

Знакомство с операционной системой Linux

Андрей Тряпичников
Senior Unix engineer в ZFX



Андрей Тряпичников

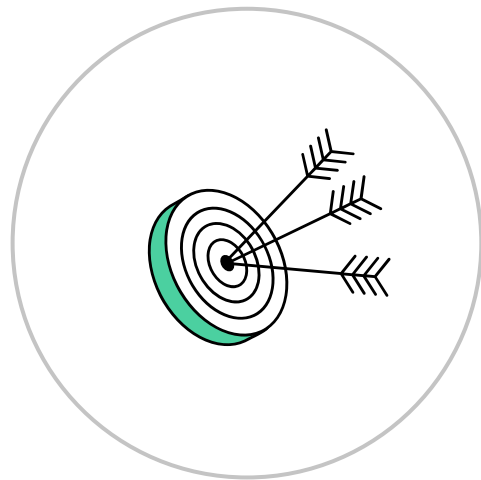
О спикере:

- senior Unix engineer в ZFX
- более 10 лет опыта работы системным администратором, из них 6 лет в социальной сети Одноклассники



Цели занятия

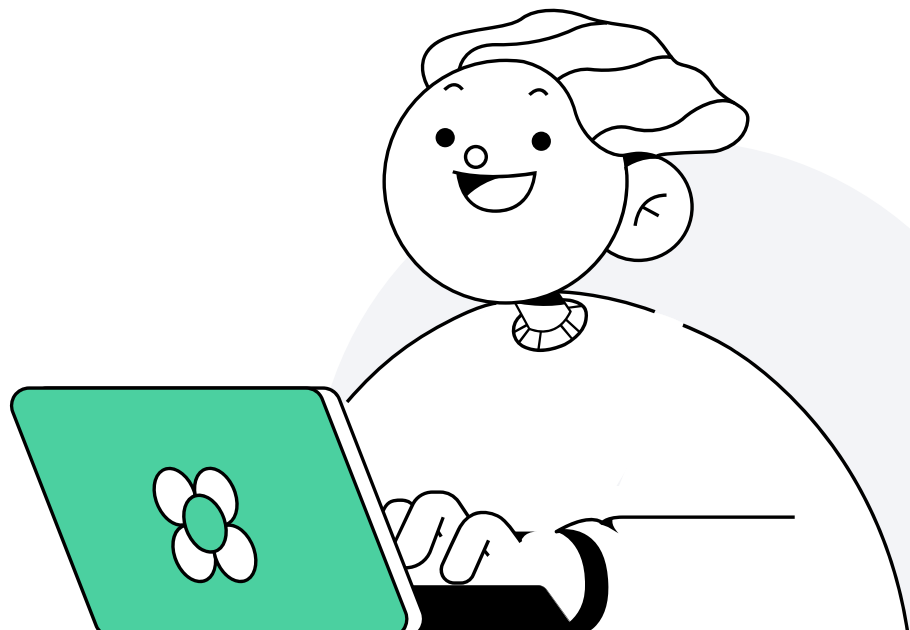
- Поговорить об истории возникновения Linux
- Познакомиться с ОС Linux и её дистрибутивами
- Поработать в командной строке
- Разобраться с такими определениями, как пользователь, права доступа, группы пользователей



План занятия

- 1 История Unix
- 2 История Linux
- 3 ОС Linux
- 4 Дистрибутивы
- 5 Командная строка
- 6 Пользователи
- 7 Итоги

*Нажми на нужный раздел для перехода



Вспоминаем предыдущие занятия

Вопрос: назовите популярные
операционные системы?



История Unix

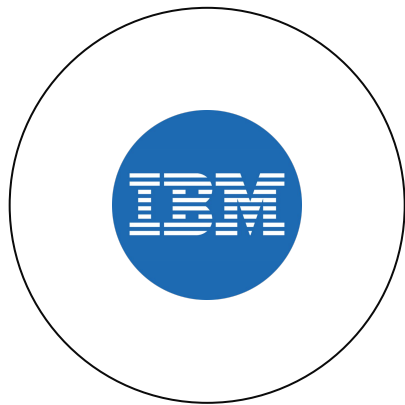


1

Как было до появления операционных систем

До появления универсальных ОС производители компьютерного оборудования (IBM, DEC) вместе с оборудованием предлагали одну или несколько ОС.

ОС писались на ассемблере, который был уникален для каждой платформы



Как было до появления операционных систем

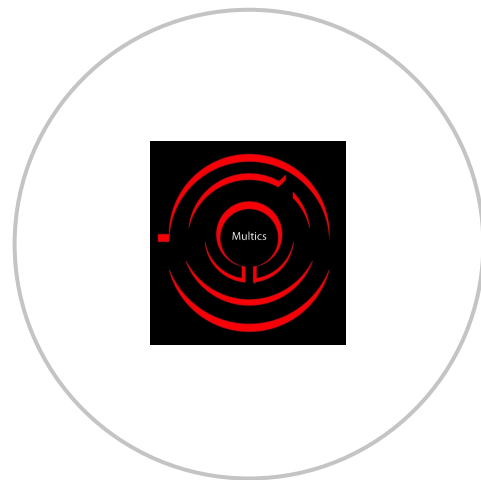
	1945–60-е	1960–70-е	1970–80-е	1980–90-е	Настоящее время
ПО	Машинные языки	Алгоритмические языки, пакетный режим, ОС	ОС, разделение времени	Базы данных, мультимикропроцессорные системы	-
Носитель	Перфокарта	Магнитная лента	Магнитный диск	Гибкий диск	SSD
Применение	Научно-технические расчёты	Обработка числовой и текстовой информации	АСУ, САПР	Все сферы деятельности + интернет	Интеллектуальные системы

Multics

Multics (сокращение от Multiplexed Information and Computing Service — Мультиплексная информационная и вычислительная служба) — это проект ОС с разделением времени от MIT (Массачусетского технологического института).

Проект стартовал в 1965 г. В нём принимали участие технологические гиганты General Electric и AT&T.

ОС разрабатывалась для работы с компьютерами GE-645 и Honeywell 6180, в ней было реализовано несколько инновационных идей, которые перекочевали в Unix и другие ОС. Например, концепция виртуальной памяти и централизованная иерархическая файловая система



Unix

Unix (Uniplexed Information and Computing Service, сокращённо — UNICS) — семейство переносимых, многозадачных и многопользовательских операционных систем, которые основаны на идеях оригинального проекта AT&T Unix, разработанного в конце 1960-х годов в исследовательском центре Bell Labs



UNIX®

Unix

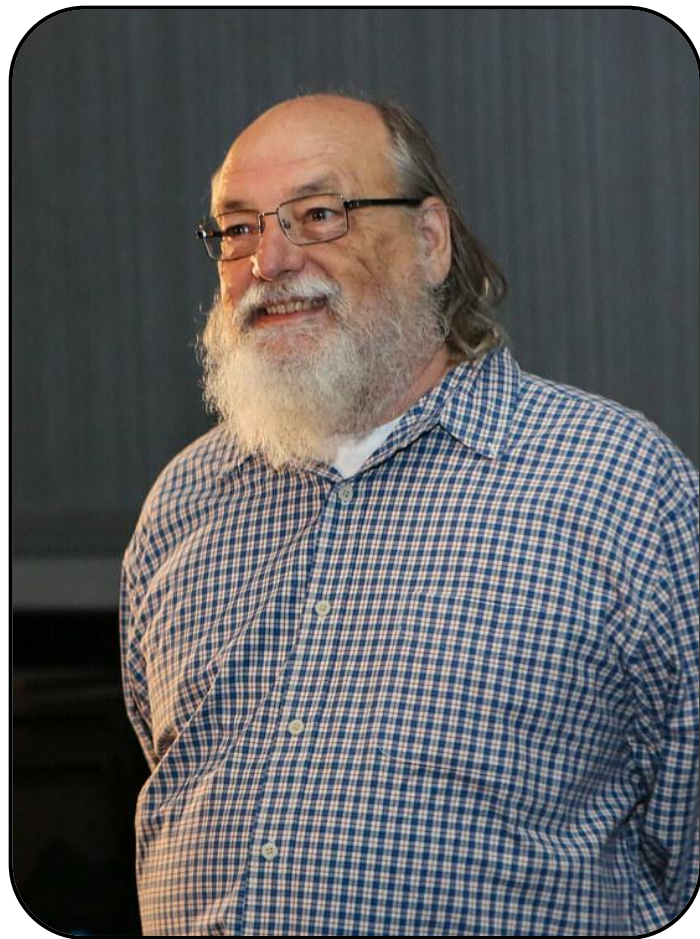
Unix написан на ассемблере DPD-7 от компании Digital Equipment Corporation (DEC) и потом переписан на языках высокого уровня: сначала В, потом С. Язык С был достаточно высокоуровневым и при этом достаточно мощным для создания ОС.

Перенос первоначальных версий Unix между машинами состоял в перекомпиляции исходного кода компилятором под новую архитектуру. В 1974 году было 50 установок Unix



Кен Томпсон

Кеннет Лейн (Кен) Томпсон (Kenneth Lane Thompson, род. 4 февраля 1943 г.) — пионер компьютерной науки, известен своим вкладом в создание языка программирования С и операционной системы Unix. Работал в лаборатории Bell Labs



Деннис Ритчи

Деннис Макалистэйр Ритчи

(Dennis MacAlistair Ritchie, 1941–2011 гг.) —
компьютерный специалист, известен
своим вкладом в создание языков
программирования В, С, участием
в разработке операционных систем
Multics и Unix



Дуглас Макилрой

Малкольм Дуглас Макилрой

(Malcolm Douglas McIlroy, род. 1932 г.) — американский математик, инженер и программист. Известен разработкой конвейера в операционной системе Unix, принципов компонентно-ориентированного программирования и нескольких популярных Unix-утилит: spell, diff, sort, join, speak, tr



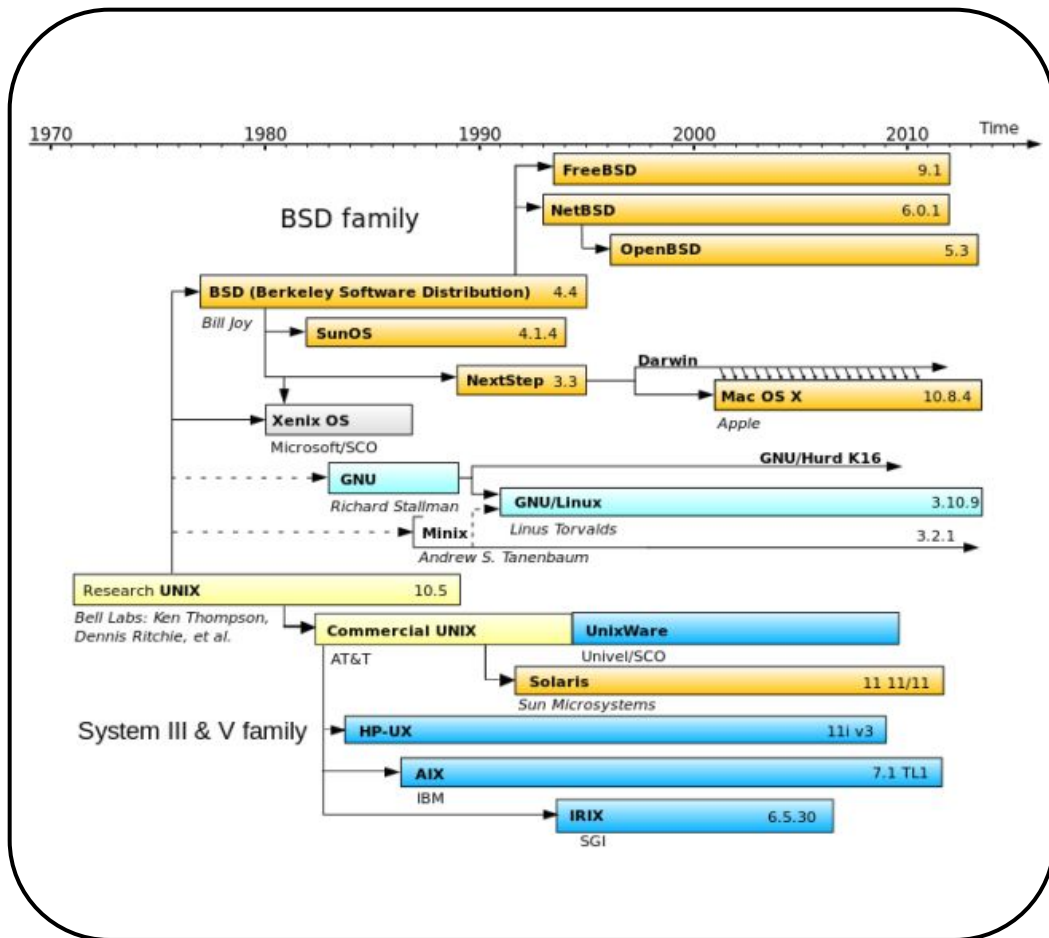
Появление Unix-подобных ОС

Из-за ограничений, наложенных правительством США на AT&T, компания не могла использовать в коммерческих целях ОС Unix. Код Unix был передан нескольким вузам, которые начали выпускать свою версию ОС. Например, университет Беркли в 1978 году выпускает BSD Unix, продолжая дорабатывать её, и к 1980-м BSD Unix получает поддержку сетей



Хронология появления Unix-подобных ОС

На графике отображены жизненные циклы ОС, произошедших от Unix, начиная с 1970-х годов



История Linux



2

Эндрю Таненбаум

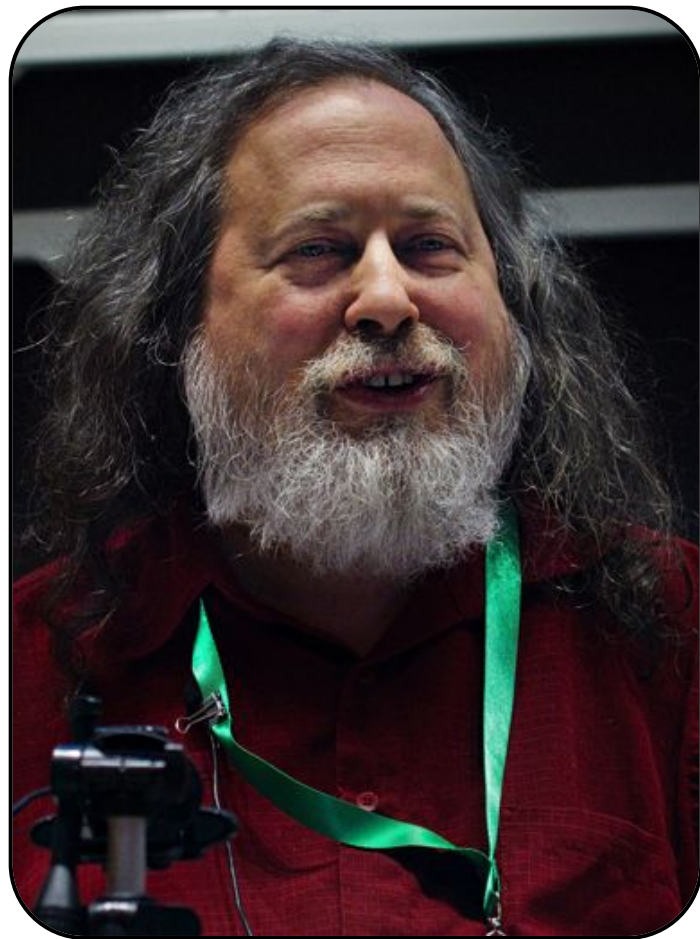
Эндрю Таненбаум (Andrew Tanenbaum, род. 16 марта 1944 г.) — профессор Амстердамского свободного университета, известен как автор ОС Minix (свободная Unix-подобная операционная система для образовательных целей) и книг по компьютерным наукам: «Компьютерные сети», «Современные операционные системы»



Ричард Столлман

Ричард Столлман (Richard Stallman, род. 16 марта 1953 г.) — основатель движения свободного ПО, проекта GNU, Фонда свободных программ и Лиги за свободу программирования.

В рамках проекта GNU программистами из разных стран разрабатывалось ПО, которое впоследствии вошло в GNU/Linux



Проект GNU

Проект GNU (GNU's Not Unix — «GNU — не Unix») — проект по разработке свободного программного обеспечения (СПО).

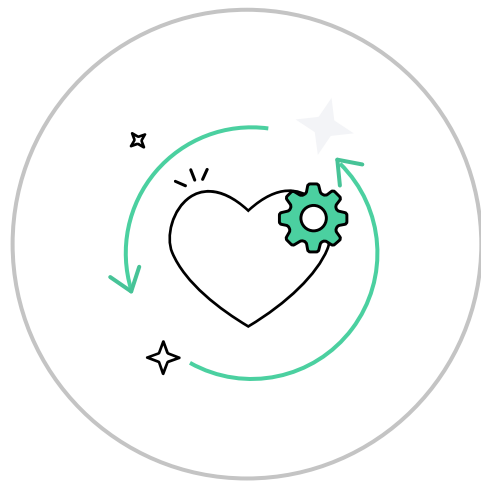
Проект был запущен Ричардом Столлманом в 1983 году



Проект GNU

GNU GPL — это лицензия, согласно которой автор передаёт ПО в общественную собственность.

У проекта GNU есть собственное ядро — Hurd.
В отличие от Linux имеет микроядерную архитектуру



Линус Торвальдс

Линус Торвальдс (Linus Torvalds, род. 28 декабря 1969 г.) — финно-американский программист.

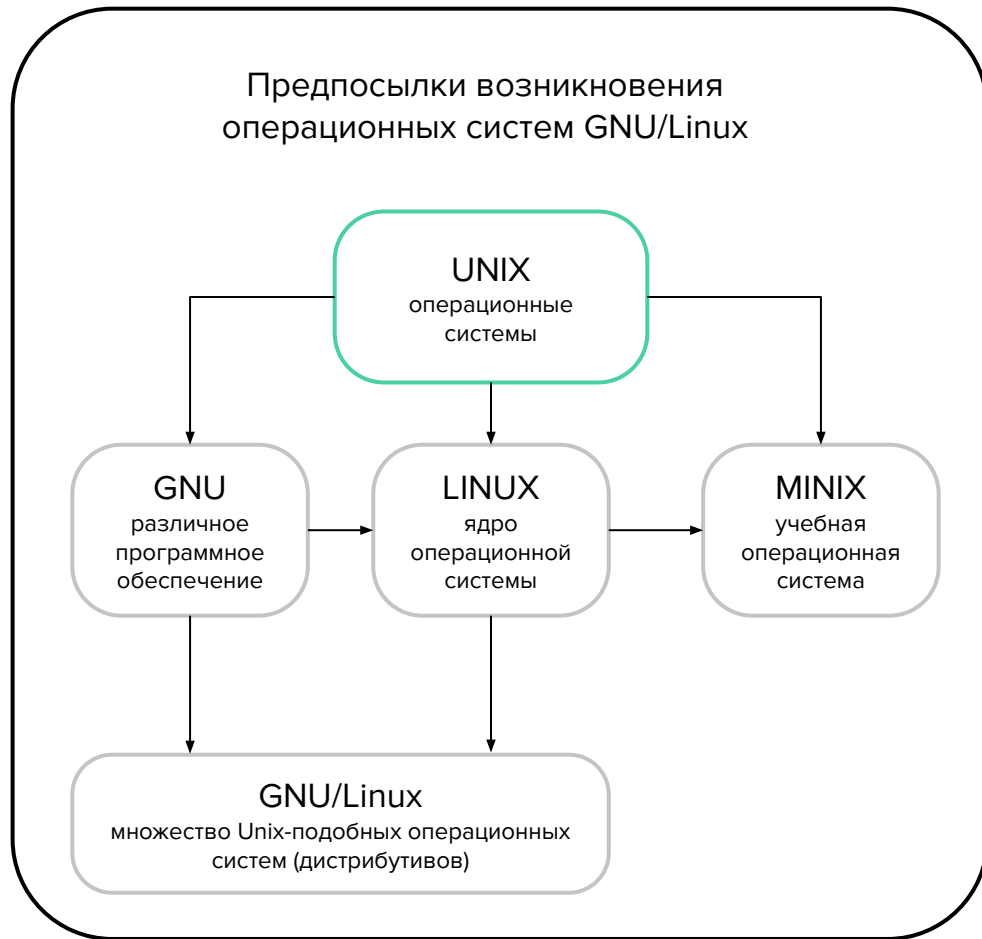
Известен как автор и разработчик Linux — ядра операционной системы GNU/Linux, являющейся сегодня самой распространённой из свободных операционных систем, а также наиболее популярной серверной



Появление Linux

Благодаря проекту GNU, Линус Торвалдс получил возможность **использовать с Linux свободные утилиты**: Bash, компилятор GCC, Tar, Gzip и многие другие уже известные и широко используемые приложения, которые в свою очередь могли работать с его Unix-совместимым ядром

- 1991 г. — начало работы над ядром Linux
- 1994 г. — выходит версия 1.0



OC Linux



1

Linux

Linux — это общее название Unix-подобных операционных систем, основанных на одноимённом ядре.

Особенности:

- открытая и бесплатная ОС
- состоит из большого числа независимо развивающихся компонентов
- применяется гораздо шире на серверах, чем на рабочих станциях
- управляется в большей степени с помощью командной строки и текстовых конфигурационных файлов, в меньшей — с помощью графических инструментов

Область применения Linux

Ядро Linux и системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения.

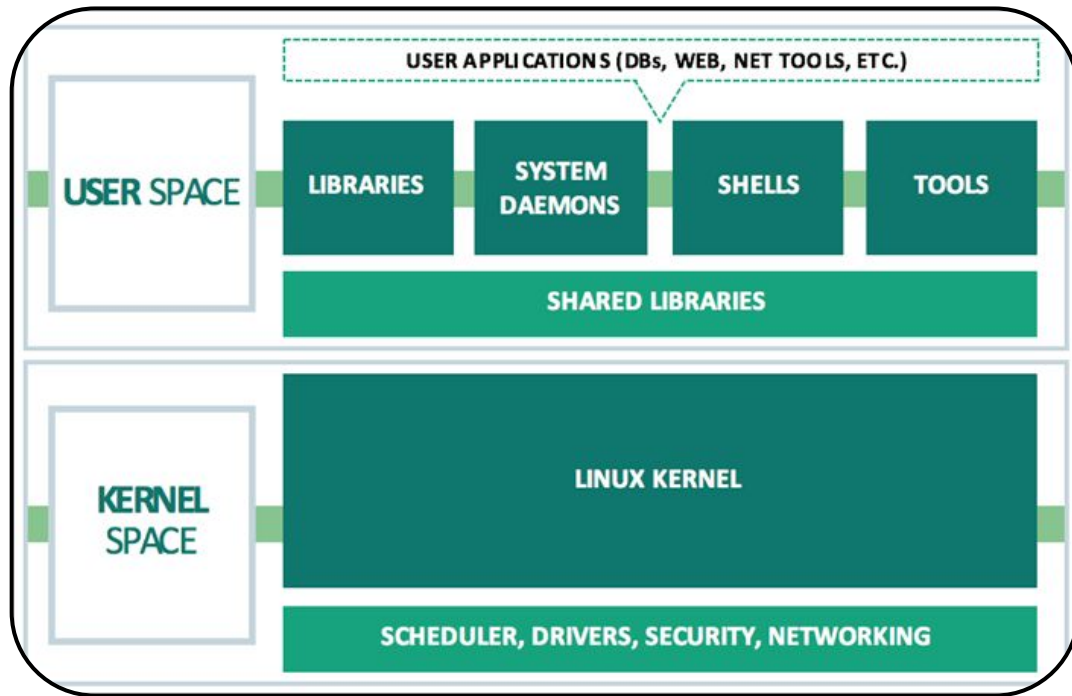
Применяются:

- на серверах
- суперкомпьютерах
- мобильных устройствах
- встраиваемых устройствах
- сетевом оборудовании и др.



Архитектура Linux

- Загрузчик
- Ядро
- Драйверы
- Службы (демоны)
- Оболочка
- Библиотеки
- Утилиты
- Графический сервер



Дистрибутивы

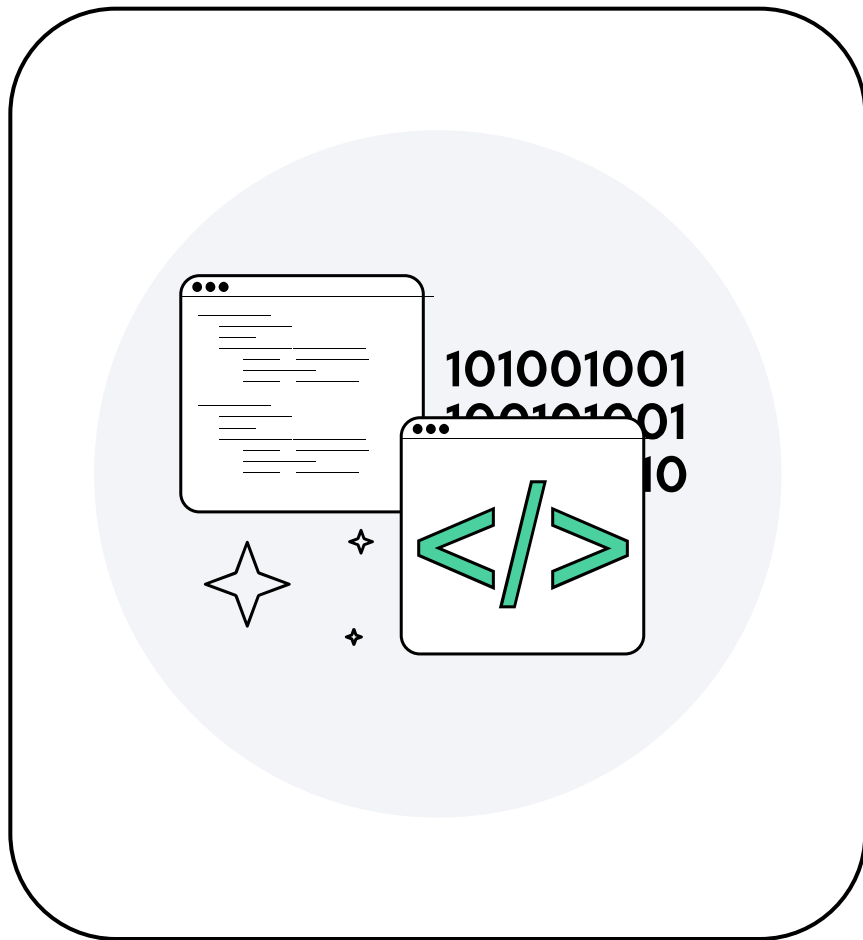


2

Дистрибутивы

Ядро Linux и большинство совместимых с ним приложений поставляются с открытым исходным кодом. Это позволяет создавать собственные системы на их базе.

Такие системы называются **дистрибутивами**



Дистрибутивы

Каждый производитель:

- выбирает определённую версию ядра Linux — при необходимости дорабатывая её для себя: добавляя модули, усиливая безопасность и т. д.
- выбирает систему инициализации: Systemd, OpenRC и т. д.
- компилирует пакеты: сервера, графические приложения, инструменты разработки — и предоставляет инфраструктуру репозитория, включая проверку безопасности
- выпускает патчи и обновления

Дистрибутивы Linux

В зависимости от задач можно выделить следующие дистрибутивы:

- поддержка и гарантии совместимости: с оборудованием, стабильности ПО и т. д.
— **Red Hat, Debian**
- удобство использования для конечного пользователя — **Fedora, Ubuntu, Mint**
- минимализм — **Alpine Linux**
- специальные решения: для маршрутизаторов, коммутаторов и т. п.
- соответствие государственным требованиям и сертификациям — **Astra Linux**,
специальные редакции **ALT Linux**
- пен-тестинг, безопасность — **Kali Linux, BlackArch, Bugtraq, Parrot Linux**
- универсальные, общего назначения — многие из перечисленных

Командная строка



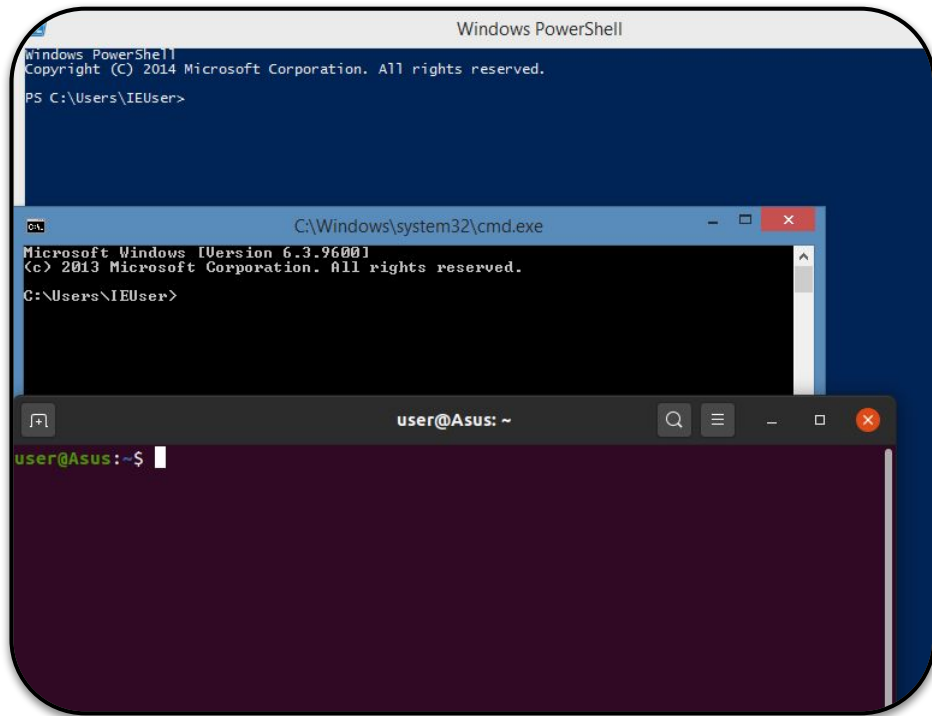
3

Командная строка

Shell — командная оболочка (интерпретатор командной строки).

Обработывает вводимые пользователем команды и файлы сценариев (скрипты).

Bash — де-факто стандартная оболочка Linux и многих других Unix-подобных операционных систем



Почему используем командную строку

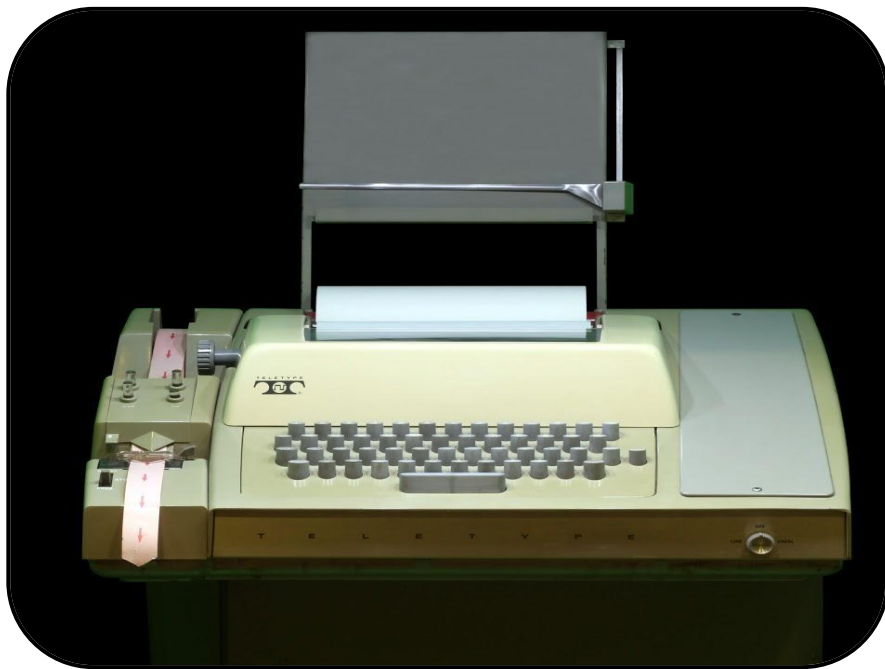
Преимущества:

- широкие возможности для автоматизации
- позволяет передавать результаты работы одной команды в другую
- низкие системные требования
- примитивность: позволяет не усложнять код программ, выполняемых в командной строке

Терминал (teletypewriter, TTY)

Ранний текстовый интерфейс.

Такие использовались для телеграфии задолго до появления компьютеров, однако с появлением компьютеров их стали применять и в этой области



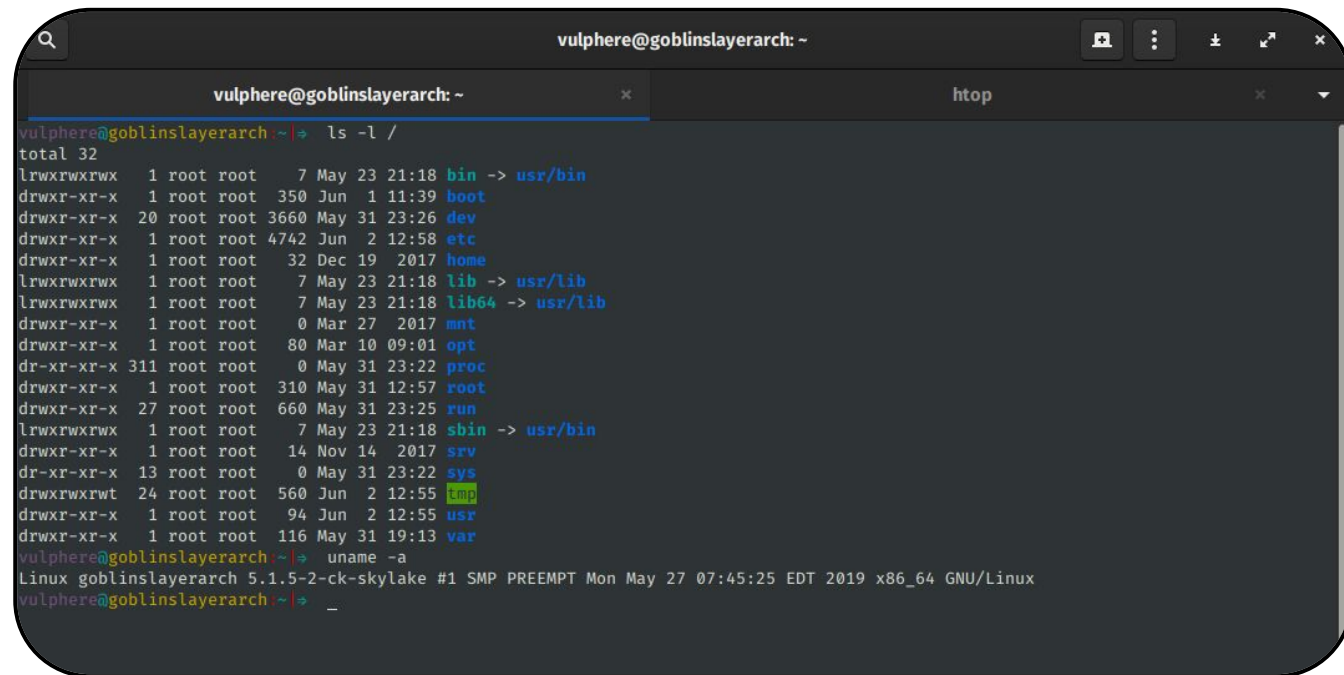
Консоль или графический терминал

По сути всё тот же принтер, но выводит текст на экран. Именно эти устройства и назывались консолями



Эмулятор терминала

То, чем пользователи и администраторы Unix-подобных операционных систем пользуются сегодня для работы с командной строкой



The image shows a terminal window titled "vulphere@goblinslayerarch: ~". The window has a dark theme and a search bar in the top left. The terminal displays the output of the command "ls -l /". The output lists the root directory's contents, including "bin", "boot", "dev", "etc", "home", "lib", "lib64", "mnt", "opt", "proc", "root", "run", "sbin", "srv", "sys", "tmp", "usr", and "var". Each entry shows its permissions, owner, group, size, date, and name. The window also shows a tab titled "htop" and a system status bar at the bottom.

```
vulphere@goblinslayerarch: ~  
vulphere@goblinslayerarch:~$ ls -l /  
total 32  
lrwxrwxrwx 1 root root 7 May 23 21:18 bin -> usr/bin  
drwxr-xr-x 1 root root 350 Jun 1 11:39 boot  
drwxr-xr-x 20 root root 3660 May 31 23:26 dev  
drwxr-xr-x 1 root root 4742 Jun 2 12:58 etc  
drwxr-xr-x 1 root root 32 Dec 19 2017 home  
lrwxrwxrwx 1 root root 7 May 23 21:18 lib -> usr/lib  
lrwxrwxrwx 1 root root 7 May 23 21:18 lib64 -> usr/lib  
drwxr-xr-x 1 root root 0 Mar 27 2017 mnt  
drwxr-xr-x 1 root root 80 Mar 10 09:01 opt  
dr-xr-xr-x 311 root root 0 May 31 23:22 proc  
drwxr-xr-x 1 root root 310 May 31 12:57 root  
drwxr-xr-x 27 root root 660 May 31 23:25 run  
lrwxrwxrwx 1 root root 7 May 23 21:18 sbin -> usr/bin  
drwxr-xr-x 1 root root 14 Nov 14 2017 srv  
dr-xr-xr-x 13 root root 0 May 31 23:22 sys  
drwxrwxrwt 24 root root 560 Jun 2 12:55 tmp  
drwxr-xr-x 1 root root 94 Jun 2 12:55 usr  
drwxr-xr-x 1 root root 116 May 31 19:13 var  
vulphere@goblinslayerarch:~$ uname -a  
Linux goblinslayerarch 5.1.5-2-ck-skylake #1 SMP PREEMPT Mon May 27 07:45:25 EDT 2019 x86_64 GNU/Linux  
vulphere@goblinslayerarch:~$ _
```

Команды в консоли

Каждая команда в консоли может состоять из нескольких частей:

- имя запускаемой команды или утилиты
- параметры запуска — опции, аргументы
- механизмы взаимодействия — перенаправление или каналы

```
ls /var/log
```

```
ls -l /var/log
```

```
ls -l --all /var/log
```

```
ls -la /var/log > result
```

```
ls -la /var/log | grep 'suricata'
```

Автодополнение

Набирать текст команд полностью — прямой путь к подобным ошибкам:

```
command not found
```

```
no such file or directory
```

Достаточно ввести несколько символов и пару раз нажать клавишу **Tab**, чтобы произошло автодополнение команды, имени директории или файла:

```
$ ec<Tab><Tab>  
$ echo
```

Если вхождений несколько, вместо полного дополнения будут предложены варианты:

```
$ cd /etc/sysc<Tab><Tab>  
$ cd /etc/sysctl.  
sysctl.conf  sysctl.d/
```

Запуск эмулятора терминала

- Без графической среды: не требуется — система сразу загружается в текстовую консоль
- В графической среде: можно использовать меню или горячие клавиши. Например, Ctrl + Alt + T в GNOME — среде по умолчанию во многих дистрибутивах

Горячие клавиши

- Остановить выполнение текущей команды: Ctrl + C
- Очистить экран: Ctrl + L (или команда **clear**)
- Выход из консоли: Ctrl + D (команда **exit**)

История команд

- 1 История команд: history
- 2 Ctrl + R, слово для поиска

Стрелки вверх и вниз дают возможность путешествовать по истории команд



Man

Man — инструмент документации (manual) для большинства команд.

Справку по большинству инструментов можно получить с помощью команды:

`man <имя команды>`

Справку по самому man с помощью: `man man`

Навигация:

- j — вниз, k — вверх
- q — выход
- / — поиск, n — следующее совпадение, q — выход

Работа с текстовыми файлами

- Вывести строку **text** на экран: **echo text**
- Вывести содержимое файла **filename**: **cat filename**
- Постраничный просмотр файла **filename**: **less filename**
- Редактировать файл **filename** в терминале: **nano filename**
- Редактировать файл **filename** в GUI: **gedit filename**
- Показать последние 10 строк файла **filename**: **tail filename**
- Поиск подстроки **строка** в файле **filename**: **grep строка filename**

Текстовые редакторы Vim и Nano

Nano — простой в освоении и управлении текстовый редактор, отображает горячие клавиши во время правки и управляется привычным образом

Vim — модальный текстовый редактор. Мощный инструмент, но может быть непростым в освоении. Имеет несколько режимов работы:

- нормальный для навигации по тексту
- визуальный для выделения текста
- вставки для ввода текста
- и др.

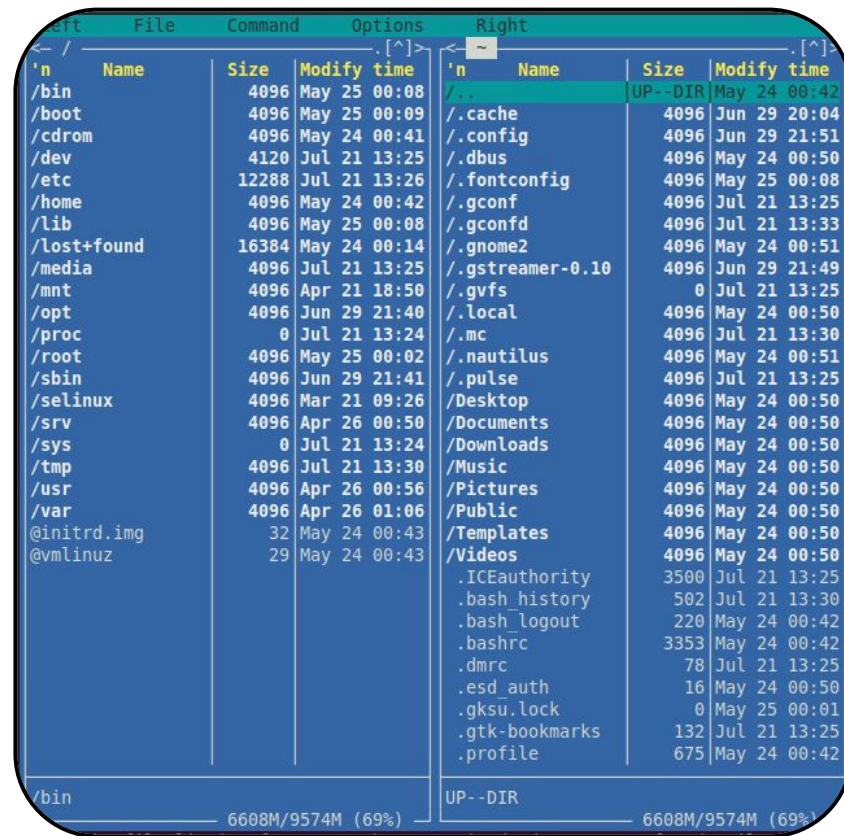
Vi — более компактный и менее производительный предшественник

Vim — входит в стандартную поставку любого дистрибутива Linux

Менеджер файлов Midnight Commander

Двухпанельный файловый менеджер, позволяющий использовать команды консоли. Включает в себя встроенный редактор и просмотрщик текста.

Может быть удобен для управления файлами



Командная строка: перенаправление и конвейер

Потоки ввода-вывода:

- `stdin` — стандартный поток ввода
- `stdout` — стандартный поток вывода
- `stderr` — стандартный поток ошибок

`sort < file.txt` перенаправит **файл** `file.txt` **в** **`stdin`** команды `sort`

`ls > file.txt` перенаправит **`stdout`** команды `ls` **в файл** `file.txt`

`ls 2> file.txt` перенаправит **`stderr`** команды `ls` **в файл** `file.txt`

`>>` **допишет в файл новые строки, в то время как `>` перезапишет его**

`ls | sort` отправит **`stdout`** команды `ls` **в `stdin`** команды `sort`

Пакетный менеджер APT

APT — команда для управления пакетами (установленными программами) в некоторых дистрибутивах ОС Linux (например, Ubuntu и Kali Linux).

Операция управления пакетами требует записи в защищённые директории файловой системы, поэтому требует повышения привилегий (sudo).

Репозитории можно сравнить с Google Play Store или App Store, но поддерживаются они в рамках дистрибутивов и содержат не только программы, но и отдельные библиотеки, документацию, конфигурацию и прочие необходимые для работы программ компоненты

Использование APT

- **sudo apt update** — обновление индекса пакетов: скачивание информации из репозитория о имеющихся версиях
- **sudo apt upgrade** — обновление самих пакетов
- **sudo apt install package** — установка пакета с названием package
- **sudo apt remove package** — удаление пакета package из системы
- **sudo apt autoremove** — удаление неиспользуемых зависимостей

Выключение и перезагрузка

Выключение: **systemctl poweroff** — аналог устаревшего **shutdown -P now**

Перезагрузка: **systemctl reboot** — аналог устаревшего **shutdown -r now**



Пользователи

Основные определения



4

ОС Linux: пользователи

Пользователь — тот, от имени кого запускаются процессы. Не все пользователи используются для входа в систему

Права доступа — атрибуты файла, обозначающие, что могут делать с ним пользователь, группа и все остальные

Группы пользователей — способ объединения пользователей для выдачи общего доступа

Показать имя текущего пользователя: `whoami`

Вывести подробную информацию о пользователе `username: id username`

ОС Linux: пользователи

Пользователи объединяются в группы (например, маркетинг), и права доступа могут выдаваться не конкретным пользователям, а группам.

Владельцем нового файла по умолчанию будет тот, кто создал файл или каталог, а также группа, к которой принадлежит сам владелец.

Посмотреть список групп пользователя username:

```
groups username
```

```
id username
```

ОС Linux: su\sudo

Root — имя суперпользователя (администратора). Разрешать от имени **root** вход в систему не принято, вместо этого используют другие инструменты.

Для выполнения команд, требующих повышенных привилегии, следует ставить перед командой команду `sudo`. В этом случае нужно использовать пароль своего пользователя. Можно также получить доступ к командной строке пользователя **root**: `sudo -i`.

Переключиться на другого пользователя можно с помощью команды `su -`. В этом случае потребуется пароль пользователя **root**

Итоги занятия

Сегодня мы

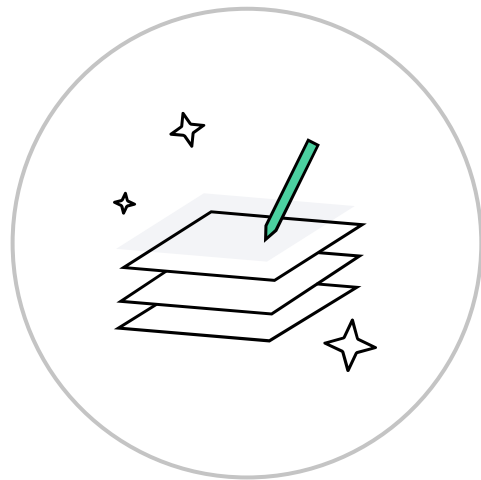
- 1 Познакомились с ОС Linux и историе возникновения
- 2 Узнали, что такое дистрибутивы
- 3 Поработали в командной строке
- 4 Поговорили о пользователях и правах доступа



Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Андрей Тряпичников
Senior Unix engineer в ZFX

