Uzyskiwanie informacji systemowych

Jacek Łuczak

9 marca 2005

1 Wstęp

Jądro systemu Linux dostarcza specjalny mechanizm dostępu do informacji systemowych. Jest to pseudosystem plików `/proc`. Pliki znajdujące się w nim są głównym źródłem informacji systemowych. Większość z tych plików to pliki tekstowe. Kod jądra implementujący pseudosystem plików `/proc` znajduje się w katalogu `fs/proc` źródła jądra.

2 Informacje o procesach

W `/proc` można znaleźć osobny katalog dla każdego działającego procesu. Nazwą takiego katalogu jest PID (Process ID – numer identyfikujący każdy proces) procesu. W każdym z tych katalogów znajduje się wiele różnych plików, których znaczenie omawiam niżej. Wystąpienie w tekście frazy $PID oznacza PID jakiegokolwiek interesującego nas procesu. Plik `fs/proc/array.c` zawiera większość kodu odpowiedzialnego za generowanie (w `/proc`) wpisów dotyczących procesów.

2.1 Plik `cmdline`

Plik ten zawiera kompletną linię poleceń, z którą został wywołany proces, tj. nazwę programu i listę wszystkich argumentów przekazanych podczas uruchamiania. Jeżeli proces jest procesem zombie, to plik ten powinien być pusty. Jest on niezwykle przydatny. Możemy na przykład wyświetlić wszystkie procesy wpisując:

```
strings -f /proc/[0-9]*/cmdline
```

Opcja `-f` powoduje, że `strings` będzie wyświetlał nazwy plików przed ich zawartością.

2.2 Plik `environ`

W pliku tym możemy znaleźć wszystkie zmienne danego procesu. Do ich wyświetlenia można użyć również komendy `strings`. Jeśli tak zrobimy, każda zmienna będzie wyświetlona w nowej linii.

2.3 Katalog `fd`

Katalog ten zawiera deskryptory otwartych plików jako dowiązania symboliczne do aktualnego inoda (i-węzła) pliku. I-węzeł jest to specjalna struktura zawiera-

jąca informacje o pliku. Istnieje możliwość interakcji z procesem poprzez otwarcie jego pliku deskryptora. Można również użyć wywołań systemowych `fstat()` i `lstat()` aby uzyskać informacje o danym pliku. Prawa dostępu dla właściciela reprezentują tryb w jakim plik został otwarty.

2.4 Plik `mem`

Plik `/proc/$PID/mem` daje dostęp do pamięci procesu. Można się do niego odwołać używając wywołania systemowego `mmap()`.

2.5 Plik `stat`

W pliku tym możemy znaleźć większość informacji wyświetlanych przez takie programy jak na przykład `ps`. W tabeli na następnej stronie umieszczone są objaśnienia pól łącznie z inch nazwami używanymi przez `ps`.

2.6 Plik `status`

Plik ten zawiera podobne informacji jak plik `stat`. Jest ich mniej, ale są one przedstawione w bardziej czytelnej formie.

2.7 Dowiązanie symboliczne `cwd`

Dowiązanie to wskazuje na bieżący katalog procesu.

2.8 Dowiązanie symboliczne `exe`

Wskazuje ono na plik wykonywalny - plik binarny. Może również prowadzić do skryptu lub pliku interpretera skryptu.

2.9 Dowiązanie symboliczne `root`

Dowiązanie symboliczne `root` prowadzi do głównego katalogu procesu ustawianego za pomocą wywołania systemowego `chroot()`.

2.10 Plik `statm`

Plik ten zawiera statystyki wykorzystania pamięci. W kolejności mamy: całkowity wykorzystywany rozmiar (kod, dane i stos), rozmiar części rezydentalnej, strony wspólne, kod, dane/stos, biblioteki, strony nieaktualne. Program `top` wykorzystuje te pliki do wyświetlania informacji o używanej przez procesy pamięci.

L.p.	Nazwa	Opis
1	pid	ID procesu
2	cmd	Nazwa programu używana przez `ps` jesli plik `cmdline` jest pusty
3	state	Reprezentuje stan procesu (R, S, D, T, Z, W, N)
4	ppid	ID procesu rodzica
5	pgrp	ID grupy procesu
6	session	ID sesji
7	tty	Minorowy numer urządzenia tty
8	tpgid	ID grupy procesu kontrolującego
9	flags	Flagi procesu (jak w polu F długiego listingu - `ps axl`)
10	min_flt	Ilość mniejszych błędów stron
11	cmin_flt	Kumulujące mniejsze błędy stron
12	maj_flt	Ilość większych błędów stron
13	cmaj_flt	Kumulujące większe błędy stron
14	utime	Czas poniżejużytkownika
15	stime	Czas systemu
16	cutime	Kumulujący czas użytkownika
17	cstime	Kumulujący czas systemowy
18	counter	Rozmiar kwantu czasu procesu
19	priority	Priorytet w shedulerze jądra
20	nice	Wartość nice - dynamiczny priorytet procesu w shedulerze
21	tiomeout	Wartość upływu czasu w jiffies (1/100 s)
22	it_real_value	Interwał upływu czsu w jiffies
23	start_time	Kiedy proces został uruchomiony
24	vsize	Całkowity rozmiar pamięci VM w bajtach
25	rss	Resident Set Size; kilobajty programu w pamięci
26	rss_rlimit	RSS rlimit
27	start_code	Początek segmentu kodu
28	end_code	Koniec segmentu kodu
29	start_stack	Początek segmentu stosu
30	kstk_esp	Aktualna ramka stosu
31	kstk_eip	Aktualna ramka stosu
32	signal	Obsługiwane sygnały
33	blocked	Zablokowane sygnały
34	sigignore	Ignorowane sygnały
35	sigcatch	Przechwytywane sygnały - sygnały z zarejestrowaną obsługą

3 Główne informacje systemowe

W katalogu `/proc` istnieje wiele plików i kataolgów, które nie odnoszą się konkretnie do procesów. Program `procinfo` korzysta właśnie z tych plików. W dalszej części omówię najważniejsze z nich. Podane tu informacje mogą się różnić od tych w twoim systemie, dlatego, że jądro jest cały czas dynamicznie rozwijane i wiele rzeczy mogło w międzyczasie uledz zmianie.

3.1 Plik `/proc/cmdline`

W pliku tym przechowywane są parametry podane do jądra podczas bootowania.

3.2 Plik `/proc/cpuinfo`

W pliku tym znajdują się informacje na temat procesora (procesorów). Można tu znaleźć model procesora, wersję, prędkość w MHz, bogomips (stała systemowa wyliczona przy starcie systemu) i wiele innych użytecznych informacji.

3.3 Plik `/proc/devices`

Zawiera numery grup głównych, grup urządzeń oraz nazwę zarejestrowaną przez dane urządzenie.

3.4 Plik `/proc/dma`

W pliku znajduje się lista zarezerwowanych i używanych przez sterowniki kanałów DMA (Direct Memory Access).

3.5 Plik `/proc/filesystems`

Można tu znaleźć listę wkompilowanych w jądro systemów plików. Informacje z tego pliku są wykorzystywane przez `mount`.

3.6 Plik `/proc/interrupts`

Plik ten zawiera po jednej linii dla zarezerwowanego IRQ. Kolumny oznaczają kolejno: numer przerwania, liczbę otrzymanych przerwań dla danego IRQ, specjalną flagę oraz nazwę sterownika służącego do obsługi danego przerwania.

3.7 Plik /proc/ioports

Moña tu znaleźć listę obecnie zarejestrowanych i używanych obszarów portów I/O (wejścia/wyjścia).

3.8 Plik /proc/kcore

Jest to bardzo ważny i użyteczny plik. Reprezentuje on pamięć fizyczną w formacie pliku `core`. Wykorzystuje się go do testowania stanu dowolnej struktury jądra przy użyciu GDB. Całkowity rozmiar tego pliku powinien być równy ilości posiadanej pamięci plus 4KB.

3.9 Plik /proc/kmsg

Czytać z tego pliku może tylko jeden proces posiadający uprawinienia roota. Służy on do uzyskania dostępu do wiadomości generowanych przez jądro przy użyciu `printk()`. Do tego celu należy używać wywołania systemowego `syslog()` (nie należy mylić z funkcją biblioteczną `syslog()`). Narzędzia takie jak `dmesg` lub demon `klogd` używają tego pliku.

3.10 Plik /proc/ksyms

W jądrach serii 2.6.x plik ten nazywa się `kallsyms`. W pliku znajduje się lista zarejestrowanych w jądrze definicji symboli (adresów zmiennych lub funkcji). Pierwsze pole, to adres, drugie nazwa, trzecie - nazwa modułu, który zarejestrował dany symbol.

3.11 Plik /proc/loadavg

Plik zawiera liczby średniego obciążenia. Są to uśrednione w ciągu 1, 5 i 15 minut liczby zadań w kolejce do wykonania. Ostanie pole w pliku to PID ostatnio uruchomionego procesu. Komenda `uptime` służy do wyświetlania tych trzech wartości (i nie tylko).

3.12 Plik /proc/locks

Plik ten pokazuje aktulne blokady nałożone na pliki. Jest on generowany przez funkcję `get_locks_status()` (kod tej funkcji można znaleźć w `fs/locks.c`). Do nakładania blokad służą funkcje `fcntl()` i `flock()`.

3.13 Plik /proc/meminfo

Plik ten dostarcza informacji o rozmiarze wolnej i zajętej pamięci (zarówno fizycznej, jak i wymiany), a także o pamięci wspólnej i buforach używanych przez jądro. Jest on wykorzystywany przez `free`.

3.14 Plik /proc/misc

Przechowuje nazwy sterowników zarejestrowanych przy użyciu funkcji jądra `misc_register()`.

3.15 Plik /proc/modules

Plik ten zawiera informacje o załadowanych modułach. Są to: nazwa, zajmowany rozmiar, użycie oraz powiązania z innymi modułami. Informacje te wykorzystywane są przez `lsmod`.

3.16 Plik /proc/mounts

W pliku można znaleźć informacje o aktualnie zamontowanych systemach plików w formacie takim samym jak ten w pliku `/etc/mtab`.

3.17 Plik /proc/pci

Znajduje się tu listing i konfiguracja wszystkich urządzeń PCI znalezionych podczas inicjalizacji jądra. W jądrach serii 2.6.x plik ten jest już nieużywany. Posiadacze jąder tej serii mogą uzyskać informacje o urządzeniach PCI używając komendy `lspci`. Mimo to istnieje możliwość powrotu do starego modelu poprzez ponowną kompilację jądra z zaznaczoną opcją `Legacy /proc/pci interface`.

3.18 Plik /proc/stat

Plik zawiera statystyki jądra i systemu. Do najważniejszych można zaliczyć:

1. `cpu` Liczba jiffies, które system spędził w trybie użytkownika, trybie o obniżonym priorytecie, trybie systemowym i w zadaniu idle. Ostatnia wartość powinna być równa stukrotności drugiego wpisu w pliku `uptime`.
2. `page` Liczba stron, które system wstronicował i wystronicował (z dysku).
3. `intr` Liczba przerwań otrzymanych od uruchomienia systemu.
4. `ctct` Liczba przełączeń kontekstu.

3.19 Plik `/proc/uptime`

W pliku tym znajdują się zawsze dwie wartości wyrażone w sekundach. Pierwsza z nich to czas pracy systemu, druga - ilość czasu spędzonego na wykonaniu procesu idle.

3.20 Plik `/proc/version`

Plik przechowuje wersję aktualnie działającego jądra. Jest on wykorzystywany przez `uname`. Oto co zawiera ten plik na moim komputerze:

```
cat /proc/version
Linux version 2.6.10-ac10 (root@slawek) (gcc version
3.3.5) #5 SMP Sun Mar 6 20:51:13 CET 2005
```

4 Podsumowanie

Wymienione tutaj pliki i katalogi to tylko mała część tego co można znaleźć w katalogu `/proc`. Inne podkatalogi tego pseudosystemu plików mogą być ogromnym źródłem informacji. Są to m.in. katalogi `/proc/sys`, `/proc/net`, `/proc/bus`, itd. Więcej informacji można znaleźć na stronie manuala (`man 5 proc`) lub bezpośrednio w plikach źródłowych jądra. Zwłaszcza ta druga metoda zdobywania informacji o katalogu `/proc` jest niezwykle pouczająca. Zachęcam do własnych poszukiwań:-).