

Семинар #8: Процессы

Программа ps

Команда	Действие
ps	Показывает процессы текущего терминала.
ps -e	Показать все процессы в системе.
ps -f	Показывает расширенную информацию: uid, pid, ppid, c, stime, tty, time, cmd.
ps -ef	Показать все процессы в расширенном формате.
ps -u alice,bob	Показать процессы пользователей alice и bob.
ps -p 100,200	Показать процессы, со значениями PID равными 100 или 200.
ps -o pid,user,cmd	Вывод в пользовательском формате. Для каждого процесса покажет идентификатор процесса (pid), имя пользователя (user) и команду которой был запущен процесс (cmd).
ps -o pid=,user=,cmd=	То же, но не выведет первую строку с описанием столбцов.
ps -C top	Все процессы с именем программы top .
ps --sort time	Отсортировать по ключевому слову time .
ps --sort -time	Сортировка в обратном порядке по ключевому слову time .
ps aux	Все процессы с кл. словами: user, pid, pcpu, pmem, vsz, rss, tty, stat, start, time, cmd

Распространённые ключевые слова для ps -o и ps --sort

Ключевое слово	Расшифровка
pid	Идентификатор процесса (Process ID).
ppid	Идентификатор родительского процесса (Parent Process ID).
user	Имя пользователя, запустившего процесс.
uid	Идентификатор пользователя.
comm	Имя команды (без аргументов).
cmd	Полная командная строка (включая аргументы).
tty	Управляющий терминал.
time	Совокупное время ЦП (CPU time).
etime	Прошедшее время с момента запуска процесса (Elapsed time).
c	Использование ЦП (в минутах).
pcpu	Процент использования ЦП.
pmem	Процент использования физической памяти.
vsz	Размер виртуальной памяти в КиБ (Virtual Size).
rss	Размер резидентного набора в КиБ (Resident Set Size).
stat	Текущее состояние процесса (например, R - выполняется, S - спит).
ni	Значение nice (приоритет).
priority	Приоритет ядра.
start	Время или дата запуска процесса.

Больше опций и ключевых слов можно найти в `man ps`.

Сигналы

Распространённые сигналы

Номер	Название	Действие по умолчанию	Примечание
1	SIGHUP	Завершение	Посылается оболочке и процессам в <code>jobs</code> при закрытии терминала.
2	SIGINT	Завершение	Посылается передним процессам, при нажатии <code>Ctrl+C</code> . Используется для корректного завершения процесса.
3	SIGQUIT	Создание дампа ядра и завершение	Посылается, когда пользователь нажимает <code>Ctrl+\</code> . Корректно завершает процесс. Дополнительно создает файл дампа памяти, полезный для отладки
9	SIGKILL	Завершение без возможности перехвата	Принудительное завершение, не дающее программе возможности сохранить данные или корректно освободить ресурсы. Не может быть перехвачен процессом.
15	SIGTERM	Завершение	Стандартный запрос на корректное завершение процесса. Используется по умолчанию, если не указать сигнал в <code>kill</code>
20	SIGTSTP	Приостановка	Посылается, когда пользователь нажимает <code>Ctrl+Z</code> . Приостанавливает процесс, который затем может быть возобновлен командами <code>fg</code> или <code>bg</code> .
19	SIGSTOP	Приостановка без возможности перехвата	Принудительно приостанавливает выполнение процесса. Не может быть проигнорирован или перехвачен процессом
18	SIGCONT	Возобновление	Возобновляет выполнение процесса, который был остановлен сигналами <code>SIGSTOP</code> или <code>SIGTSTP</code> . Используются командами <code>fg</code> и <code>bg</code> для возобновления работы процесса.

Разновидности команды `trap`

- `trap 'команда' сигнал`

Перехватывает сигнал и выполняет команду. Например, если мы напишем в скрипте:

```
trap 'echo Hello' TERM
```

то, если процессу этого скрипта послать сигнал `SIGTERM`, вместо завершения будет выполнена команда `echo Hello`. Данное поведение не наследуется дочерними процессами. Если послать сигнал дочернему процессу, то он завершится, даже если у родительского процесса переопределён обработчик сигнала.

- `trap '' сигнал`

Блокирует поступающий сигнал. Например, если мы напишем в скрипте:

```
trap '' TERM
```

то, если процессу этого скрипта послать сигнал `SIGTERM`, он не будет реагировать на данный сигнал. Данное поведение наследуется дочерними процессами. Если послать сигнал дочернему процессу, то он не завершится, потому что он унаследует это блокирование сигнала у родительского процесса.

- `trap 'команда' EXIT`

Выполняет команду при выходе из скрипта. Команда выполнится при любом завершении скрипта кроме:

- Завершении программы с помощью сигнала `SIGKILL`.
- Аварийном завершении программы.

- `trap 'команда' ERR`

Выполняет указанную команду при возникновении ошибки в любой команде скрипта. То есть если какая-нибудь команда в скрипте завершится с ненулевым кодом возврата. За исключением команд, используемый в `if`, `while` и операторах `&&` и `||`.

Основные файлы в /proc/[pid]

Файл	Содержимое
<code>cmdline</code>	Команда и аргументы, использованные для запуска процесса (разделены нулевыми байтами).
<code>status</code>	Обзор состояния процесса: состояние, PID, PPID, UID/GID, потребление памяти и другое.
<code>stat</code>	Однострочный набор числовых данных о процессе, предназначенный для парсинга.
<code>environ</code>	Переменные окружения процесса (разделены нулевыми байтами).
<code>limits</code>	Текущие ограничения ресурсов, установленные для процесса.
<code>maps</code>	Карты памяти процесса, показывающие, что отображено в адресное пространство процесса.
<code>smaps</code>	Усовершенствованная версия <code>maps</code> , предоставляющая более детальную информацию.
<code>exe</code>	Символическая ссылка на оригинальный исполняемый файл программы.
<code>cwd</code>	Символическая ссылка на текущий рабочий каталог процесса.
<code>fd</code>	Поддиректория с символическими ссылками на открытые файловые дескрипторы процесса.
<code>io</code>	Статистика ввода/вывода процесса (количество байт, считанных и записанных).
<code>statm</code>	Краткая статистика использования памяти в страницах.
<code>cgroup</code>	Информация о группе контроля (cgroups), к которой принадлежит процесс.
<code>net</code>	Поддиректория с информацией о сетевых соединениях процесса
<code>oom_score</code>	Оценка для OOM killer. Чем выше, тем вероятнее процесс будет убит при нехватке памяти.
<code>pagemap</code>	Предоставляет информацию о том, как виртуальные страницы отображаются на физические.
<code>fdinfo</code>	Подробная информация о каждом открытом файловом дескрипторе.
<code>syscall</code>	Номер и аргументы последнего системного вызова, выполненного процессом.

Директория /proc. Информация о (почти) каждом файле.

Файлы в /proc

Файл	Содержимое
cmdline	Аргументы командной строки, переданные ядру при загрузке.
version	Версия ядра Linux, компилятор и дата сборки.
kallsyms	Таблица символов ядра.
kmsg	Буфер сообщений ядра (то, что выводит команда dmesg).
modules	Список загруженных в ядро модулей (драйверов).
loadavg	Средняя загрузка системы (load average) за 1, 5 и 15 минут, а также информация о запущенных/общем количестве процессов.
stat	Общая статистика системы
uptime	Время работы системы (uptime) и время простоя (idle) в секундах.
filesystems	Список файловых систем, которые в данный момент поддерживаются ядром.
sys	Директория, содержащая параметры ядра, которые можно просматривать и изменять "на лету".
sysvipc	Информация о механизмах межпроцессного взаимодействия.
keys	Информация о ключах аутентификации, используемых ядром.
cpuinfo	Содержит подробную информацию о каждом процессоре.
schedstat	Статистика планировщика задач (scheduler) для каждого CPU.
meminfo	Показывает общую, свободную, доступную и используемую память, а также информацию о буферах, кэше, страницах и подкачке.
vmstat	Статистика виртуальной памяти
swaps	Информация об активных разделах или файлах подкачки.
buddyinfo	Информация о фрагментации памяти.
slabinfo	Информация о кэшах SLAB/SLUB.
zoneinfo	Подробная информация о зонах памяти
kpage*	Низкоуровневая информация о страницах физической памяти.
pressure	Показывает, насколько система замедляется из-за нехватки ресурсов.
vmallocinfo	Информация об областях памяти, выделенных с помощью vmalloc
mtrr	Конфигурация того, как CPU кэширует различные области памяти.
devices	Список зарегистрированных символьных и блочных устройств
interrupts	Статистика по используемым линиям прерываний.
irq	Директория для управления прерываниями
ioports	Зарегистрированные порты ввода-вывода
iomem	Карта физической памяти (I/O memory map), показывающая, какие диапазоны адресов используются какими устройствами
dma	Используемые каналы прямого доступа к памяти.
acpi	Информация об управлении питанием, батарее, температуре.
asound	Директория для звуковой подсистемы.
bus	Директория с информацией о шинах.
driver	Директория с информацией о драйверах.
fb	Список доступных фреймбуферов (устройства вывода видео).
misc	Различные драйверы и устройства.
tty	Директория с информацией о терминалах.

consoles	Список зарегистрированных консолей.
timer_list	Список всех ожидающих таймеров в ядре.
softirqs	Статистика по "мягким" прерываниям
diskstats	Статистика ввода-вывода для всех блочных устройств.
partitions	Таблица разделов дисков, известная ядру.
mounts	Список всех смонтированных файловых систем.
locks	Файлы, заблокированные ядром
fs	Директория с информацией о состоянии файловых систем (квоты, inode).
net/tcp	Список активных TCP-соединений.
net/udp	Список активных UDP-сокетов.
net/dev	Статистика по сетевым интерфейсам (принятые/отправленные пакеты, ошибки).
net/arp	ARP-таблица ядра.
net/route	Таблица маршрутизации ядра.
self	Символическая ссылка на директорию процесса, который в данный момент читает этот файл.
thread-self	Аналогично self, но указывает на директорию конкретного потока
kcore	Виртуальный файл, представляющий всю физическую память системы в формате ELF core.
cgroups	Информация о контрольных группах, используемых для ограничения ресурсов

Файлы в /proc/[pid]

Файл	Содержимое
cmdline	Команда и аргументы, использованные для запуска процесса (разделены нулевыми байтами).
environ	Переменные окружения процесса (разделены нулевыми байтами).
status	Обзор состояния процесса: состояние, PID, PPID, UID/GID, потребление памяти и другое.
stat	Одноточный набор числовых данных о процессе, предназначенный для парсинга.
statm	Краткая информация об использовании памяти (в страницах).
auxv	Информация, переданная ядром в пространство пользователя при запуске.
loginuid	ID пользователя, который вошел в систему, инициировавший сессию этого процесса.
sessionid	ID сессии процесса
arch_status	Информация о состоянии, специфичная для архитектуры процессора.
personality	Домен выполнения.
cwd	Ссылка на текущую рабочую директорию процесса.
exe	Ссылка на исполняемый файл, который был запущен.
root	Ссылка на корневую директорию (/) с точки зрения процесса. Для процессов в chroot или контейнерах она будет отличаться от реального корня.
fd	Директория, содержащая файловые дескрипторы, открытые процессом.
fdinfo	Директория с дополнительной информацией о каждом файловом дескрипторе.
maps	Карта памяти процесса. Показывает, какие области памяти куда отображены.
smaps	Подробная карта памяти.
smaps_rollup	Суммарная информация из smaps .
pagemap	Низкоуровневая информация о том, как виртуальные страницы памяти процесса отображаются на физическую память.
mem	Файл, представляющий всю виртуальную память процесса.

clear_refs	Файл для сброса счетчиков "обращения" к страницам памяти.
numa_maps	Информация о политике использования памяти на NUMA-системах.
coredump_filter	Битовая маска, определяющая, какие части памяти процесса будут включены в coredump.
ksm_*	Статистика по KSM (Kernel Same-page Merging).
task	Содержит поддиректории для каждого потока внутри этого процесса.
sched	Информация о политике планировщика для этого процесса
schedstat	Детальная статистика планировщика.
stack	Текущий стек вызовов ядра для этого процесса/потока.
wchan	Если процесс спит (Sleeping), этот файл показывает, в какой функции ядра он "ждет".
syscall	Информация о системном вызове, который в данный момент выполняется процессом.
timerslack_ns	"Допуск" для срабатывания таймеров в наносекундах.
timens_offsets	Смещения времени для временных пространств имен.
patch_state	Статус ядра по применению "живых" патчей (live patching) к этому процессу.
limits	Ограничения на ресурсы, установленные для процесса.
oom_score	Оценка для OOM killer. Чем выше, тем вероятнее процесс будет убит при нехватке памяти.
oom_score_adj	Файлы для корректировки oom_score.
cgroup	Контрольные группы, к которым принадлежит процесс.
cpuset	Какие CPU и узлы памяти разрешено использовать процессу.
autogroup	Информация о группе автоматической балансировки планировщика.
cpu_resctrl_groups	Группы контроля ресурсов CPU.
ns	Директория со ссылками на Namespaces, в которых "живет" процесс.
uid_map	Карты для трансляции ID пользователей в Namespaces.
gid_map	Карты для трансляции ID пользователей в Namespaces.
projid_map	Карты для трансляции ID проектов в Namespaces.
setgroups	Используется в User Namespaces, чтобы разрешить или запретить процессу изменять свой список групп.
attr	Атрибуты безопасности.
io	Статистика по операциям ввода-вывода, выполненным процессом (прочитано/записано байт).
mounts	Список файловых систем, смонтированных с точки зрения этого процесса.
mountinfo	Более подробная информация о монтировании.
mountstats	Статистика по точкам монтирования.
net	Директория с сетевой статистикой, специфичной для сетевого пространства имен процесса.