

Pseudo sistema de ficheiros /proc

José Pedro Oliveira
(jpo@di.uminho.pt)

Grupo de Sistemas Distribuídos
Departamento de Informática
Escola de Engenharia
Universidade do Minho

Sistemas Operativos
2006-2007



Conteúdo

- 1 Pseudo sistema de ficheiros /proc
 - Informação sobre o kernel
 - Processos
 - Configurar kernel em runtime
 - Referências



Descrição

- O pseudo sistema de ficheiros **/proc** actua como interface a estruturas de dados do kernel.
- Pode ser utilizado para obter informação sobre o sistema e alterar certos parâmetros do kernel em *runtime* (sysctl).
- Enquanto que a grande maioria do pseudo sistema de ficheiros /proc só pode ser acedida para leitura, o ramo **/proc/sys** é utilizado para configurar o kernel em *runtime*, ou seja, permite operações de escrita.



Informação

- subdirectórios por processo
- informação sobre o kernel
- dispositivos IDE (/proc/ide)
- informação sobre rede (/proc/net)
- informação SCSI (/proc/scsi)
- informação sobre porta paralela (/proc/parport)
- informação sobre TTY (/proc/tty)
- estatísticas diversas sobre o kernel (/proc/stat)



/proc/{version.cmdline}

Versão do kernel

\$ cat /proc/version

```
Linux version 2.6.20-1.2300.fc5
(brewbuilder@hs20-bc1-5.build.redhat.com) (gcc
version 4.1.1 20070105 (Red Hat 4.1.1-51)) #1 Sun
Mar 11 19:29:11 EDT 2007
```

Linha de comando usada no arranque

\$ cat /proc/cmdline

```
ro root=LABEL=/1 nolapic
```



/proc/cpuinfo

Informação sobre o(s) processador(es)

\$ cat /proc/cpuinfo

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 15
model         : 2
model name    : Intel(R) Pentium(R) 4 CPU 2.40GHz
stepping     : 4
cpu MHz       : 2405.473
cache size   : 512 KB
...
```



/proc/modules

Informação sobre os módulos carregados

\$ cat /proc/modules

```
8139cp 22977 0 - Live 0xe818a000
8139too 27073 0 - Live 0xe8182000
asus_acpi 12372 0 - Live 0xe8276000
autofs4 21701 1 - Live 0xe841d000
battery 10053 0 - Live 0xe82cb000
bluetooth 47909 3 hidp,rfcomm,l2cap, Live 0xe83f0000
button 6993 0 - Live 0xe8270000
cisco_ipsec 549868 0 - Live 0xe8586000
drm 70229 2 i915, Live 0xe843b000
ehci_hcd 32461 0 - Live 0xe8191000
ext3 129609 3 - Live 0xe8101000
fat 51933 1 vfat, Live 0xe8300000
...
```



Processo nnn (/proc/nnn)

\$ ls -l /proc/7218/

```
total 0
dr-xr-xr-x  2 jpo jpo 0 Feb 21 00:07 attr
-r-----  1 jpo jpo 0 Feb 21 00:09 auxv
-r--r--r--  1 jpo jpo 0 Feb 21 00:06 cmdline
lrwxrwxrwx  1 jpo jpo 0 Feb 21 00:09 cwd -> /home/users/jpo/csi
-r-----  1 jpo jpo 0 Feb 21 00:06 environ
lrwxrwxrwx  1 jpo jpo 0 Feb 21 00:09 exe -> /usr/X11R6/bin/gvim
dr-x-----  2 jpo jpo 0 Feb 21 00:09 fd
-r-----  1 jpo jpo 0 Feb 21 00:09 maps
-rw-----  1 jpo jpo 0 Feb 21 00:09 mem
-r--r--r--  1 jpo jpo 0 Feb 21 00:09 mounts
lrwxrwxrwx  1 jpo jpo 0 Feb 21 00:09 root -> /
-r--r--r--  1 jpo jpo 0 Feb 21 00:06 stat
-r--r--r--  1 jpo jpo 0 Feb 21 00:09 statm
-r--r--r--  1 jpo jpo 0 Feb 21 00:06 status
dr-xr-xr-x  3 jpo jpo 0 Feb 21 00:09 task
-r--r--r--  1 jpo jpo 0 Feb 21 00:09 wchan
```



Processo nnn (/proc/nnn)

Informação sobre a linha de comando e as variáveis de ambiente

```
$ cat /proc/7218/cmdline | tr '\0' '\n'
```

```
gvim
processos.tex
```

```
$ cat /proc/7218/environ | tr '\0' '\n'
```

```
KDE_MULTIPLEHEAD=false
HOSTNAME=localhost.localdomain
SHELL=/bin/bash
TERM=xterm
...
```

Configurar o kernel em *runtime*: /proc/sys

Introdução

O directório /proc/sys não só é uma fonte de informação, como permite alterar em *runtime* parâmetros do kernel.

Alterar parâmetros do kernel

Para alterar um valor basta apenas fazer `echo` do novo valor para o ficheiro.



Processo nnn (/proc/nnn)

Informação sobre descritores de ficheiros abertos

```
$ ls -l /proc/7218/fd
```

```
total 6
lrwx----- 1 jpo jpo 64 Feb 21 00:18 0 -> /dev/pts/5
lrwx----- 1 jpo jpo 64 Feb 21 00:18 1 -> /dev/pts/5
lrwx----- 1 jpo jpo 64 Feb 21 00:18 2 -> /dev/pts/5
lrwx----- 1 jpo jpo 64 Feb 21 00:18 3 -> socket:[17997]
lrwx----- 1 jpo jpo 64 Feb 21 00:18 4 -> socket:[17999]
lrwx----- 1 jpo jpo 64 Feb 21 00:18 6 -> /tmp/.processos.tex.swap
```



Alterar um parâmetro do kernel

Exemplo

Alterar o comportamento da *stack* TCP/IP na recepção de pacotes ICMP broadcast.

Executar os seguintes comandos

- \$ cd /proc/sys/net/ipv4
- \$ cat icmp_echo_ignore_broadcasts


```
0
```
- \$ echo 1 > icmp_echo_ignore_broadcasts
- \$ cat icmp_echo_ignore_broadcasts


```
1
```



Alterar um parâmetro do kernel: comando sysctl

Comando `sysctl`

Permite configurar parâmetros em *runtime*. Os parâmetros disponíveis são os que se encontram listados em `/proc/sys`.

Synopsis

```
sysctl [opções] variável ...
sysctl [opções] -w variável=valor ...
...
```

Exemplos

- `$ sysctl -n kernel.hostname`
- `$ sysctl -w kernel.hostname=posto123`



Referências

Referências

- `man 8 sysctl`
- `man 5 sysctl.conf`
- `<kernel>/Documentation/filesystems/proc.txt`
Exemplos:
`linux-2.4.34/Documentation/filesystems/proc.txt`
`linux-2.6.20/Documentation/filesystems/proc.txt`

Ficheiro de configuração `/etc/sysctl.conf``$ cat /etc/sysctl.conf`

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0
...
```

