Источники информации о системе

Операционная система рассказывает нам о себе при помощи особенной файловой системы под названием procfs.

Директория /proc довольно странная. Она не существует на самом деле, но вы можете заглянуть в неё. Её файлы нулевой длины не являются ни бинарными, ни текстовыми, но вы можете открыть и просмотреть их. Эта специальная директория хранит все детали о вашей системе Linux, включая её ядро, процессы и параметры конфигурации.

Директория /proc содержит виртуальные файлы. Их можно вывести списком, но на самом деле они не существуют на диске; операционная система создаёт их «на лету», если вы делаете попытку их прочитать.

Большинство виртуальных файлов всегда имеют текущую метку даты/времени, говорящую о том, что они постоянно поддерживаются в рабочем состоянии.

Директория /proc создаётся сама по себе каждый раз при загрузке системы. Вам нужно работать с правами суперпользователя, чтобы просмотреть всю директорию.

Существует специальный подкаталог /proc/sys:

```
ls /proc/sys
```

Он позволяет изменять некоторые параметры ядра в реальном режиме времени и отображать их.

Директория /proc состоит из виртуальных директорий и поддиректорий, которые группируют файлы по определённому принципу. Давайте посмотрим, что вернёт нам команда:

```
ls /proc
```

Некоторые виртуальные файлы предоставляют информацию об аппаратном обеспечении, например /proc/cpuinfo, /proc/meminfo и /proc/interrupts. Другие дают информацию, связанную с файлами, например /proc/filesystems или /proc/partitions. Файлы в /proc/sys относятся к конфигурации параметров ядра.

Например, если вы хотите узнать, какую версию ядра вы используете, можете попробовать uname -srv или набрать cat /proc/version. Есть ещё несколько других интересных файлов.

- /proc/apm содержит информацию о Advanced Power Management, если она установлена.
- /proc/acpi — похожая директория, дающая хорошую информацию о более современном ACPI. К примеру, чтобы узнать, подключён ли ваш ноутбук к

питанию AC, вы можете выполнить cat /proc/acpi/ac_adapter/AC/state и получить либо on line, либо off line.

- /proc/cmdline показывает параметры, переданные ядру при загрузке.
- /proc/cpuinfo даёт данные о процессоре вашего компьютера. Например, на моём ноутбуке команда cat /proc/cpuinfo выводит листинг, начинающийся с:

```
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 6
model           : 8
model name      : Mobile AMD Athlon(tm) XP 2200+
stepping        : 1
cpu MHz         : 927.549
cache size      : 256 KB
```

- /proc/loadavg — файл, показывающий среднюю загрузку процессора.
- /proc/stat также предоставляет статистику, но последней загрузки.
- /proc/uptime — короткий файл, в котором только два числа: сколько секунд ваша система работает и сколько секунд она простаивает.
- /proc/devices показывает все настроенные и загруженные символьные и блочные устройства. /proc/ide и /proc/scsi дают данные об устройствах IDE и SCSI.
- /proc/ioports показывает информацию о зонах, используемых для ввода-вывода с вышеуказанными устройствами.
- /proc/dma показывает используемые каналы прямого доступа к памяти (DMA).
- /proc/filesystems показывает типы файловых систем, поддерживаемых вашим ядром. Вот как это может выглядеть:

```
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   cpuset
...несколько строк пропущено...
nodev   ramfs
nodev   hugetlbfs
nodev   mqueue
       ext3
nodev   usbfs
       ext2
nodev   autofs
```

Первая колонка показывает, какая файловая система монтирована на блочное устройство.

- /proc/mounts показывает всё смонтированное, что используется вашим компьютером (его вывод очень похож на /etc/mtab).

- /proc/partitions и /proc/swaps покажет все разделы и пространство swap.
- /proc/fs: если вы открываете общий доступ к файловым системам с помощью NFS, в этой директории среди множества поддиректорий и файлов есть /proc/fs/nfsd/exports, который показывает файловую систему, открытую для общего доступа и права на неё.
- /proc/net содержит сетевую информацию. Описание каждого из файлов в этой директории займёт слишком много места, но в ней есть /dev (каждое сетевое устройство), несколько файлов, связанных с iptables, сетевая статистика и статистика портов, информация о беспроводных подключениях и так далее.

Есть также несколько файлов, связанных с ОЗУ, например /proc/iomem, который показывает, сколько памяти использует ваша система, и /proc/kcore, демонстрирующий физическую оперативную память вашей системы. В отличие от большинства других виртуальных файлов размер /proc/kcore соответствует размеру вашей оперативной памяти и даже немного больше.

Здесь же есть много файлов и директорий, связанных с аппаратным обеспечением, такие как /proc/interrupts и /proc/irq, /proc/pci (все устройства PCI), /proc/bus и так далее, но они включают в себя очень специфическую информацию, которая не нужна большинству пользователей.

Процессы

Директории, имена которых пронумерованы, представляют все запущенные процессы. Когда процесс завершается, его директория в /proc исчезает автоматически. Если вы проверите любую из этих директорий во время их существования, то обнаружите множество файлов, таких как:

```
ls /proc/pid/
```

Давайте изучим основные файлы.

- cmdline содержит команду, запустившую процесс, со всеми её параметрами.
- cwd — символьная ссылка на текущую рабочую директорию (CWD) процесса; exe ссылается на исполняемый процесс, а root — на его корневую директорию.
- environ показывает все переменные окружения процесса.
- fd содержит все файловые дескрипторы для процесса, показывающие, какие файлы или устройства он использует.

Если мы выведем список каталога fd для нашего процесса PID, мы получим следующее:

```
ls -l fd
```

Файловый дескриптор — это число. Оно указывает на какой-то файл, который использует процесс. Они по умолчанию создаются для каждого процесса. Это потоки стандартного ввода, вывода и ошибок.

- maps, statm и mem работают с памятью, используемой процессом.
- stat и status предоставляют информацию о статусе процесса, но последний — более чёткую и упорядоченную.

Настройка системы: /proc/sys

/proc/sys не только даёт информацию о системе, но и позволяет изменять параметры ядра «на лету», активирует и дезактивирует его возможности.

Чтобы определить, какой файл для настройки, а какой только для чтения, выполните:

```
ls -ld
```

Если файл имеет атрибут W, это значит, что вы можете использовать его каким-либо образом для настройки ядра. К примеру, ls -ld /proc/kernel/* начинается примерно так:

```
dr-xr-xr-x 0 root root 0 2008-01-26 00:49 pty
dr-xr-xr-x 0 root root 0 2008-01-26 00:49 random
-rw-r--r-- 1 root root 0 2008-01-26 00:49 acct
-rw-r--r-- 1 root root 0 2008-01-26 00:49 acpi_video_flags
-rw-r--r-- 1 root root 0 2008-01-26 00:49 audit_argv_kb
-r--r--r-- 1 root root 0 2008-01-26 00:49 bootloader_type
-rw------- 1 root root 0 2008-01-26 00:49 cad_pid
-rw------- 1 root root 0 2008-01-26 00:49 cap-bound
```

Как видите, bootloader_type не может быть изменён, но остальные файлы могут. Чтобы изменить файл, используйте echo 10 >/proc/sys/vm/swappiness.

Используйте sysctl и файл /etc/sysctl.conf, чтобы сделать более постоянные изменения.
Давайте поверхностно посмотрим на директории в /proc/sys.

- debug содержит отладочную информацию. Это полезно, если вы занимаетесь разработкой ядра.
- dev предоставляет информацию о специфических устройствах в вашей системе. Например, проверьте директорию /dev/cdrom.
- fs даёт данные о каждом возможном аспекте файловой системы.
- kernel позволяет вам затрагивать работу и настройки ядра напрямую.
- net допускает вас до вопросов, связанных с сетью. Будьте осторожны, потому что, запутавшись в этом, вы можете потерять подключение.
- vm работает с подсистемой VM.

Именно из procfs берут информацию все диагностические утилиты, такие как:

```
top
atop
iotop
iftop
sar
vmstat
```

Лог-файлы

Лог-файлы — это способ сохранения журнала событий.
Если мы вспомним стандарт расположения различных типов файлов в файловой системе, логи нужно искать в /var/log.
Мы сразу можем увидеть логи наших прикладных программ, например nginx и mysql.

Кроме этих файлов, у нас довольно много системных логов, например syslog или messages. Последнего чаще всего нет, и вместо него присутствует только syslog. Это традиционные глобальные системные журналы операционной системы Linux. Сюда пишутся события загрузки, ядра системы, системы инициализации systemd и так далее.

- auth.log — лог авторизации и аутентификации в системе.
- dmesg — в этом логе хранится информация о загрузке ядра и драйверов оборудования.
- alternatives.log — лог-файл программы update-alternatives. Непонятно, почему ей выделили отдельный лог-файл, а cron, к примеру, нет.
- kern.log — лог сообщений ядра Ubuntu, да и любой другой Linux-системы.
- maillog — сообщения почтовой системы. Обычно postfix или exim. Если на сервере Ubuntu они не установлены, то и почтового лога не будет.
- dpkg.log — логирование работы пакетных менеджеров Ubuntu. Обычно это apt или apt-get.
- lastlog и wtmp — информация о прошлых авторизациях пользователей.

Посмотреть лог загрузки Ubuntu можно следующим образом:

```
sudo dmesg
sudo dmesg | less
```

Если нужно узнать информацию только о диске:

```
sudo dmesg | grep sda
```

Для того чтобы узнать, кто и когда проходил авторизацию на сервере Ubuntu, можно воспользоваться логами из файла /var/log/auth.log.

Рассмотрим теперь вопрос с расположением лога ошибок в Ubuntu. Как такового отдельного error log в традиционных Linux-системах нет. Обычно используют следующие фразы:

```
error или err
critical или crit
debug
war
```

```
ls /var/log
```

Программа logrotate запускается по cron и в конфигурационных файлах в директории /etc/logrotate описывает, что мы делаем с лог-файлами (архивируем ли, когда удаляем, сколько файлов храним и так далее).