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) |

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 (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 | **→** | Опрос аппаратных устройств ядром | **→** | Запуск демона  ***init***  (PID=1) |

|  |  |  |
| --- | --- | --- |
| ***Процедуры, которые реализуются с помощью сценариев интерпретатора команд, последовательно запускаемых демоном init (перед полной загрузкой системы)*** |  | **Загрузка системы** |
| Проверка и монтирование файловых систем |
| Запуск системных демонов |

!

Главное – активизировать командную оболочку!

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

Тестирование системных шин

Инвентаризация оборудования

**Ядро**

Внутри Linux нет принципиальных отличий между программой и командой.

Как узнать, с *программой* или *командой* мы имеем дело?

- Необходимо знать, какой файл при этом выполняется.

Программный файл: имеет бит доступа *x* (execute);

Остальные файлы: не имеют бита доступа x

Программный файл Процесс (под управлением ядра Linux)

**ЗАПУСК**

Библиотека Mono – библиотека, относящаяся к платформе .NET-Framework, позволяет запускать отдельные exe-файлы, написанные на языке C#. Библиотека с открытым исходным кодом.

Запуск программ

Быстрый запуск для систем с GUI KDE и Gnome: Alt + F2

Запуск команд в командном окне (xterm, konsole) или в текстовой консоли:

→ указать название программы и нажать ENTER (если программный файл находится в одном из каталогов, перечисленных в переменной окружения PATH)

$PATH: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

|  |
| --- |
| Двоеточие — это эквивалент пустой команды. Может рассматриваться как синоним встроенной команды true.  Команда **:** также является встроенной командой Bash, которая всегда возвращает true (0) |

→ указать полный путь к программному файлу (если файле находится не в директории, описанной в PATH)

Если каталог текущий, путь указываем с точкой: ./program

Приоритетные и фоновые программы

Программа, запущенная через GUI (меню) → фоновый процесс;

фоновые процессы могут запускать другие программы до завершения инициирующей программы

Программа, запущенная через текстовую консоль/программное окно:

по умолчанию → приоритетный процесс (требует завершения текущего процесса перед запуском новой программы)

с символом **&** после названия программы: → фоновый процесс

Как перевести запущенную программу в фоновый режим?

1. Прервать выполнение программы, нажав Ctrl + Z
2. Возобновить программу с помощью команды bg

Полезные команды

|  |  |  |
| --- | --- | --- |
| Команда | Описание | Примечания |
| iotop | отобразить доступ к жёстким дискам и другим носителям данных | -o отобразить процессы, которые заняты операциями ввода-вывода  -u отобразить процессы текущего пользователя (собственные процессы) |
| pidof <программа> | узнать соответствующий программе PID | Если имеется несколько процессов с одинаковым названием, выводится весь список номеров |
| fuser [параметр] <имя\_файла/каталога> | вывести список процессов, работающих с заданным файлом, сетевым портом или файловой системой | Каталог считается задействованным, если в нём была запущена программа  -v вывести максимально подробную инфо об объекте  [https://linux-faq.ru/page/komanda-fuser] |
| pstree [параметр] | отобразить иерархию процессов | -h выделить процессы, являющиеся родительскими для текущего |

!

Факт доступа к файлу может быть установлен лишь в том случае, когда программа открыла файл на достаточно долгий период времени.

Текстовый редактор:

* Открывает файл
* Загружает данные из файла
* Закрывает файл
* Если требуется сохранить изменения, снова открывает файл

Некоторые фоновые процессы сохраняют в каталоге ***PID-файл***.

Первая строка PID-файла: номер процесса

Остальные строки:

|  |  |
| --- | --- |
| Вопрос | Описание |
| Кем создаются | Некоторыми фоновыми процессами |
| Где сохраняются | /var/run |
| Что содержится:  - в первой строке  - в остальных строках | номер процесса;  дополнительная информация (например, данные о сетевом интерфейсе) |
| Зачем нужны | обеспечивают целенаправленное завершение определённого процесса с помощью системы init (в том числе когда существует несколько одноимённых процессов) |