实验 23 proc 文件系统

23.1 本章导读

所有的 Linux 文件系统都自带 proc 文件系统，类似于 window 系统的任务管理器，在调试驱动的时候会用到。

学习方法类似前面的“linux 命令”，了解到有这个功能，学习几个基本的命令，然后在后面用到的时候去网上查找其对应用法即可。

proc 文件系统一般是只读，也是可以通过编写代码给 proc 中添加信息，感兴趣可以在网上找一找代码来实现，用处不是很大，系统自带的功能已经够用了。

23.1.1 工具

23.1.1.1 硬件工具

1）iTOP4412 开发板

2）PC 机

3）串口

23.1.1.2 软件工具

1）虚拟机 Vmware

2）Ubuntu12.04.2

3）超级终端（串口助手）

23.1.2 预备课程

无

23.1.3 视频资源

本节配套视频为“视频 23_proc 文件系统”

23.2 学习目标

本章需要学习以下内容：

了解 proc 文件系统

23.3 实验操作

在开发板中输入命令“cat proc/meminfo”，如下图所示。

```
[root@iTOP-4412]# cat proc/meminfo
MemTotal:         953748 kB
MemFree:          939020 kB
Buffers:             476 kB
Cached:             2744 kB
SwapCached:            0 kB
Active:             1560 kB
Inactive:           1988 kB
Active(anon):        332 kB
Inactive(anon):        0 kB
Active(file):       1228 kB
Inactive(file):     1988 kB
Unevictable:           0 kB
Mlocked:               0 kB
HighTotal:        261120 kB
HighFree:         257480 kB
LowTotal:         692628 kB
```

常用参数如下：

MemTotal:总内存

MemFree：空闲内存

Cached：缓存

Active：活跃内存

Inactive：非活跃内存

在开发板中输入命令“cat /proc/cpuinfo”，如下图所示。

```
[root@iTOP-4412]# cat /proc/cpuinfo
cpu id          : 0xe4412011

Processor       : ARMv7 Processor rev 0 (v7l)
processor       : 0
BogoMIPS        : 1992.29

Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x3
CPU part        : 0xc09
CPU revision    : 0

Hardware        : SMDK4X12
Revision        : 0000
Serial          : 0000000000000000
```

常用参数如下：

cpu id：cpu 代号

Processor：处理器

在开发板中输入命令“cat /proc/interrupts”，如下图所示。

```
[root@iTOP-4412]# cat /proc/interrupts
           CPU0
 24:        112  s3c-uart  s5pv210-uart
 26:        640  s3c-uart  s5pv210-uart
 98:          0       GIC  s3c-pl330.0
 99:          0       GIC  s3c-pl330.1
100:          0       GIC  s3c-pl330.2
107:          0       GIC  s3c2410-wdt
108:          0       GIC  s3c2410-rtc alarm
121:          9       GIC  mct_comp_irq
123:      23616       GIC  s3c2440-i2c.1
125:          1       GIC  s3c2440-i2c.3
126:         60       GIC  s3c2440-i2c.4
127:          0       GIC  s3c2440-i2c.5
129:          6       GIC  s3c2440-i2c.7
134:      17526       GIC  ehci_hcd:usb1
135:         41       GIC  s3c-udc
139:          0       GIC  mmc1
140:         52       GIC  mmc2
141:       1477       GIC  mmc0
160:          0       GIC  samsung-rp
281:          0  COMBINER  s3cfb
352:          1  exynos-eint
359:          0  exynos-eint  s3c-sdhci.2
367:          1  exynos-eint  s5m87xx-irq
370:          0  exynos-eint  switch-gpio
428:          0   s5m8767  rtc-alarm0
```

如上图所示，显示的是一些中断相关的参数，如果新注册了中断，都会在这里显示。

23.4 proc 参数介绍

下面是网上介绍的一个相对详细的帖子，我给大家做了简单的编辑，大家可以浏览一下，对其有个整体上的把握即可。

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@rhel5 ~]# 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@rhel5 ~]# 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
-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        /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；

[root@rhel5 ~]# 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@rhel5 ~]# more /proc/devices

Character devices:

1 mem

4 /dev/vc/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@rhel5 ~]# 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@rhel5 ~]# 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@rhel5 ~]# more /proc/filesystems
nodev     sysfs
nodev     rootfs
nodev     proc
          iso9660
          ext3
.....
.....
```

2.12、/proc/interrupts

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@rhel5 ~]# 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@rhel5 ~]# more /proc/iomem

00000000-0009f7ff : System RAM

0009f800-0009ffff : reserved

000a0000-000bffff : Video RAM area

000c0000-000c7fff : Video ROM

.....

2.14、/proc/ioports

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@rhel5 ~]# less /proc/ioports

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@rhel5 ~]# 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/dmsg 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@rhel5 ~]# more /proc/loadavg

0.45 0.12 0.04 4/125 5549

[root@rhel5 ~]# 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 5504 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

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 1208

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。