**Работа со сценариями запуска системы**

|  |  |  |
| --- | --- | --- |
| Выход из однопользовательского режима |  | Демон ***init*** выполняет сценарии запуска системы (= сценарии интерпретатора sh (=bash)) |
| Стандартная загрузка (по завершении работы интерпретатора команд, запущенного с правами суперпользователя /bin/sh) |

Сценарии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Нумеруются и выполняются по порядку |  | Хранятся в каталоге:  /etc/init.d |  | Ссылки созданы в каталогах:  /etc/rc0.d  /etc/rc1.d |
|  |  |  |  |  |
| Система может обеспечивать соответствующие зависимости между службами |  |  |  |  |
|  |  |  |  |  |

**Начальная загрузка**

Начальная загрузка (bootstrapping) – запуск системы при включении питания.

В ходе этого процесса

* ядро системы загружается в память и активизируется;
* выполняется ряд инициализационных задач
* система переходит к состоянию готовности к обслуживанию пользователей

Начальная загрузка - период особой уязвимости системы.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Включение  питания | **→** | Запуск на выполнение загрузочного кода, хранящегося на ПЗУ | **→** | Запуск  ядра (после его загрузки в RAM) | **→** | Опрос аппаратных устройств ядром | **→** | Запуск демона  ***init***  (PID=1) |

|  |  |  |
| --- | --- | --- |
| ***ПЕРЕД полной загрузкой системы***  **init** → интерпретатор команд → последовательный запуск сценариев = ***реализация процедур*** |  | **Загрузка системы** |
| Проверка и монтирование файловых систем |
| Запуск системных демонов |

!

Главное – активизировать командную оболочку!

(запуск /bin/sh при загрузке является последним шагом перед объявлением kernel\_panic в случае, если не был обнаружен init)

*Однопользовательский режим* (режим восстановления, профилактический режим) – режим работы ОС \*NIX, при котором только активизируется командная оболочка на системной консоли (/bin/sh).

В этом режиме нельзя выполнять сетевые операции.

Как перейти в однопользовательский режим:

1. во время начальной загрузки передать ядру в командной строке определённый параметр;
2. если система уже загружена и работает, можно перевести её в однопользовательский режим с помощью команды shutdown или telinit

Этапы загрузки:

* считывание начального загрузчика с главной загрузочной записи;
* загрузка и инициализация ядра;
* обнаружение и конфигурирование устройств
* создание процессов ядра;
* [вмешательство администратора # только в однопользовательском режиме]
* Выполнение системных сценариев запуска

Как администратор может управлять загрузкой системы:

* отредактировать файлы конфигурации для сценариев запуска системы;
* изменить аргументы, передаваемые загрузчиком → ядру

**Инициализация ядра**

Ядро = программа.

Первая задача начальной загрузки – записать эту программу (ядро) в память для последующего выполнения.

Имя файлу ядра даёт его изготовитель.

Традиционно оно называется:

/unix или /vmunix

В Linux-системах - /boot/vmlinuz

**Два этапа загрузки ядра:**

1. С помощью кода, записанного на ПЗУ, в память PC с диска скачивается начальный загрузчик;
2. Начальный загрузчик выполняет собственно загрузку ядра

Часть внутренних структур ядра имеет фиксированный размер → ядро резервирует определённый объём физической памяти для самого себя. Эта память не доступна пользовательским процессам

**Ядро**

Тестирование системных шин

Инвентаризация оборудования

Вывод на консоль краткой информации о каждом обнаруженном устройстве

Загружает драйверы устройств

как *независимые модули ядра*

Готовые ядра имеют модульную структуру и автоматически обнаруживают большую часть устройств

**Создание процессов ядра**

|  |  |  |
| --- | --- | --- |
| **Параметр сравнения** | **UNIX-системы** | **Linux** |
| После завершения этапа базовой инициализации ядро создаёт в области памяти, выделенной для пользовательских программ, несколько «самовыполняющихся» процессов ***в обход fork [exec без fork]*** | | |
| PID процессов ядра | Используется *sched*, PID = 0 | процессы имеют низкие номера PID (процесс с PID = 0 отсутствует) |
| init и другие процессы ядра: особенности, представление |  | Демон *init* работает в сопровождении с различными обработчиками памяти и сигналов ядра;  Имена процессов ядра в листинге команды ps заключены в квадратные скобки.  Иногда имена процессов могут содержать в конце: */<цифра>* (например, [kblockd**/0**])  число указывает процессор, на котором выполняется данный процесс |
| Главное отличие init от других процессов ядра | init – полноценный пользовательский процесс; остальные фактически представляют собой части ядра, которые были сделаны процессами из концептуальных соображений | |
|  | процессы ядра отображают особенности конкретной реализации ядра →  никакие имена и функции не могут быть одинаковыми для разных систем |  |

Параметры командной строки, передаваемые ядру Linux программой загрузчика (GRUB)

|  |  |
| --- | --- |
| **Параметр** | **Назначение** |
| acpi=off | Отключает компоненты Advanced Configuration (усовершенствованный интерфейс конфигурации) и Power Interface (интерфейс управления питанием) |
| init=/bin/bash | Заставляет ядро запускать только интерпретатор bash; используется при восстановлении системы в случае сбоев |
| root=/dev/foo | Сообщает ядру о том, что корневым является устройство **/dev/foo** |
| single | Задаёт режим однопользовательской загрузки (только для Linux. Используйте ключ –s в системах Solaris – он предназначен для администраторов, знакомых со стандартом OpenBoot в других CPU-архитектурах) |

Параметры ядра, отредактированные во время загрузки, **не сохраняются**.

Чтобы сохранить изменения на будущие перезагрузки:

→ отредактировать строку kernel в файле grub.conf или menu.lst

Ядро Linux постоянно совершенствуется с помощью заплат.

Старые версии ядра Linux не заменяются новыми, а сосуществуют с ними.

**Демон init и его уровни выполнения**

Демон init (PID = 1) – это первый процесс, который выполняется после завершения загрузки системы.

init - базовый для всех пользовательский и почти для всех системных процессов.

В разных системах – разная реализация демона init (незначительно).

|  |  |  |
| --- | --- | --- |
| Уровень | Описание | Особенности |
| 0 | Система полностью прекратила работу | Система не может оставаться на этом уровне. |
| 1 | Однопользовательский режим\* работы системы |  |
| S | Создаётся отдельный процесс, отображающий приглашение ввести пароль;  в Solaris и AIX: однопользовательский режим «в действии»  в Linux: имеет переходный характер и завершается сразу после ввода пароля. |
| 2 | Предназначены для поддержки работы в сети | (в большинстве случаев, уровень системы по умолчанию) |
| 3 | (в большинстве случаев, уровень системы по умолчанию) |
| 4 | Используется редко |
| 5 | В Linux используется регистрационными процессами X Windows |
| 6 | Определяет этап перезагрузки системы | Система не может оставаться на этом уровне. |

\* При однопользовательском режиме работы

* запрещены все сетевые сеансы и процессы удалённой регистрации;
* в системе выполняется минимальный набор программ;
* доступ к системе осуществляется с правами суперпользователя.

В файле /etc/inittab содержатся параметры, определяющие, что должен делать демон init на каждом уровне.

Суть: в файле задаются команды, которые должны быть выполнены (или продолжить выполнение), когда система переходит на конкретный уровень.

Команда telinit

* позволяет изменить уровень выполнения демона init, когда система полностью функциональна.

telinit –q → предписывает демону init перечитать файл /etc/inittab.

Пример указания уровня 5 выполнения (SUSE, inittab) в качестве используемого по умолчанию:

**id:5:initdefault:**

# E.Nemeth: UNIX and Linux System Administrator (2018)

Несколько общих задач процесса загрузки:

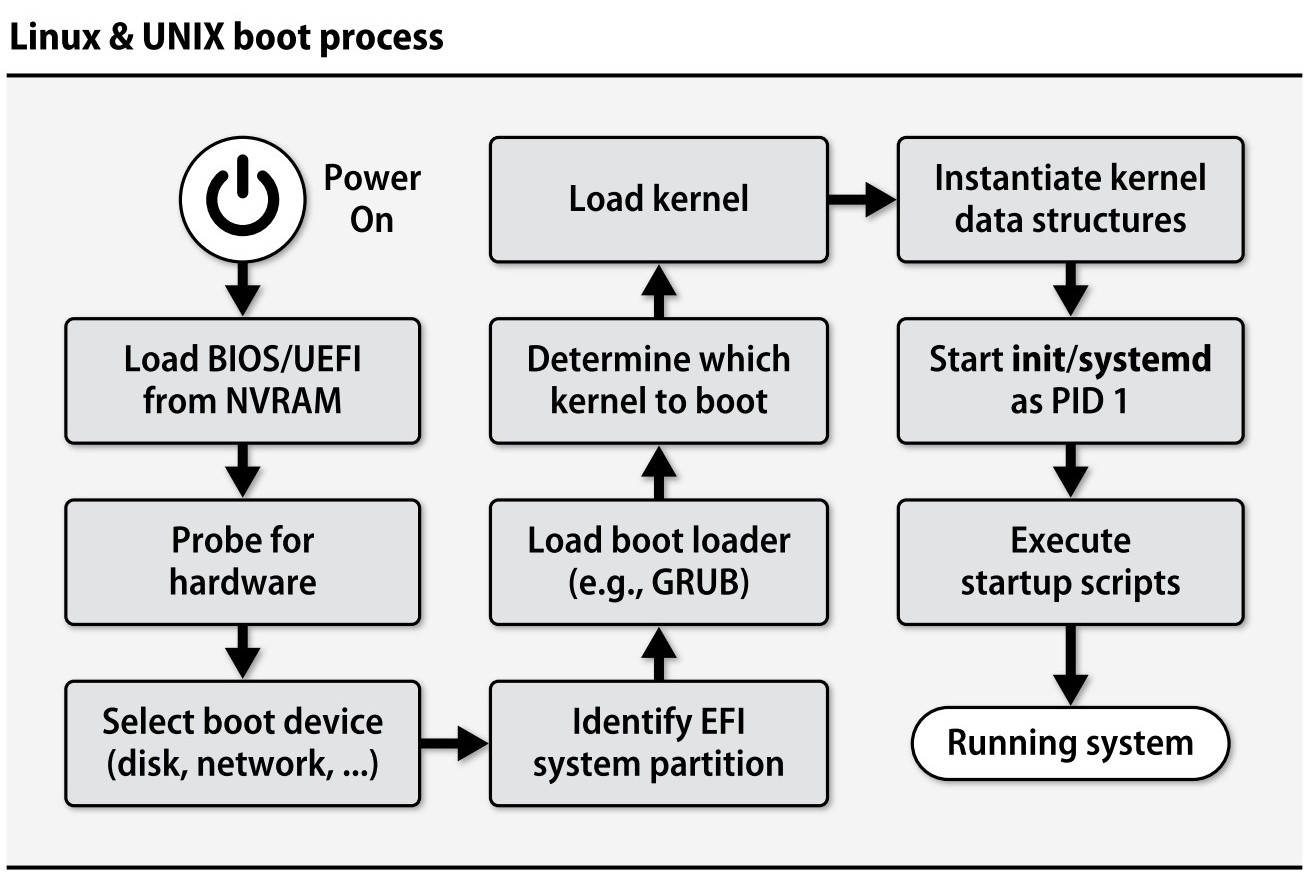
* нахождение, загрузка и запуск кода начальной загрузки (bootstrapping code);
* нахождение, загрузка и запуск ядра операционной системы;
* запуск сценариев запуска (startup scripts) и системных демонов;
* поддержание гигиены процесса (maintaining process hygiene) и управление переходами состояний системы.

Действия, включенные в этот последний пункт, продолжаются до тех пор, пока система остается работоспособной, поэтому грань между начальной загрузкой и нормальной работой по своей сути немного размыта.

Процедуры запуска сильно изменились за последние годы. Появление современных (UEFI) BIOS упростило ранние этапы загрузки, по крайней мере, с концептуальной точки зрения. На более поздних стадиях большинство дистрибутивов Linux теперь используют демон системного администратора под названием systemd вместо традиционного процесса инициализации UNIX.

systemd упрощает процесс загрузки, добавляя управление зависимостями, поддержку параллельных процессов запуска и комплексный подход к ведению журнала, среди других функций.

Управление загрузкой также изменилось по мере миграции систем в облако. Сдвиг в сторону виртуализации, облачных инстансов (экземпляров) и контейнеризации уменьшил потребность администраторов в работе с физическим оборудованием. Вместо этого теперь у нас есть управление образами, API и панели управления.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Включение питания |  | Загрузка ядра |  | Создание экземпляров структур данных ядра |
|  |  |  |  |  |
| Загрузка BIOS/UEFI из NVRAM |  | Определение ядра для загрузки |  | Запуск init/system как первого процесса (PID 1) |
|  |  |  |  |  |
| Сканирование оборудования |  | Загрузка в память загрузчика (например, GRUB) |  | Выполнение сценариев запуска |
|  |  |  |  |  |
| Выбор устройства загрузки (диск, сеть,…) |  | Определение системного раздела EFI |  | Запуск системы |

Администраторы имеют ограниченный непосредственный интерактивный контроль над большинством шагов, необходимых для загрузки системы. Вместо обычной загрузки администраторы могут изменять конфигурации начальной загрузки, редактируя файлы конфигурации для сценариев запуска системы или изменяя аргументы, которые загрузчик передает ядру.

Перед полной загрузкой системы необходимо:

|  |  |  |
| --- | --- | --- |
| * проверить и смонтировать файловые системы | процедуры управляются серией сценариев оболочки («скриптами инициализации») или юнит-файлами | последовательно запускаются init  или  анализируются systemd |
| * запустить системные демоны |

Точная компоновка сценариев запуска и способ их выполнения различаются в зависимости от системы.