



ТЕХНОТРЕК

Занятие №2

Системное администрирование Linux

Сергей Ключков

Загрузка системы



- На данном этапе BIOS (UEFI) передает управление загрузчику, а загрузчик — ядру операционной системы
- Ядро отвечает за распределение ресурсов системы и абстракцию оборудования от приложений
- После инициализации оборудования ядро запускает `init` — основной процесс пространства пользователя. `Init` отвечает за запуск пользовательского окружения

Загрузка ОС



- Два основных варианта загрузки системы — по сети или с локального диска (CD, HDD, дискеты, ...)
- Для загрузки по сети система получает конфигурацию сети и указания о том, где взять загрузчик, по DHCP/Bootp.
- Для загрузки с локального диска необходимо, чтобы на этом диске была поддерживаемая BIOS'ом таблица разделов и присутствовал загрузчик.
- Основные виды таблицы разделов — MBR и GPT. Загрузку с GPT поддерживают новые версии BIOS на базе UEFI.

Загрузчик



- На MBR загрузчик и таблица разделов расположены на первых 512 байтах диска.
- `dd if=/dev/sda bs=512 count=1 | file -`
- При использовании GPT необходимо создать отдельный раздел для загрузчика.
- Загрузчик на примере grub2
- Stage1 и stage2
- `/boot/grub2/grub.cfg` и его содержимое
- Формирование конфигурации загрузчика. `grub2-mkconfig: /etc/grub.d` и `/etc/default/grub`

Загрузка ядра



- На следующем этапе загрузчик запускает ядро ОС.
- Ядро Linux имеет модульную архитектуру. Для того, чтобы можно было воспользоваться модулями на раннем этапе загрузки, используется `initrd`.
- После подгрузки требуемых модулей ядра монтируется корневая ФС и запускается `init` — процесс, отвечающий за инициализацию пользовательского окружения.



- Зачем вообще нужно ядро ОС
- Как управлять ядром. Sysctl. /etc/sysctl.{conf,d}
- Модули ядра — динамически загружаемые части ядра.
- Взаимодействие ядра и пользовательского окружения.
- ПсевдоФС, предоставляемые ядром. /dev, /sys и /proc

Что такое процесс



- Процессы состоят из следующих компонентов:

Образ исполняемого кода

Адресное пространство

Набор дескрипторов

Атрибуты доступа (uid, gid, лимиты, etc)

Контекст процессора

- Принципы распределения ресурсов между процессами будут подробно разобраны на 11 занятии



- В /proc содержится информация о состоянии процесса
- Подробный справочник — `man proc`
- Адресное пространство процесса. Информация из `/proc/<pid>/status`.
- Разница между виртуальной и резидентной памятью.
- Выделение памяти процессам. Понятие страницы.
- Huge pages, обычные и прозрачные

Память процесса



- Разница между виртуальной и резидентной памятью.
- Выделение памяти процессам. Понятие страницы.
- Информация о выделенной процессу памяти
- Huge pages, обычные и прозрачные

Дескрипторы



- Дескриптор — идентификатор ресурса ввода/вывода
- Стандартные дескрипторы — 0 (stdin), 1 (stdout), 2 (stderr)
- Прочие дескрипторы выделяются для открытых файлов, установленных сетевых соединений и т. п.
- /proc/<pid>/fd, /proc/<pid>/fdinfo, /proc/net/* содержат информацию об открытых дескрипторах процесса
- lsof — утилита для просмотра дескрипторов процессов

Потребление ресурсов процессора



- Основная информация — в `/proc/<pid>/stat`
- Общая информация о потреблении cpu в системе - `/proc/stat`
- `User, system, iowait`
- Работать с информацией в таком виде не всегда удобно. Утилита `top`
- Отслеживание производительности дискового I/O. `iostat`
- `iostat -xmd 1`

Системные вызовы



- Системные вызовы — механизм взаимодействия ядра и приложений
- Как работают системные вызовы. `linux-vdso.so`
- Отслеживание системных вызовов. Команда `strace`
- `strace -p 1 -f -tt -y`
- `strace -p 1 -f -c`

Сигналы



- Сигналы — механизм связи с процессом
- Они предназначены для управления процессами и оповещения о событиях
- Обработчики сигналов
- Все ли сигналы можно обработать?
- kill, killall, pkill — основные утилиты командной строки для отправки сигналов.
- Man 7 signal

Лимиты процесса



- Что такое Ulimit
- `/etc/security/limits.conf`
- Мягкие и жесткие лимиты
- `man limits.conf`
- `/proc/<pid>/limits`
- `ulimit -a`
- Изменение лимитов запущенного процесса.
`prlimit`
- `prlimit --pid 1 --nofile=65536:65536`

Переменные окружения



- Переменные окружения — динамически изменяемый набор ключей и значений, относящийся к конкретному процессу
- `printenv`
- `export`
- `/proc/<pid>/environ`
- Основные служебные переменные окружения
- `LD_PRELOAD`
- `LD_LIBRARY_PATH`

Как размножаются процессы



- Как запустить процесс
- Всем известный `fork()`
- Используется ли `fork()`?
- `strace -f -u su -c ls 2>&1 | less`
- Семейство `exec()`
- Дерево процессов
- Откуда берутся зомби

Что делает процесс



- Системные вызовы, осуществляемые процессом, можно отслеживать с помощью `strace`
- Если процесс не делает системных вызовов, но потребляет ресурсы, как понять, чем он занимается? Утилита `pstack`
- Все равно понятно. Нужна отладочная информация
- `yum install --enablerepo=base-debuginfo glibc-debuginfo`
- Теперь видно гораздо больше



- Что делать, если нужно отследить действия, выполняемые процессом и в пространстве пользователя, и в пространстве ядра
- Подсистема perf отвечает за сбор “слепок” состояния процесса. Статистика по этим следам используется для отладки и профилирования.
- `perf top -p 1`
- `perf trace ls`
- `perf stat`



- Как узнать, что происходило раньше?
- Логи системы и логи приложений
- dmesg
- Системный логгер. Зачем он нужен?
- `/var/log/messages` — основной лог системы
- `/var/log/secure` — события, связанные с логином пользователей и повышением привилегий
- Логи приложений

Домашнее задание №2



Скрипт, автоматизирующий добавление пользователей в систему

Дано: набор имен пользователей, групп, в которые они должны входить, их домашних директорий, хешей паролей, опционально – прочих атрибутов в выбранном вами формате.

Необходимо: написать bash-скрипт, автоматизирующий добавление данных пользователей в систему

20 баллов

Спасибо за внимание!



Тема следующего занятия – Linux и сеть