI Bus #01
e8000000-e8ffffff : Matrox Graphics, Inc. MGA G200 AGP
ea000000-ea00007f : Digital Equipment Corporation DECchip 21140
[FasterNet]
ea000000-ea00007f : tulip ffff0000-ffffffff : reserved
```

The first column displays the **memory registers** used by each of the different types of memory. The second column lists the kind of memory located within those registers and displays which memory registers are used by the kernel within the system RAM or, if the network interface card has multiple Ethernet ports, the memory registers assigned for each port.

```
/opt/qtmarvell/mvqt # cat /proc/iomem
```

```
00000000-0ffffff : System RAM
 0002b000-00554fff : Kernel text
 00556000-005a22f3 : Kernel data
10000000-1ffffff : System RAM
e0000000-e7ffffff : PEX0 Memory
 e0000000-e0003fff : 0000:00:01.0
 e0004000-e0004fff : 0000:00:01.0
e8000000-efffffff : PEX1 Memory
f1012100-f10121ff : serial8250.0
 f1012100-f101211f : serial
f1090000-f10903ff : mvsdio
 f1090000-f10903ff : mvsdio
f10a0000-f10a3fff : mv88fx_snd.0
 f10a0000-f10a3fff : mv88fx_snd
```

```
f10c0000-f10d0000 : dovefb.0 寄存器地址映射到这个
f10c0000-f10d0000 : dovefb_ovly.0
```

```
/opt/qtmarvell/mvqt # ./framebuffer
The framebuffer device was opened successfully.
```

Fixed screen info:

```
id:      GFX Layer 0
smem_start: 0x12000000
smem_len: 33554432
type:     0
type_aux: 0
visual:   2
xpanstep: 1
ypanstep: 1
ywrapstep: 0
line_length: 2560
mmio_start: 0xf10c0000
mmio_len: 65537
```

#### 2.14、/proc/ioprots

当前正在使用且已经注册过的与物理设备进行通讯的输入-输出端口范围信息列表;如下面所示, 第一列表示注册的I/O端口范围, 其后表示相关的设备;

```
[root@rhe15 ~]# less /proc/ioprots
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
■■■■
```

#### 2.15、/proc/kallsyms

模块管理工具用来动态链接或绑定可装载模块的符号定义, 由内核输出; (内核2.5.71以后的版本支持此功能); 通常这个文件中的信息量相当大;

```
[root@rhe15 ~]# more /proc/kallsyms
c04011f0 T _stext
c04011f0 t run_init_process
c04011f0 T stext
■■■■
```

#### 2.16、/proc/kcore

系统使用的物理内存, 以ELF核心文件(core file)格式存储, 其文件大小为已使用的物理内存(RAM)加上4KB; 这个文件用来检查内核数据结构的当前状态, 因此, 通常由GBD通常调试工具使用, 但不能使用文件查看命令打开此文件;

#### 2.17、/proc/kmsg

此文件用来保存由内核输出的信息, 通常由/sbin/klogd或/bin/dmmsg等程序使用, 不要试图使用查看命令打开此文件;

#### 2.18、/proc/loadavg

保存关于CPU和磁盘I/O的负载平均值, 其前三列分别表示每1秒钟、每5秒钟及每15秒的负载平均值, 类似于uptime命令输出的相关信息; 第四列是由斜线隔开的两个数值, 前者表示当前正由内核调度的实体(进程和线程)的数目, 后者表示系统当前存活的内核调度实体的数目; 第五列表示此文件被查看前最近一个由内核创建的进程的PID;

```
[root@rhe15 ~]# more /proc/loadavg
0.45 0.12 0.04 4/125 5549

[root@rhe15 ~]# uptime
06:00:54 up 1:06, 3 users, load average: 0.45, 0.12, 0.04
```

### 2.19、/proc/locks

保存当前由内核锁定的文件的相关信息, 包含内核内部的调试数据; 每个锁定占据一行, 且具有一个惟一的编号; 如下输出信息中每行的第二列表示当前锁定使用的锁定类别, POSIX表示目前较新类型的文件锁, 由lockf系统调用产生, FLOCK是传统的UNIX文件锁, 由flock系统调用产生; 第三列也通常由两种类型, ADVISORY表示不允许其他用户锁定此文件, 但允许读取, MANDATORY表示此文件锁定期间不允许其他用户任何形式的访问;

```
[root@rhel5 ~]# more /proc/locks
1: POSIX  ADVISORY  WRITE 4904 fd:00:4325393 0 EOF
2: POSIX  ADVISORY  WRITE 4550 fd:00:2066539 0 EOF
3: FLOCK  ADVISORY  WRITE 4497 fd:00:2066533 0 EOF
```

### 2.20、/proc/mdstat

保存RAID相关的多块磁盘的当前状态信息, 在没有使用RAID机器上, 其显示为如下状态:

```
[root@rhel5 ~]# less /proc/mdstat
Personalities :
unused devices: <none>
```

### 2.21、/proc/meminfo

系统中关于当前内存的利用状况等的信息, 常由free命令使用; 可以使用文件查看命令直接读取此文件, 其内容显示为两列, 前者为统计属性, 后者为对应的值;

```
[root@rhel5 ~]# less /proc/meminfo
MemTotal:      515492 kB
MemFree:       8452 kB
Buffers:       19724 kB
Cached:        376400 kB
SwapCached:    4 kB
■■■■■
```

### 2.22、/proc/mounts

在内核2.4.29版本以前, 此文件的内容为系统当前挂载的所有文件系统, 在2.4.19以后的内核中引进了每个进程使用独立挂载名称空间的方式, 此文件则随之变成了指向/proc/self/mounts(每个进程自身挂载名称空间中的所有挂载点列表)文件的符号链接; /proc/self是一个独特的目录, 后文中会对此目录进行介绍;

```
[root@rhel5 ~]# ll /proc |grep mounts
lrwxrwxrwx 1 root root 11 Feb 8 06:43
mounts -> self/mounts
```

如下所示, 其中第一列表示挂载的设备, 第二列表示在当前目录树中的挂载点, 第三点表示当前文件系统的类型, 第四列表示挂载属性(ro或者rw), 第五列和第六列用来匹配/etc/mtab文件中的转储(dump)属性;

```
[root@rhel5 ~]# more /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw,data=ordered 0 0
/dev /dev tmpfs rw 0 0
/proc /proc proc rw 0 0
/sys /sys sysfs rw 0 0
/proc/bus/usb /proc/bus/usb usbfs rw 0 0
■■■■■
```

### 2.23、/proc/modules

当前装入内核的所有模块名称列表, 可以由lsmod命令使用, 也可以直接查看; 如下所示, 其中第一列表示模块名, 第二列表示此模块占用内存空间大小, 第三列表示此模块有多少实例被装入, 第四列表示此模块依赖于其它哪些模块, 第五列表示此模块的装载状态(Live: 已经装入; Loading: 正在装入; Unloading: 正在卸载), 第六列表示此模块在内核内存(kernel memory)中的偏移量;

```
[root@rhel5 ~]# more /proc/modules
autofs4 24517 2 - Live 0xe09f7000
```



```
hidp 23105 2 - Live 0xe0a06000
rfcomm 42457 0 - Live 0xe0ab3000
l2cap 29505 10 hidp,rfcomm, Live 0xe0aaa000
■■■■
```

## 2.24、/proc/partitions

块设备每个分区的主设备号(major)和次设备号(minor)等信息,同时包括每个分区所包含的块(block)数目(如下面输出中第三列所示);

```
[root@rhel5 ~]# more /proc/partitions
major minor #blocks name

 8      0 20971520 sda
 8      1  104391 sda1
 8      2  6907950 sda2
 8      3  5630782 sda3
 8      4      1 sda4
 8      5  3582463 sda5
```

## 2.25、/proc/pci

内核初始化时发现的所有PCI设备及其配置信息列表,其配置信息多为某PCI设备相关IRQ信息,可读性不高,可以用"/sbin/lspci -vb"命令获得较易理解的相关信息;在2.6内核以后,此文件已为/proc/bus/pci目录及其下的文件代替;

## 2.26、/proc/slabinfo

在内核中频繁使用的对象(如inode、dentry等)都有自己的cache,即slab pool,而/proc/slabinfo文件列出了这些对象相关slap的信息;详情可以参见内核文档中slapinfo的手册页;

```
[root@rhel5 ~]# more /proc/slabinfo
slabinfo - version: 2.1
# name          <active_objs> <num_objs> <objsize>
<objperslab> <pagesperslab> : tunables <limit> <batchcount>
<sharedfactor> : slabdata <active_slabs> <num_slabs> <sharedavail>
rpc_buffers      8      8  2048    2    1 :
tunables   24   12    8 : slabdata      4    4    0
rpc_tasks        8     20   192   20    1 :
tunables  120   60    8 : slabdata      1    1    0
rpc_inode_cache   6      9   448    9    1 :
tunables   54   27    8 : slabdata      1    1    0
■■■■
■■■■
■■■■
```

## 2.27、/proc/stat

实时追踪自系统上次启动以来的多种统计信息;如下所示,其中,

"cpu"行后的八个值分别表示以1/100(jiffies)秒为单位的统计值(包括系统运行于用户模式、低优先级用户模式、运系统模式、空闲模式、I/O等待模式的时间等);

"intr"行给出中断的信息,第一个为自系统启动以来,发生的所有的中断的次数;然后每个数对应一个特定的中断自系统启动以来所发生的次数;

"ctxt"给出了自系统启动以来CPU发生的上下文交换的次数。

"btime"给出了从系统启动到现在为止的时间,单位为秒;

"processes (total\_forks)"自系统启动以来所创建的任务的个数目;

"procs\_running":当前运行队列的任务的数目;

"procs\_blocked":当前被阻塞的任务的数目;

```
[root@rhel5 ~]# more /proc/stat
cpu 2751 26 5771 266413 2555 99 411 0
cpu0 2751 26 5771 266413 2555 99 411 0
intr 2810179 2780489 67 0 3 3 0 5 0 1 0 0 0 1707 0 0 9620 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```



```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 5504 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 12781 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
ctxt 427300
btime 1234084100
processes 3491
procs_running 1
procs_blocked 0

```

## 2.28、/proc/swaps

当前系统上的交换分区及其空间利用信息,如果有多个交换分区的话,则会每个交换分区的信息分别存储于/proc/swap目录中的单独文件中,而其优先级数字越低,被使用到的可能性越大;下面是作者系统中只有一个交换分区时的输出信息:

```
[root@rhel5 ~]# more /proc/swaps
```

Filename	Type
Size	Used
Priority	
/dev/sda8	partition
642560	0
-1	

## 2.29、/proc/uptime

系统上次启动以来的运行时间，如下所示，其第一个数字表示系统运行时间，第二个数字表示系统空闲时间，单位是秒；

```
[root@rhel5 ~]# more /proc/uptime
3809.86 3714.13
```

## 2.30、/proc/version

当前系统运行的内核版本号，在作者的RHEL5.3上还会显示系统安装的gcc版本，如下所示；

```
[root@rhel5 ~]# more /proc/version
Linux version 2.6.18-128.el5 (mockbuild@hs20-bc1-5.build.redhat.com) (gcc version 4.1.2 20080704 (Red Hat 4.1.2-44)) #1 SMP Wed Dec 17 11:42:39 EST 2008
```

## 2.31、/proc/vmstat

当前系统虚拟内存的多种统计数据，信息量可能会比较大，这因系统而有所不同，可读性较好；下面为作者机器上输出信息的一个片段：(2.6以后的内核支持此文件)

```
[root@rhel5 ~]# more /proc/vmstat
nr_anon_pages 22270
nr_mapped 8542
nr_file_pages 47706
nr_slab 4720
nr_page_table_pages 897
nr_dirty 21
nr_writeback 0
■■■■
```

## 2.32、/proc/zoneinfo

内存区域(zone)的详细信息列表, 信息量较大, 下面列出的是一个输出片段:

```
[root@rhel5 ~]# more /proc/zoneinfo
Node 0, zone          DMA
  pages free        12088
    min             28
    low             35
    high            42
```

```
active 439
inactive 1139
scanned 0 (a: 7 i: 30)
spanned 4096
present 4096
nr_anon_pages 192
nr_mapped 141
nr_file_pages 1385
nr_slab 253
nr_page_table_pages 2
nr_dirty 523
nr_writeback 0
nr_unstable 0
nr_bounce 0
    protection: (0, 0, 296, 296)
pagesets
all_unreclaimable: 0
prev_priority: 12
start_pfn: 0
■■■■
```

三、/proc/sys目录详解

与 /proc下其它文件的“只读”属性不同的是，管理员可对/proc/sys子目录中的许多文件内容进行修改以更改内核的运行特性，事先可以使用“ls -l”命令查看某文件是否“可写入”。写入操作通常使用类似于“echo DATA > /path/to/your/filename”的格式进行。需要注意的是，即使文件可写，其一般也不可以使用编辑器进行编辑。

3.1、/proc/sys/debug 子目录

此目录通常是一空目录；

3.2、/proc/sys/dev 子目录

为系统上特殊设备提供参数信息文件的目录，其不同设备的信息文件分别存储于不同的子目录中，如大多数系统上都会具有的/proc/sys/dev /cdrom和/proc/sys/dev/raid（如果内核编译时开启了支持raid的功能）目录，其内存储的通常是系统上cdrom和raid的相关参数信息文件。

好文要顶

关注我

收藏该文

[zhezhelein](#)  
[关注 - 4](#)  
[粉丝 - 94](#)  
[+加关注](#)

1

0

推荐

反对

(请您对文章做出评价)

« 上一篇:[电脑硬件资源有冲突怎么处理?](#)  
» 下一篇:[设备内存缓冲区和 /proc/iomem](#)

posted on 2011-04-20 14:27 [zhezhelein](#) 阅读(16825) 评论(5) [编辑](#) [收藏](#)

评论

#1楼 2013-03-19 10:03 [zhangkaige](#) \_

学习一下

支持(0) 反对(0)

#2楼 2013-06-03 13:57 [zhengbo2013](#) \_

学习学习

支持(0) 反对(0)

#3楼 2014-12-29 14:02 [lenmon](#) \_

又作作得... 感谢... 感谢...

总结的很好，感谢。可以转载嘛

支持(0) 反对(0)

#4楼 2015-01-22 20:02 吉祥要学习  \_

总的的很好，学习了

支持(0) 反对(0)

#5楼 2015-02-26 14:51 月光xia漫步  \_

通过/proc/进程号目录下的fd目录，可以很容易看到程序正在处理的文件，如下：

```
[root@localhost 17923]# ls -l fd
total 0
lr-x----- 1 root root 64 Feb 26 14:40 0 -> /dev/null
l-wx----- 1 root root 64 Feb 26 14:40 1 -> pipe:[316994287]
l-wx----- 1 root root 64 Feb 26 14:40 2 -> pipe:[316994287]
lrwx----- 1 root root 64 Feb 26 14:40 3 -> socket:[316994715]
l-wx----- 1 root root 64 Feb 26 14:40 4 ->
/home/fwlog_txt/FWLOG_201502261451000668027.tmp
l-wx----- 1 root root 64 Feb 26 14:40 5 ->
/home/fwlog/FWLOG_201502261451010577027.tmp
[root@localhost 17923]#
不错，谢谢分享！
```

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

[【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库](#)

[【推荐】融云即时通讯云—豆果美食、Faceu等亿级APP都在用](#)



ActiveReports  
全方位的报表解决方案

交互式报表

立即下载 

GrapeCity

最新IT新闻：

- [张朝阳：搜狐视频将支持VR内容自媒体](#)
  - [两家又合作了 乐视将独家运营TCL电视的全部开机广告](#)
  - [致创业公司创始人：请不要再拿期权忽悠人了](#)
  - [英国女王5万英镑招新媒体主管，你离出入皇室就差一份简历了！](#)
  - [这7家公司想要利用人工智能技术来颠覆7个不同的行业](#)
- » [更多新闻...](#)



90%的开发者都在用 极光推送

—— 不只是稳定 ——

最新知识库文章：

- [架构漫谈\(九\):理清技术、业务和架构的关系](#)
  - [架构漫谈\(八\):从架构的角度看如何写好代码](#)
  - [架构漫谈\(七\):不要空设架构师这个职位,给他实权](#)
  - [架构漫谈\(六\):软件架构到底是要解决什么问题?](#)
  - [架构漫谈\(五\):什么是软件](#)
- » [更多知识库文章...](#)