

# Введение в Linux

(19-01-2015)

## 1. Linux

**Linux** — общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux создаётся и распространяется в соответствии с моделью разработки свободного и открытого программного обеспечения. Поэтому общее название не подразумевает какой-либо единой «официальной» комплектации Linux. Системы распространяются в основном бесплатно в виде различных готовых дистрибутивов, имеющих свой набор прикладных программ и уже настроенных под конкретные нужды пользователя.

Примеры дистрибутивов: Debian, Ubuntu, CentOS, OpenSuse

## 2. Bash

**Bash** - Одна из наиболее популярных современных разновидностей командной оболочки UNIX. Особенно популярна в среде Linux, где она часто используется в качестве предустановленной командной оболочки. Bash также может читать команды из файла, который называется скриптом. Как и все Unix-оболочки, он поддерживает автодополнение имён файлов и директорий, подстановку вывода результата команд, переменные, контроль за порядком выполнения, операторы ветвления и цикла.

## 3. Файловые системы

Linux умеет работать с большим количеством файловых систем.

**Ext2** - файловая система, используемая в операционных системах на ядре Linux. Достаточно быстра для того, чтобы служить эталоном в тестах производительности файловых систем. Она не является журналируемой файловой системой и это её главный недостаток.

**Ext3** - журналируемая файловая система, используемая в ОС на ядре Linux. Основана на Ext2, но отличается тем, что в ней есть журналирование, то есть в ней предусмотрена запись некоторых данных, позволяющих восстановить файловую систему при сбоях в работе компьютера.

**Ext4** - журналируемая файловая система, используемая в ОС на ядре Linux. Основана на файловой системе Ext3, но отличается тем, что в ней представлен механизм пространственной записи файлов, уменьшающий фрагментацию и повышающий производительность.

**Fat16** - файловая система, сейчас широко используемая в картах памяти фотоаппаратов и других устройств.

**Fat32** - файловая система основанная на Fat16. Создана, чтобы преодолеть ограничения на размер тома в Fat16.

**NTFS** - файловая система для семейства операционных систем Microsoft Windows.

**JFS** - журналируемая файловая система. В отличие от Ext3, в которую добавили поддержку журналирования, JFS изначально была журналируемой. На момент выхода в свет JFS была самой производительной из существовавших файловых систем. На текущий момент сохраняет за собой одно из лидирующих мест по этому показателю.

**ReiserFS** - журналируемая файловая система, разработанная специально для Linux. Обычно под словом ReiserFS понимают третью версию (последняя — 3.6.21), а четвёртую называют Reiser4. В настоящий момент разработка Reiser3 прекращена.

**Reiser4** - журналируемая файловая система ReiserFS (4-я версия), разработанная специально для Linux. Одна из самых быстрых файловых систем для Linux (с включённым плагином-архиватором - самая быстрая).

**UFS** - файловая система, созданная для операционных систем семейства BSD. Linux поддерживает UFS на уровне чтения, но не имеет полной поддержки для записи UFS. Родной Linux ext2 создан по подобию UFS.

**XFS** - высокопроизводительная журналируемая файловая система. Распределение дискового пространства - экстендами, хранение каталогов в B-деревьях. Автоматическая аллокация и высвобождение I-node. Дефрагментируется «на лету». Невозможно уменьшить размер существующей файловой системы. Возможны потери данных во время записи при сбое питания (хотя этот недостаток нельзя относить к одной только XFS, он свойственен любой журналируемой ФС, но вместе с тем, XFS, по умолчанию, достаточно активно использует буферы в памяти).

**SWAP** - раздел жёсткого диска, предназначенная для виртуальной памяти (раздел подкачки).

Для установки системы Linux требуется минимум два раздела жесткого диска. Один раздел для установки собственно самой системы и раздел подкачки SWAP.

#### 4. Структура

В Linux каждому диску и разделу не назначается буква (как в Windows и DOS). Вместо этого задается точка монтирования для каждого диска и раздела. Linux работает по принципу иерархического дерева каталогов, где корневой каталог ( / ) является основной точкой монтирования, в которую по умолчанию входят все остальные.

Можно считать, что / в Linux - это эквивалент C:\ в Windows, где по умолчанию хранятся все системные файлы и папки.

Директория	Описание
/	Корневая директория, содержащая всю файловую иерархию
/bin/	Основные системные утилиты
/boot/	Загрузочные файлы
/dev/	Основные файлы устройств системы
/etc/	Общесистемные конфигурационные файлы
/home/	Содержит домашние директории пользователей, которые в свою очередь содержат персональные настройки и данные пользователя
/lib/	Основные библиотеки, необходимые для работы программ из /bin/ и /sbin/.
/media/	Точки автоматического монтирования для сменных носителей.
/mnt/	Точки ручного монтирования для сменных носителей.
/opt/	Дополнительное программное обеспечение.
/proc/	Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде каталогов файлов.
/root/	Домашняя директория пользователя root.
/sbin/	Основные системные программы для администрирования и настройки системы
/tmp/	Временные файлы (см. также /var/tmp).
/usr/	Вторичная иерархия для данных пользователя; содержит большинство пользовательских приложений и утилит
/usr/bin/	Дополнительные программы для всех пользователей
/usr/include/	Стандартные заголовочные файлы.
/usr/lib/	Библиотеки для программ, находящихся в /usr/bin/ и /usr/sbin/.
/usr/sbin/	Дополнительные системные программы (такие как демоны различных сетевых сервисов).
/var/	Изменяемые файлы, такие как log-файлы, временные почтовые файлы
/var/lib/	Информация о состоянии. Постоянные данные, изменяемые программами в процессе работы (например, базы данных, метаданные пакетного менеджера и др.).
/var/lock/	Lock-файлы, указывающие на занятость некоторого ресурса.
/var/log/	Различные log-файлы.
/var/mail/	Почтовые ящики пользователей.
/var/run/	Информация о запущенных программах (в основном, о демонах).
/var/spool/	Задачи, ожидающие обработки (например, очереди печати, непрочитанные или неотправленные письма).
/var/tmp/	Временные файлы, которые должны быть сохранены между перезагрузками.

#### 5. Библиотеки

Библиотека — это сборник подпрограмм или объектов, используемых для работы программного обеспечения. В Linux представлена в виде файла.

Динамическая библиотека — библиотека, которая загружается в память процесса при создании процесса, либо по запросу уже работающего процесса, то есть динамически при необходимости.

Расширения файлов в системе Linux **“.so”** (англ. shared object).

В зависимости от назначения различают:

- библиотеки, используемые одной программой и содержащие критические для работы программы функции.

Недостаток: при отсутствии библиотеки программа не сможет работать;

- библиотеки, используемые одной программой и содержащие дополнительные функции. Например, библиотеки плагинов используются для расширения функциональности программы;

- библиотеки общего пользования (англ. shared library). Содержат функции, используемые несколькими программами. Могут загружаться в адресное пространство ОС (англ. system library) для экономии памяти: одна копия библиотеки будет использоваться несколькими процессами.

Статическая библиотека — файл с исходным кодом или объектный файл, предназначенный для вставки в программу на этапе компоновки.

Библиотеки, распространяемые в виде исходного кода, преобразуются компилятором в объектные файлы. Затем компоновщик соединяет объектные файлы библиотек и объектные файлы программы в один исполняемый файл.

Расширения файлов в системе Linux **“.a”**.

## 6. Уровни выполнения

«Уровень выполнения» означает режим функционирования операционной системы компьютера, в которой реализована инициализация в стиле ОС UNIX System V. Традиционно существуют семь уровней выполнения, пронумерованных от 0 до 6.

Система в определенный момент времени находится на соответствующем уровне выполнения. Администратор системы может переводить её с одного уровня выполнения на другой. Это делается при помощи программы `init` (или `telinit`). Для этого программе в качестве аргумента передается число соответствующее уровню выполнения. Например, чтобы перевести систему на 3-й уровень выполнения, необходимо запустить команду `«init 3»`

В различных дистрибутивах Linux уровни выполнения используются для различных целей.

Современная версия программы `init` может использовать десять уровней выполнения, но обычно используются только семь.

0 — выполняются действия по выключению системы.

1 — однопользовательский режим (single user mode). Предназначен для различных административных действий по восстановлению системы. По своему смыслу аналогичен Safe Mode Windows, но полностью его не повторяет. На этом уровне выполнения система полностью сконфигурирована, но не запущен ни один сервис, а из пользователей может работать только администратор (`root`).

2 — не используется, но сконфигурирован как уровень выполнения 3. В RedHat и SuSE Linux сконфигурирован как уровень выполнения 3, но без поддержки сетевых файловых систем. В Ubuntu и Debian используется как многопользовательский режим.

3 — многопользовательский режим (multiuser mode). Нормальный режим работы сервера.

4 — В Slackware Linux используется для графического входа в систему. В RedHat и SuSE Linux не сконфигурирован.

5 — В RedHat и SuSE Linux используется для графического входа в систему. В Slackware Linux не сконфигурирован.

6 — выполняются действия по перезагрузке системы.

В любой момент времени проверить текущий уровень можно командой `runlevel`

## 7. Пользователи, группы и права доступа

Любой файл и каталог в Linux имеет пользователя-владельца и группу-владельца. То есть любой файл и каталог принадлежит какому-то пользователю системы и какой-то группе. Кроме того, у любого файла и каталога есть три группы прав доступа:

- для пользователя-владельца,
- для членов группы-владельца,
- для всех остальных пользователей системы.

Каждая группа состоит из прав на чтение, запись и запуск файла на исполнение.

Владельцы		Права								
		Пользователь — владелец			Группа — владелец			Остальные пользователи		
Пользователь	Группа	Чтение	Запись	Выполнение	Чтение	Запись	Выполнение	Чтение	Запись	Выполнение
User	Group	x	x	x	x	x	x	x	x	x

Могут быть установлены следующие права на файлы и каталоги:

2я система с.	8я система с.	Право	Представление
000	0 (0+0+0)	Нет прав	---
001	1 (0+0+1)	Выполнение	--x
010	2 (0+2+0)	Запись	-w-
011	3 (0+2+1)	Запись+Выполнение	-wx
100	4 (4+0+0)	Чтение	r--
101	5 (4+0+1)	Чтение+Выполнение	r-x
110	6 (4+2+0)	Чтение+Запись	rw-

8. Системные переменные (переменные среды)

В системе Linux используется большое количество различных системных переменных. Ниже перечислены основные из них.

- \$BASH - содержит полный путь к исполняемому файлу командной оболочки Bash.
- \$CDPATH - хранит пути поиска каталога. (используется при вводе команды cd имя\_каталога без слэша)
- \$HOME - домашний каталог текущего пользователя.
- \$HOSTNAME - имя компьютера.
- \$LANG - текущая установка локализации
- \$PATH - список каталогов для поиска команд и приложений, когда полный путь к файлу не задан.
- \$PWD - полный путь к текущему рабочему каталогу.
- \$SHELL - полный путь к текущей командной оболочке.
- \$USER - имя текущего пользователя.

Администратор системы может менять значения этих переменных и применить изменения для всей системы. Пользователи системы также могут менять значения переменных, но изменения будут использоваться только пользователем в его среде.

Для получения значений системных переменных (переменных среды) следует пользоваться командой printenv

9. Установка и удаление приложений

В системе Linux приложения устанавливать и удалять обычно можно различными способами.

Основные способы установки:

- 1) Установка с помощью пакетного менеджера со скачиваем пакета из хранилища пакетов

Пример:

- CentOS: yum
- Ubuntu: apt-get
- OpenSuse: yast

- 2) Установка уже скачанного пакета с помощью пакетного менеджера (без взаимодействия с хранилищем пакетов)

Пример:

- CentOS: rpm
- Ubuntu: dpkg
- OpenSuse: rpm

### 3) Установка из исходных кодов

Примеры:

make

make install

configure, make, make install

cmake, make, make install

Пакетные менеджеры используются для установки пакетов (packages).

Пакет — это архив, содержащий уже скомпилированное приложение, его компоненты и доп. информацию для установки приложения в системе.

Установка из исходных кодов производится дольше чем установка из пакетов, но у нее есть свои особенности и преимущества:

- установка самой последней версии приложения;
- возможность внесения необходимых изменений в исходный код приложения;
- установка приложения, подобного уже установленному из пакета, но с другими настройками;
- установка приложения и его компонентов в определенную директорию.

Для удаления программ, установленных с помощью пакетного менеджера, следует также использовать пакетный менеджер.

Для удаления программ, установленных из исходных кодов достаточно удалить все установленных файлы.

## 10. Работа с процессами

В системе Linux любой пользователь может завершать свои процессы (приложения), сворачивать их и возвращаться к работе с ними снова.

Команда/комб. Клавиш	Описание
ctrl + c	остановка процесса
ctrl + z	сворачивание процесса
jobs	получение списка свернутых процессов
fg	восстановление работы процесса на переднем плане
bg	восстановление работы процесса в фоне