

Этапы загрузки ОС Linux

Масалыгин К.К. группа 11-ИБ

Когда-то давно термин bootstrapping (загрузка) в компьютерной области означал загрузку бумажной ленты, на которой хранилась программа начальной загрузки, или же ввод программы начальной загрузки вручную при помощи расположенных на передней панели переключателей адреса/данных/управления. Современные компьютеры оборудованы устройствами, которые значительно упрощают процесс первоначальной загрузки — однако это не означает, что этот процесс является простым.

Далее мы рассмотрим процесс загрузки ОС Linux, который состоит из 3 этапов.

Описание 1 этапа

Не углубляясь в кучу терминов и определений, данный этап можно описать следующими словами: BIOS из MBR (первые 512 байт диска, выбранного для загрузки) загружает First Boot Loader. FSB находит вторичный загрузчик, используя таблицу разделов, просматривая ее, обнаруживает активный раздел, после обнаружения этого раздела — загружает SSB в оперативную память и запускает его. Для корректной загрузки, активный раздел должен содержать каталог /boot, который должен находиться в начале диска и содержать Second Stage Boot Loader. В целом, SSB - это программа, которая выводит список вариантов загрузки (меню выбора загрузки операционной системы). Загрузчиком может быть LILO (более старый) или GRUB. Загрузчик берет свои настройки из конфигурационного файла (/etc/lilo.conf — для LILO и /boot/grub/grub.conf или /boot/grub/menu.lst -для GRUB). Существуют и другие версии загрузчиков, такие как syslinux, PXELinux, Isolinux, uBoot, но для наглядности, в статье затронуты только LILO и GRUB. Отметим, что исторически (до появления загрузчиков LILO, GRUB и др. и когда образ ядра занимал объем не более 1,44 Мб) данного этапа не существовало и загрузка

происходила с дискеты без файловой системы, на которую был записан образ ядра Linux, который (образ) содержал в себе MBR, то есть BIOS загружал сразу образ ядра и передавал ему управление.

Описание 2 этапа

Второй этап можно характеризовать так: подготовка системы для запуска служб. При подготовке, Загрузчик загружает в память образ ядра из каталога /boot. Образ ядра, инициализирует и подготавливает память, процессор (процессоры), остальное оборудование, монтирует корневой раздел в режиме только для чтения для загрузки остальной системы (устройство и раздел на котором размещен корень системы должен быть указан в настройках загрузчика GRUB (/boot/grub.conf) или LILO (/boot/lilo.conf)) в виде параметра root=. При этом, выводится сообщение VFS: Mounted root (ext2 filesystem) readonly. Кроме того, ядро из конфигурационного файла загрузчика получает параметры загрузки, такие как корневая файловая система, отображать сообщения ядра или нет и т.п.

Т.к. ядро Linux является модульным, то при загрузке может возникнуть необходимость подключить модуль ядра, который находится на еще не примонтированной файловой системе. Для решения данной проблемы при загрузке подгружается архив файловой системы (он же инициализационный RAM диск или initrd), содержащий в себе необходимый для загрузки набор модулей ядра.

Т.к. стандартный вывод (вывод сообщений на экран) должен быть связан с каким-либо процессом, соответственно с идентификатором процесса, а у ядра нет идентификатора, оно помещает сообщения ядра (и модулей) в буфер кольца ядра и выводит на экран. Данный буфер еще называется dmesg. Его содержимое можно просмотреть, выполнив команду dmesg. После полной инициализации ядро передает управление процессу init (первому системному процессу с PID=1). На экран выводится сообщение INIT: version 2.76 booting. При этом, бинарный файл init последовательно ищется в корневом разделе в каталогах: /sbin/init,/etc/init, /bin/init, если в указанных местах не обнаружен файл, то ядро

пытается запустить шелл `/bin/sh` (это, собственно, есть однопользовательский режим загрузки, он же режим восстановления). При этом, не запускается ни одна служба. Если не найден и шелл, то вываливается ошибка `Kernel panic: No init found. Try passing init= option`. Данная ситуация может возникнуть скорее всего, потому что неверно смонтирован корневой раздел.

Описание 3 этапа

До текущего момента процесс запуска любой UNIX системы практически не отличался. Третий этап загрузки может отличаться в зависимости от платформы, будь то Linux, BSD, MacOS и др. Мы рассмотрим процесс загрузки операционной системы Linux с реализацией процесса запуска с помощью пакета `sysvinit` (так же именуемого System V - "систем 5"). В Linux в последнее время активно внедряется разработанный "убунтологами" пакет `upstart`, который приходит на смену SysV, данный процесс так же будет кратко описан.

System V

После запуска, процесс `init`, согласно конфигурации в файле `/etc/inittab` (а точнее строке, начинающийся на `si::sysinit:/etc/.....`) первым делом выполняет скрипт `/etc/rc.d/rc.sysinit` (для RedHat) или `/etc/init.d/rcS` (для Debian), которые выполняют базовое конфигурирование системы (загрузка модулей, проверка корневой ФС и монтирование не чтение/запись, установка имени хоста, времени, монтирование оставшихся разделов, запуск сети, монтирование сетевых ФС и др.), а так же данный скрипт с помощью утилиты `initlog` направляет сообщения о загрузке в `/var/log/messages`. На данном этапе, нет уровня выполнения. Далее, запускается скрипт инициализации `/etc/rc.d/rc` (для RedHat) или `/etc/init.d/rc` (для Debian), которому передается уровень запуска в виде параметра от 0 до 6 (в соответствии с настройками из файла `/etc/inittab`, в котором указан уровень загрузки (выполнения) ОС по умолчанию и каталог `/etc/rc*.d`, в котором расположены скрипты запуска демонов/служб для соответствующего уровня запуска), запускает скрипты из каталога, соответствующего текущему уровню запуска.

Далее, процесс `init` согласно уровню загрузки просматривает каталог `/etc/rc.d/rc0.d/` (в данном примере, цифра 0(ноль) в имени `rc0.d` соответствует уровню загрузки - нулевому), в котором содержатся симлинки (ссылки) на скрипты запуска системных служб, которые, в свою очередь, расположены в `/etc/rc.d/init.d/` (для Red Hat) или в `/etc/init.d/` (для Debian).

Ссылки имеют следующий формат:

`<S|K><число><имя_службы>`, в котором: S - запуск службы (Start), K - остановка службы (Kill), `<число>` - число, определяющее последовательность запуска служб (00 - самая первая), `<имя_службы>` - имя запускаемой службы.

Уровни выполнения бывают следующие:

- 0: полная остановка машины;
- 1: single-user (однопользовательский) режим; (используется в случае серьезных проблем или для восстановления системы)
- 2: multi-user (многопользовательский) режим, без поддержки сети;
- 3: Multi-user (многопользовательский) режим с поддержкой сети; (используется преимущественно на серверных системах)
- 4: неиспользуемый;
- 5: Multi-user (многопользовательский) режим с поддержкой сети + графический интерфейс для входа в систему (login);
- 6: перезагрузка.

Upstart

Начиная с Ubuntu 6.10 старый добрый `init` был заменен более функционально-продвинутым аналогом, которое авторы назвали `Upstart`. Основной козырь `Upstart` в том, что его работа основана на событиях. Это означает, что, в отличие от `init`, `Upstart` запускает и останавливает задачи не просто вызывая соответствующие shell-скрипты, а еще и наблюдает за работой запущенных им задач, основываясь на событиях, получаемых им от приложений. Это дает возможность, например, перезапустить в случае чего внезапно упавшую службу самим демоном `Upstart`, а не возлагать это на какие-то потусторонние программы.

Итак, процессом `init` генерируется событие `startup` (запуск - одно из двух основных событий) - старт системы, а событие `shutdown` – при выключении системы. Другое ключевое событие — это `ctrlaltdel`, которое указывает, на то, что вы нажали клавиши `Ctrl-Alt-Delete`. В соответствии с событием генерируемым процессом `init`, обрабатываются конфигурационные файлы (`*.conf`) из каталога `/etc/init/` (в более старых версиях `upstart` — каталог `/etc/event.d/`). В этом, собственно, и заключается вся работа `upstart`. Для обратной совместимости с `System V` существует файл `/etc/init/rc.conf`, который запускает демоны, согласно `SysV`.

Рассмотрим содержимое файлов `/etc/init/*.conf`. Каждый из файлов должен содержать `start on event` (по какому (`on`) событию (`event`) осуществлять запуск (`start`) службы), `stop on event` (по какому событию останавливать запуск службы) ,`exec` либо `script`(что, собственно, запустить по указанному выше событию). Наиболее часто применяемые события в конфигурационных файлах — это `startup`, `runlevel`, `stopped` и `started`. `Startup` - это событие запуска операционной системы, `runlevel` (с указанием уровня или нескольких уровней) - событие при смене уровня запуска, `stopped` - событие после остановки задания, `started` - событие после завершения старта задания. Просмотрев конфигурационные файлы в каталоге `/etc/init/` можно найти и другие события. В дополнение к стандартным записям `exec` и `script` существуют записи `pre-start` и `post-stop`, реализующие выполнение подготовительных действия до выполнения `exec` и завершающие после завершения `exec`.

Данной системой загрузки также можно управлять командами.

Для отключения автоматического запуска службы в `Upstart`, необходимо:

- Переименовать конфигурационный файл запуска службы в каталоге `/etc/init` в файл без расширения `".conf"`.
- Или закомментировать строку `"start on"` с помощью символа `'#'`.

Список литературы

1. «IBM developer Works» [Электронный ресурс] - (дата посещения 11.11.12) - Режим доступа: <http://www.ibm.com/developerworks/ru/library/l-linuxboot/>.
2. «OpenSours в заметках» [Электронный ресурс] — (дата посещения 11.11.12) - Режим доступа: <http://www.ashep.org/2009/linux-upstart/>.
3. «Блог Любителя экспериментов» [Электронный ресурс] – (дата посещения 11.11.12) - Режим доступа: <http://www.k-max.name/linux/nachalo-etapy-zagruzki-os-linux-v-sxeme/>.