

# Введение в Linux

Юрий Литвинов  
y.litvinov@spbu.ru

28.02.2025

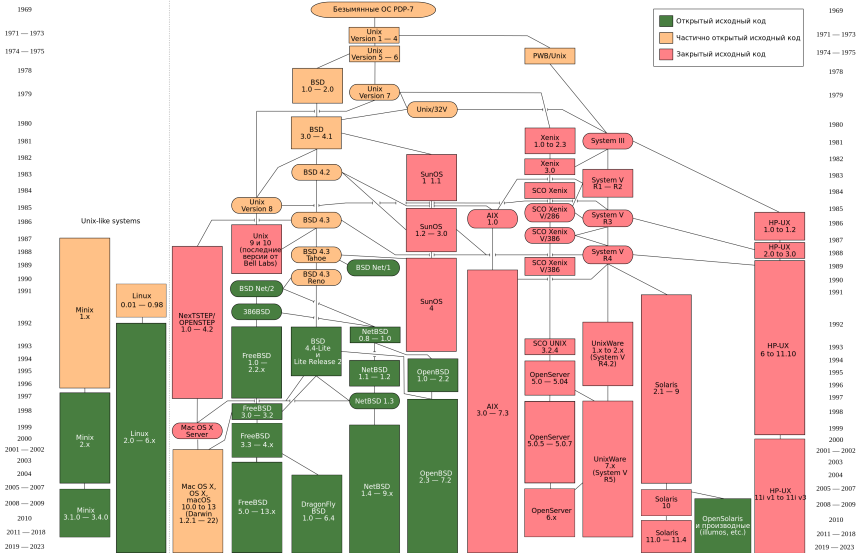
# Linux — что это и зачем?

- ▶ Linux — семейство Unix-подобных операционных систем
  - ▶ Есть сотни дистрибутивов (из которых десятков только широкоизвестных), на базе более-менее общего ядра
  - ▶ <https://github.com/torvalds/linux>
- ▶ С открытым исходным кодом, широкое сообщество разработчиков
- ▶ В основном для серверов, встроенных устройств и всякого странного оборудования
  - ▶ На волне импортозамещения популяризуется и для конечных пользователей
- ▶ Удобнее для программирования
- ▶ Хуже для игр, хуже с поддержкой железа

# Linux для программистов

- ▶ Бесплатна
  - ▶ Поэтому можно ставить на виртуальные машины/контейнеры в любом количестве
  - ▶ В типичном серверном окружении каждый процесс работает в своём контейнере, так что лицензии на другие ОС легко разорят кого угодно
  - ▶ Есть нюанс — целый рынок небесплатных дистрибутивов. Свобода ПО не значит, что на нём нельзя зарабатывать.
- ▶ Очень конфигурируема — от полноценных десктопных дистрибутивов до миниатюрных систем в десятки мегабайт
- ▶ Удобная командная строка
- ▶ Пакетный менеджер
- ▶ Некоторые инструменты есть только для Linux
  - ▶ Valgrind, QEMU, Gem5, ...
  - ▶ Кое-что можно запустить и на Windows, но это нетривиально

## Что такое Linux и зачем он нам



# Отличия от Windows

## С точки зрения пользователя

- ▶ Пакетный менеджер
  - ▶ Как NuGet, только в масштабах всей системы
  - ▶ Часто старые версии
  - ▶ Специфичен для дистрибутива
  - ▶ Может быть несколько репозиториев пакетов
- ▶ Регистрозависимая файловая система
- ▶ Одно дерево файлов, начинающееся в «/», монтирование
- ▶ Много мелких консольных команд, из которых можно собирать сложные штуки
- ▶ Оконный менеджер — обычное внешнее приложение
- ▶ Запускаемое приложение по умолчанию не ищется в текущей папке
- ▶ Человекочитаемые файлы конфигурации

# Дистрибутивы

- ▶ Дистрибутив — это готовое к установке ядро, набор программ и пакетная экосистема
- ▶ Популярные дистрибутивы:
  - ▶ Ubuntu
  - ▶ Mint
  - ▶ Fedora
  - ▶ Debian
  - ▶ Arch
  - ▶ Gentoo
  - ▶ Альт, РОСА, Астра
- ▶ Откуда пошли разные дистрибутивы: [https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux\\_Distribution\\_Timeline.svg](https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)

# Пакеты

- ▶ Пакет — способ распространения приложений и библиотек.
- ▶ Состоит из метаданных, бинарников, конфигов, доков, иногда исходников
- ▶ Дерево зависимостей
- ▶ Популярные форматы пакетов: deb, rpm
- ▶ Базовые утилиты: rpm, dpkg
- ▶ Популярные пакетные менеджеры:
  - ▶ apt (Advanced Packaging Tool) — Debian, Ubuntu, Mint, Альт
  - ▶ yum (Yellowdog Updater, Modified) — старая Fedora и дистрибутивы на её основе
  - ▶ dnf (Dandified YUM) — Fedora, POCA
- ▶ Формат имени пакета:  
<название>-<версия>-<выпуск>.<архитектура>
- ▶ Альтернативный подход: ставить приложение со всеми зависимостями целиком
  - ▶ Flatpak, Snappy, AppImage

# Среды рабочего стола

- ▶ Оконные системы: X Window System, Wayland
- ▶ Среды рабочего стола
  - ▶ GNOME (на GTK+) и его производные (MATE, Cinnamon)
  - ▶ KDE (Plasma) (на Qt)
  - ▶ XFCE (на GTK+)
- ▶ Более-менее исключают друг друга
  - ▶ Приложение для одной среды запустится в другой, но потащит за собой огромную кучу зависимостей и будет выглядеть необычно
- ▶ Дистрибутивы иногда имеют варианты (Ubuntu — GNOME, KUbuntu — KDE)



# Основные консольные команды (1)

- ▶ `cd` — переход в директорию
  - ▶ Абсолютные пути с /
  - ▶ Относительные пути, специальные директории `.` и `..`
- ▶ `pwd` — вывести текущую директорию (обычно в приглашении)
- ▶ `ls` — показать содержимое директории
  - ▶ Ключи `-a`, `-l`
- ▶ `mkdir` — создать директорию
  - ▶ Ключ `-p`
- ▶ `touch` — обновить время последнего обращения к файлу или создать пустой файл
- ▶ `echo` — распечатать что-то в консоль

## Основные консольные команды (2)

- ▶ `cat` — вывести содержимое файла на консоль
- ▶ `cp` — скопировать файл/директорию
- ▶ `mv` — переместить файл/директорию
- ▶ `rm` — удалить файл/директорию
  - ▶ Ключ `-rf`
- ▶ `ln` — создать ссылку
  - ▶ Жёсткие и символичные ссылки (`ln -s`)
- ▶ `sudo` — запустить от имени суперпользователя

## Отвлечение про файловую систему

- ▶ Файловая система суть набор inode (индексный дескриптор)
- ▶ Посмотреть можно `ls -li`
- ▶ Режим доступа: Read, Write, Execute, для владельца, группы, всех
  - ▶ `chmod`
- ▶ Имена файлов живут отдельно от inode-ов
- ▶ Что-то вроде сборки мусора — если ни одно имя не указывает на inode, его удаляют
- ▶ Символьная ссылка — это путь к файлу, а не inode
- ▶ Файлы вовсе не обязательно лежат на диске
  - ▶ Всё, что поддерживает потоковый ввод-вывод — файл!
  - ▶ `/dev/null`, `/dev/random`, `/proc/...`

# Filesystem Hierarchy Standard

- ▶ Бывают изменяемые и статичные файлы, и разделяемые и неразделяемые
  - ▶ Разные условия доступа по сети и бэкапов
  - ▶ Следовательно, должны лежать в разных директориях
- ▶ Корневая файловая система — только самое необходимое для загрузки и восстановления системы, к ней подмонтируются другие директории
  - ▶ Никто не мешает им быть на одном разделе с корневой
  - ▶ Могут быть на других дисках или вообще в сети
- ▶ Всё не критичное для системы барахло — в /usr
- ▶ Запрещено создавать файлы и директории в корне

## Стандартные директории

- ▶ /bin — нужные программы (в т.ч. для пользователя)
- ▶ /boot — файлы загрузчика
- ▶ /dev — файлы устройств (они не лежат на диске, если что)
- ▶ /etc — конфиги (не разделяемые)
- ▶ /lib — библиотеки и модули ядра
- ▶ /media — сюда монтируются внешние носители типа флешек
- ▶ /mnt — директория для временного монтирования
- ▶ /opt — директория для дополнительных пакетов (обычно проприетарных)
- ▶ /sbin — нужные программы (для администратора)
- ▶ /tmp — временные файлы
- ▶ /usr — вторичная иерархия (примерно та же иерархия, но из некритичных файлов)
  - ▶ Сейчас популярен «joined root», когда bin -> /usr/bin, а /sbin -> /usr/sbin
- ▶ /var — изменяемые данные
- ▶ ...

## Немного более продвинутая работа с консолью

- ▶ wildcards (globs) — `file[1-8]`, `file*`, есть ещё `?`, `[!]`
  - ▶ Это те самые глобы из `.gitignore`
  - ▶ Они раскрываются до вызова команды
- ▶ Сильное и слабое кватирование
- ▶ Переменные окружения: `$PATH`, `export`
- ▶ `which` — найти программу в путях, `whereis` — найти все её файлы
- ▶ `find`: `find /usr/share/doc -name README`
- ▶ `locate` — ищет и пути

# Управление процессами

- ▶ Убить процесс — Ctrl-C
- ▶ Остановить — Ctrl-Z (как бы ставит на паузу)
- ▶ fg и bg, оператор &
- ▶ kill, killall — послать сигнал процессу
  - ▶ SIGINT, SIGTERM, SIGKILL
- ▶ ps, top
- ▶ Перенаправление: >, >>, |
  - ▶ echo lol > file.txt
  - ▶ echo lol | wc
  - ▶ cat file.txt | sort | uniq | wc -l

# Работа с текстом

- ▶ sort, uniq, wc
- ▶ head, tail
- ▶ more, less
- ▶ sed, awk
- ▶ vim



# Полезные штуки

- ▶ Табуляция
- ▶ `.bashrc`, `.bash_profile`, `alias`
- ▶ Midnight Commander
- ▶ `reverse-i-search` (Ctrl-R, Ctrl-S)
- ▶ Ctrl-W, Ctrl-U
- ▶ Выделение мышью, вставка средней кнопкой
- ▶ Клавиша Compose
- ▶ Ctrl-Alt-F1, Ctrl-Alt-F2 и т.д., Ctrl-Alt-Backspace
- ▶ Всегда имеет смысл ставить проприетарные драйвера для видеокарты, и они почти никогда не ставятся по умолчанию

# Документация

- ▶ Стандартизована, в целом лучше поддерживается, чем для типичных Windows-программ
- ▶ `man`
  - ▶ RTFM!
  - ▶ Секции `man`, 9 разделов, 1 — пользовательские программы, 2 — системные вызовы и т.д.
- ▶ `whatis`
- ▶ `apropos`
- ▶ `info` — более продвинутый формат документации, с гиперссылками
- ▶ Документация дистрибутива