

Типы и назначение операционных систем. ОС Linux



Андрей
Тряпичников



Андрей Тряпичников

Системный администратор

Одноклассники



[Андрей Тряпичников](#)

a.tryapichnikov@gmail.com

План занятия

1. [Операционные системы](#)
2. [История и структура Linux](#)
3. [Стандарт POSIX](#)
4. [Real Time или RT-ядро](#)
5. [Интерфейс командной строки](#)
6. [Программное обеспечение для Linux](#)
7. [Итоги](#)
8. [Домашнее задание](#)



Операционные системы

Что такое операционная система?

Операционная система — комплекс программ, обеспечивающий управление аппаратными средствами компьютера, организующий работу с файлами и выполнение прикладных программ, осуществляющий ввод и вывод данных.

Пользовательская ОС отличается продвинутым графическим интерфейсом.

Серверная ОС — высокой надежностью и производительностью.

Дистрибутив

Дистрибутив — форма распространения операционной системы, разнящийся программами для начальной инициализации системы (в частности, установщиком), наборами предустановленного ПО и наборы пакетов.

На данный момент самыми распространенными семействами дистрибутивов являются **.DEB** (Debian) и **.RPM** (Red Hat).



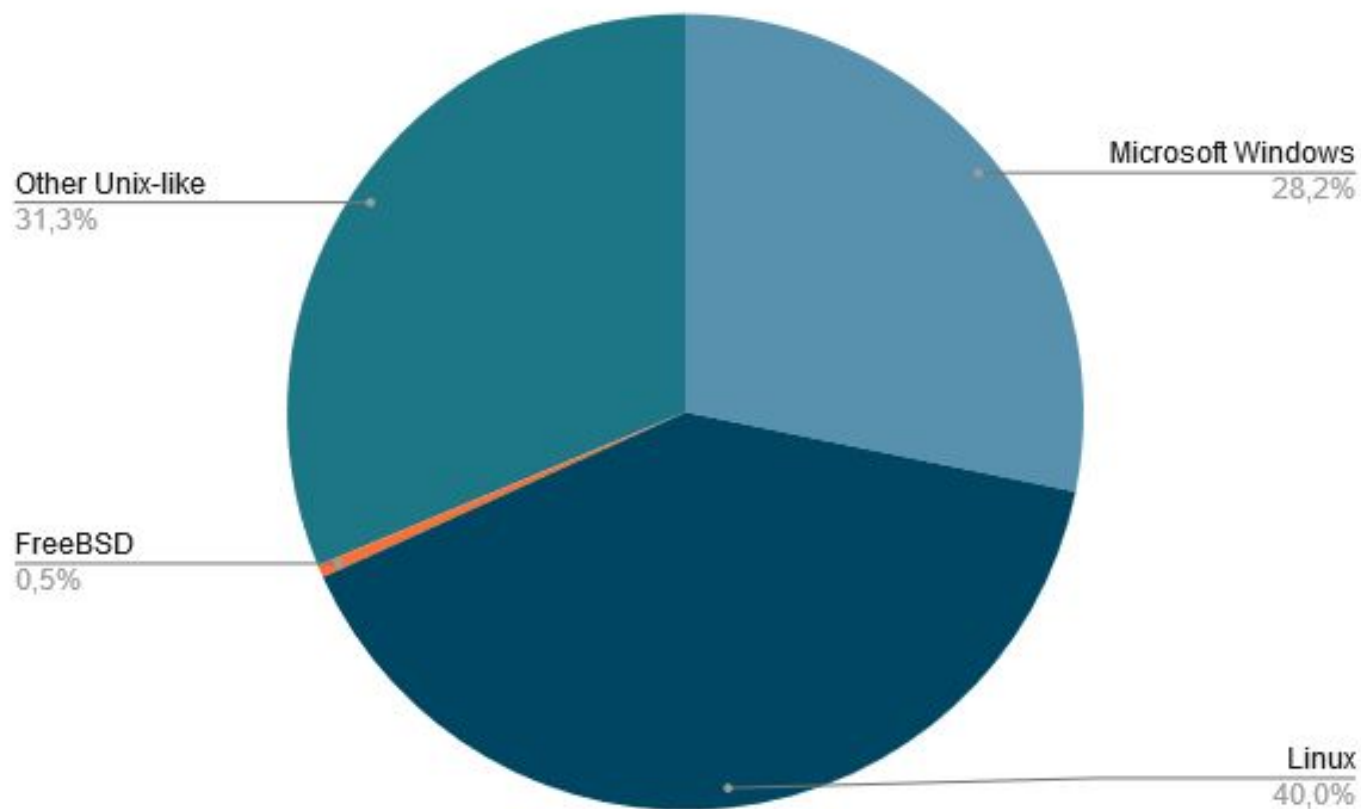
Операционные системы в мире

Наиболее популярных семейств операционных систем в мире всего четыре:

- **MacOS**, разработки компании Apple;
- **BSD** (семейство);
- **Microsoft Windows**;
- **GNU / Linux**.

Хорошо зная одну систему из семейства, можно разобраться в остальных.

Распределение ОС на web-серверах



Данные взяты: https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

Почему Linux?

- бесплатность и открытость;
- популярность;
- отказоустойчивость;
- информативность;
- безопасность;
- гибкость.

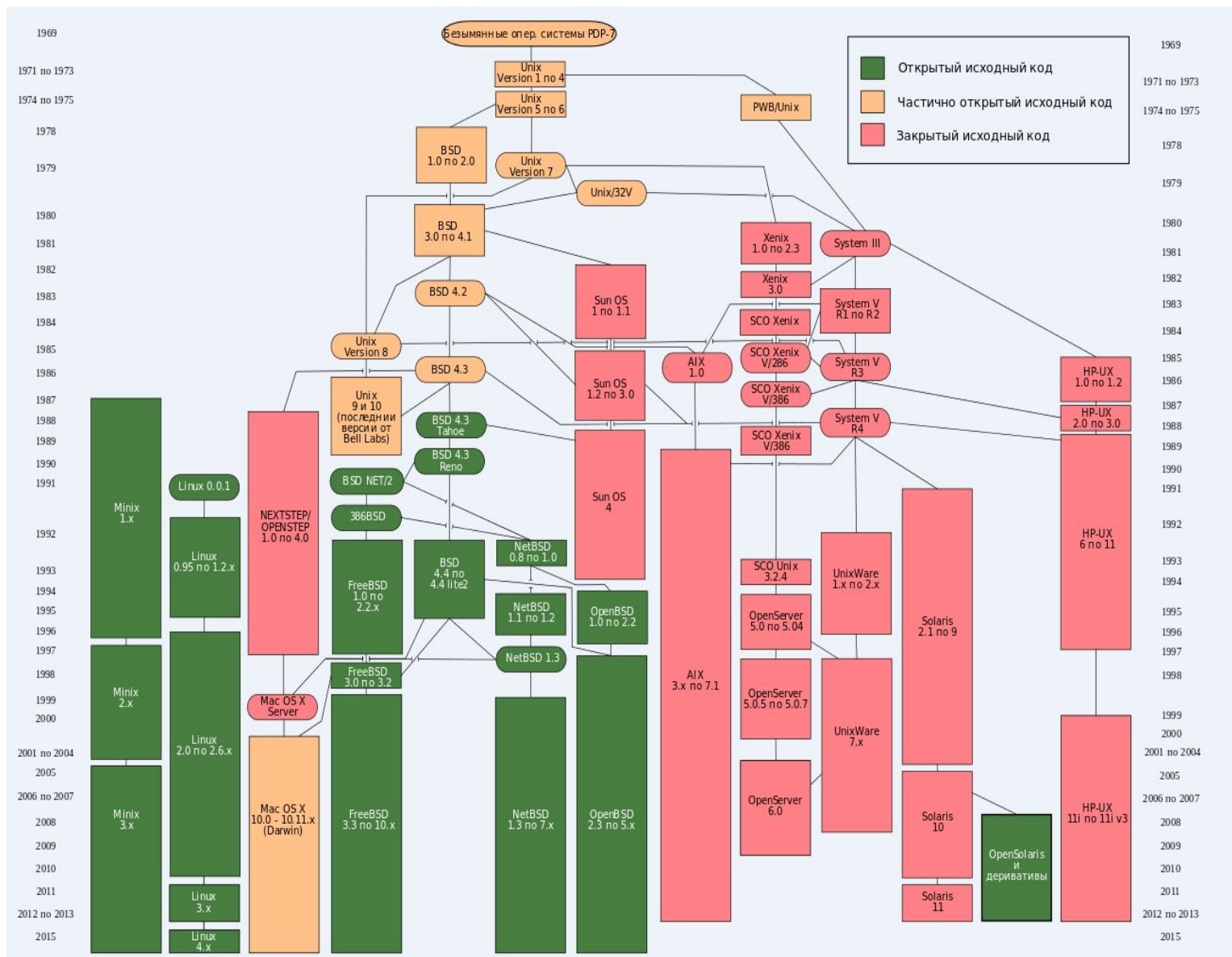


История и структура Linux

Unix как основа всего

- Unix разработан в конце 60-х годов в исследовательском центре Bell Labs Кеном Томпсоном, Деннисом Ритчи и Дугласом Макилроем;
- повлиял на огромный пласт Unix-подобных операционных систем;
- изначально многопользовательский, многозадачный и мультиплатформенный.
- в Unix все является файлом: клавиатура, настройки, каталоги.

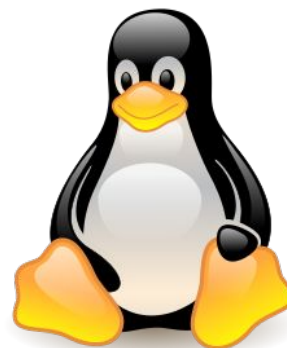
Генеалогическое древо Unix



Рождение Linux

Линус Торвальдс, будучи студентом, заинтересовался операционными системами, но споткнулся об лицензию MINIX, и начал разрабатывать свою систему в 1991 году.

Развивая свою систему, продолжил разработку уже на фактической базе **ядра Linux**, начав использовать ПО, распространяемое под **лицензией GPL**.



Принципы открытого ПО и GPL

- свобода запуска программы с любой целью;
- свобода изучения того, как программа работает, и её модификации;
Предварительное условие: доступ к исходному коду.
- свобода распространения копий как исходного, так и исполняемого кода;
- свобода улучшения программы и выпуска улучшений в публичный доступ.
Предварительное условие: доступ к исходному коду.

Структура Linux

1. User Applications

2. O/S Services

3. Kernel

4. Hardware Controllers

1. **Пользовательские приложения**
Редакторы, браузеры, медиа
2. **Службы операционной системы**
Оконный менеджер, командная строка, компиляторы и библиотеки
3. **Ядро Linux**
4. **Контроллеры оборудования**
Процессора, памяти, средств ввода-вывода



Стандарт POSIX

Стандарт POSIX

- набор стандартов, который описывает интерфейсы между ОС и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов;
- все Unix-подобные операционные системы являются POSIX-совместимыми в той или иной мере;
- POSIX определяет утилиты командной строки, которые должны быть в совместимых системах;
- Linux используют свой стандарт: Linux Standard Base (LSB), основанный на POSIX.



GNU Core Utilities

GNU Core Utilities (или `coreutils`) — пакет программного обеспечения GNU, содержащий большое количество основных утилит, необходимых для UNIX-подобных операционных систем.

Основывается на утилитах стандарта POSIX, практически полностью их копируя.

Применим во всех вариациях Linux.

Владея этими командами, вы не потеряетесь ни в одном дистрибутиве Linux.

Real Time Kernel

RT-ядро жертвует **производительностью** ради **предсказуемости** результата.

Бывает *мягкое* и *жесткое*: жесткое всегда укладывается в план, мягкие могут позволить себе некоторые вольности.

Важно для обеспечения **гарантированной работоспособности** служб или приложений.



Интерфейс командной строки

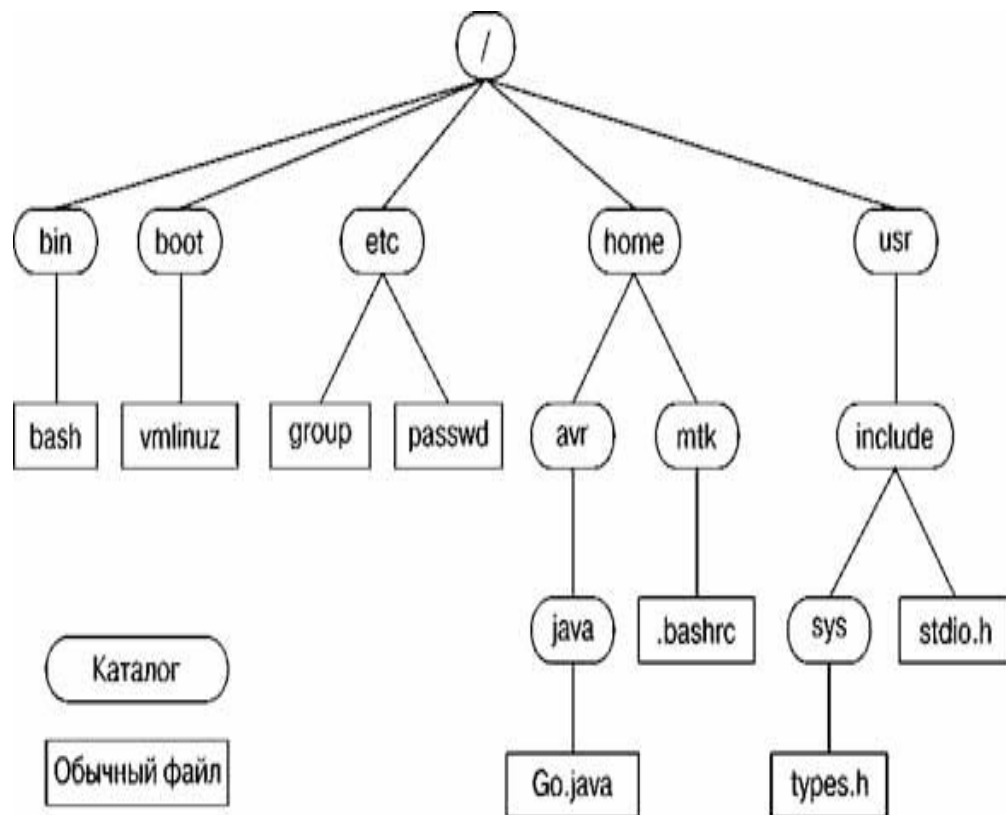
CLI: консоль

Существует два вида интерфейсов: Graphic User Interface (GUI) и Command Line Interface (CLI), известный как «консоль».

Преимущества консоли:

- простота;
- гибкость;
- наглядность;
- информативность;
- удобство.

Иерархия каталогов Linux



Рабочий каталог — каталог, с которого начинается относительный путь к цели команды.

`.` — алиас (псевдоним) рабочего каталога.

`..` — алиас, (псевдоним) каталога на уровень выше текущего каталога.

Источник: <https://oracle-patches.com>

Стандартный синтаксис команды и понятия

Синтаксис команды стандартно **включает** в себя саму **команду**, **ключи**, или, как их еще называют, **опции**, и **цель или объект** команды. Например:

ls -lah /etc/init.d/

Где:

ls - команда

-lah - набор ключей (опций) этой команды

/etc/init.d/ - цель этой команды

Стандартные потоки ввода-вывода

Поток номер 0 (STDIN) зарезервирован для *чтения команд* пользователя или *входных данных*. Как правило, не применим в системных программах.

- По умолчанию нацелен на чтение с клавиатуры.
- Перенаправляется при помощи <

Поток номер 1 (STDOUT) зарезервирован для *вывода данных*, как правило текстовых.

- По умолчанию выводит данные в консоль.
- Перенаправляется при помощи >

Поток номер 2 (STDERR) зарезервирован для вывода *диагностических и отладочных сообщений* в текстовом виде.

Команды

- **cd** *<путь к каталогу>* — переход в целевой каталог
- **ls** *<ключи>* *<путь к каталогу>* — просмотр содержимого каталога
- **mkdir** *<путь к каталогу>* — создает каталог в указанном месте
- **rm** *<ключи>* *<путь к файлу>* — удаляет файл (применимо для удаления каталогов)
- **cp** *<исходный файл>* *<целевой файл>* — копирует исходный файл в целевой
- **mv** *<исходный файл>* *<целевой файл>* — перемещает исходный файл в целевой

Команды

- **echo** *<строка>* — выводит строку в STDOUT
- **cat** *<файл>* — выводит в STDOUT содержимое файла
- **head** *<файл>* — выводит первые 10 строк файла в STDOUT
- **tail** *<файл>* — выводит последние 10 строк файла в SDTOUT
- **less** *<файл>* — показывает файл в построчном выводе

Команды

- **grep** *<паттерн>* — выводит строки в которых есть совпадения с паттерном. Может работать с целевым файлом\файлами и стандартным выводом.
- **tar** — стандартный архиватор Linux. Архивирует и разархивирует
- **du** — показывает размер каталогов.
- **find** — поиск файлов и каталогов.
- **ln** — создание ссылки на файл или каталог
- **man** — открыть документацию по команде

Права пользователя и повышение привилегий

Пользователь по умолчанию не имеет права работать с системными файлами, устанавливать приложения, запускать программы, работающие ниже пользовательского уровня.

root — пользователь, обладающий всей полнотой власти на компьютере. Работать из-под root категорически не рекомендуется!

Для повышения привилегий пользователя используется команда **sudo**, написанная перед выполняемой командой. Установлен по умолчанию не на всех дистрибутивах. Необходимо выдать пользователю право повышать привилегии.

su *<пользователь>* — команда, позволяющая сменить пользователя.



Программное обеспечение для Linux

Репозитории и менеджеры пакетов

Репозиторий — хранилище данных, в данном случае — пакетов ПО.

Менеджер пакетов - специальное программное обеспечение, которое управляет загрузкой, установкой, удалением пакетов, а также решением зависимостей.

Наиболее популярными менеджерами пакетов являются **apt** (семейство Debian) и **yum** (семейство Red Hat)

Менеджер файлов Midnight Commander

Left	File	Command	Options	Right			
← /			[^]>	← /			
'n	Name	Size	Modify time	'n	Name	Size	Modify time
/bin		4096	May 25 00:08	UP--DIR			
/boot		4096	May 25 00:09	/.cache		4096	Jun 29 20:04
/cdrom		4096	May 24 00:41	/.config		4096	Jun 29 21:51
/dev		4120	Jul 21 13:25	/.dbus		4096	May 24 00:50
/etc		12288	Jul 21 13:26	/.fontconfig		4096	May 25 00:08
/home		4096	May 24 00:42	/.gconf		4096	Jul 21 13:25
/lib		4096	May 25 00:08	/.gconfd		4096	Jul 21 13:33
/lost+found		16384	May 24 00:14	/.gnome2		4096	May 24 00:51
/media		4096	Jul 21 13:25	/.gstreamer-0.10		4096	Jun 29 21:49
/mnt		4096	Apr 21 18:50	/.gvfs		0	Jul 21 13:25
/opt		4096	Jun 29 21:40	/.local		4096	May 24 00:50
/proc		0	Jul 21 13:24	/.mc		4096	Jul 21 13:30
/root		4096	May 25 00:02	/.nautilus		4096	May 24 00:51
/sbin		4096	Jun 29 21:41	/.pulse		4096	Jul 21 13:25
/selinux		4096	Mar 21 09:26	/Desktop		4096	May 24 00:50
/srv		4096	Apr 26 00:50	/Documents		4096	May 24 00:50
/sys		0	Jul 21 13:24	/Downloads		4096	May 24 00:50
/tmp		4096	Jul 21 13:30	/Music		4096	May 24 00:50
/usr		4096	Apr 26 00:56	/Pictures		4096	May 24 00:50
/var		4096	Apr 26 01:06	/Public		4096	May 24 00:50
@initrd.img		32	May 24 00:43	/Templates		4096	May 24 00:50
@vmlinuz		29	May 24 00:43	/Videos		4096	May 24 00:50
				./ICEauthority		3500	Jul 21 13:25
				./bash_history		502	Jul 21 13:30
				./bash_logout		220	May 24 00:42
				./bashrc		3353	May 24 00:42
				./dmrc		78	Jul 21 13:25
				./esd_auth		16	May 24 00:50
				./gksu.lock		0	May 25 00:01
				./gtk-bookmarks		132	Jul 21 13:25
				./profile		675	May 24 00:42
/bin				UP--DIR			
6608M/9574M (69%)				6608M/9574M (69%)			

Двухпанельный файловый менеджер, позволяющий оперировать командами командной строки, со встроенным редактором и просмотрщиком текста.

Один из лучших инструментов для работы с файлами в консоли.

Текстовые редакторы vi и nano

Nano — простой в управлении текстовый редактор, пригодный для небольших правок.

Vi — **модальный** текстовый редактор, предназначенный для углубленной работы. Имеет два режима работы: модальный, для перемещения по тексту, поиска, замены, выполнения внутренних команд и режим ввода, для, соответственно, ввода текста.

Vi входит в пакет coreutils, и следовательно, присутствует в любой системе на базе Unix.

Итоги

Сегодня мы узнали:

- зачем нужна ОС и какие семейства существуют;
- историю Unix и как она повлияла на современный мир операционных систем;
- почему профессионалы выбирают Linux;
- что такое стандарт POSIX и для чего он нужен;
- основы работы в консоли;
- ключевые команды в консоли;
- основы работы с программным обеспечением для Linux.


Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Андрей Тряпичников

 Андрей Тряпичников

a.tryapichnikov@gmail.com