PEGINELAB

Karate Linux

Практический путь от новичка до мастера

Книга 1

Белый пояс

Серия практических книг и курсов



Белый пояс

Новичок

Основы | Практика | Углубление | Эксперт | Мастер

Освой универсальные принципы Linux и выйди на уровень мастерства

Karate Linux. Книга 1. Белый пояс.
Практический путь от новичка до мастера. — Авторское электронное издание
PEGINELAB, 2026. — Ил.

Греп Пайп (Grep Pipe)

Авторский проект и редакция: PEGINELAB
Серия: Karate Linux. Практические книги и курсы
Книга 1: Белый пояс, 259 с
Год: 2026

Авторские права
© Греп Пайп (Grep Pipe), 2026.

Авторский текст, оформление, иллюстрации и оригинальные учебные материалы охраняются законодательством об авторском праве с учётом прав третьих лиц и применимых лицензий.

Linux, Ubuntu, Astra Linux и другие наименования продуктов используются для их технической идентификации. Права на соответствующие товарные знаки принадлежат их правообладателям.

Издание является независимым авторским проектом и не представляет собой официальную документацию разработчиков упомянутых операционных систем.

Краткое содержание

КНИГА 1 / БЕЛЫЙ ПОЯС

	Предисловие	
I	НАЧАЛО ПУТИ	
1	Основы Linux: история и семейства ОС	6
2	Командная строка	16
3	Документация и пакеты	24
II	УСТАНОВКА И ФАЙЛЫ	
4	Установка Ubuntu	35
5	Astra Linux: практика и защита	49
6	UEFI, GPT и файловые системы	62
7	Каталоги, pwd, ls и tree	73
8	Файлы и ссылки	85
III	ДОСТУП И ИНСТРУМЕНТЫ	
9	Пользователи и группы	103
10	Права доступа, chmod и ACL	120
11	Текстовые редакторы	141
12	Поиск, фильтры и вывод	161
13	Пакеты и обновления	181
IV	ПЕРВЫЕ ПРОВЕРКИ	
14	Процессы и службы	201
15	Сеть и удалённый доступ	220
16	Экзамен и лабораторная работа	238

Предисловие

Эта книга начинается не с таинственной команды и не с перечня сотен параметров. Она начинается с простого вопроса: почему один человек управляет компьютером уверенно, а другой боится нажать Enter? Разница чаще не в памяти, а в понимании того, что находится перед нами: где выполняется команда, под чьим именем, что произойдёт с файлом и как проверить результат.

В боевых искусствах новичок сначала изучает стойку, равновесие и дисциплину. Сложный приём бесполезен, если не понимаешь собственное движение. Таким же будет наше «Карате Linux»: знания мы разобьём на пояса, а учебные задачи — на короткие повторяемые приёмы. Не требуется знать всё. Требуется научиться проверять то, что знаешь.

Белый пояс — первая книга серии. Мы знакомимся с историей операционных систем, различаем ядро, дистрибутив, терминал и командную оболочку. Затем создаём безопасную учебную машину и учимся работать с ней без поспешных действий. Наиболее подробно будем использовать Ubuntu; Astra Linux станет важным сравнительным направлением. В следующих томах будем повышать сложность, переходя к безопасности, службам, сетям и автоматизации.

Эта работа сознательно не превращается в энциклопедию на тысячу страниц. Первый том — самостоятельный практический курс: объяснения чередуются с заданиями, схемами и проверкой результатов. Научное утверждение будет отделяться от гипотезы, инструкция — от непроверенного сценария, а юмор — от настоящих правил безопасности.

Важный принцип: не копируйте опасные команды, пока не понимаете их последствий. Создайте виртуальную машину,

экспериментируйте с пробными каталогами, делайте снимки состояния и оставляйте себе дорогу назад. Это книга о контроле над системой, а не о риске ради эффектного результата.

«Незнание — это путь. Знание — это сияние».

Греп Пайп · PEGINELAB

Для кого эта книга

Новичку книга поможет установить первую Linux-систему и не испугаться терминала. Разработчик увидит, как находить пути к файлам, проверять права и читать справку. Пользователь Windows или macOS поймёт, какие принципы универсальны и почему некоторые команды отличаются. Начинающий администратор сможет проверить фундамент, на котором позже строятся серверные настройки.

- Не требуется предыдущий опыт в Linux; нужны навыки обычной работы на компьютере.
- Для лабораторных работ желательна виртуальная машина, не содержащая личных данных.
- Для сравнения предлагаются Ubuntu, Astra Linux, Fedora, CentOS Stream, FreeBSD и Oracle Solaris; исторически упоминаются OS/2, Windows и macOS.
- Практические сценарии в основном относятся к Ubuntu 24.04 LTS; последовательность экранов может отличаться в других выпусках.

Как читать и проверять

Каждый приём включает объяснение, пример и проверку. Команды выделены в отдельные блоки: знак `$` в приглашении вводить не нужно. Перед действиями с административными правами внимательно прочитайте предупреждения и проверьте результат на учебной машине.

PEGINELAB // КАРАТЕ LINUX // БЕЛЫЙ ПОЯС

Глава 1

Основы Linux



В ЭТОЙ ГЛАВЕ

Узнаём, что именно называется операционной системой, почему Linux не равен UNIX и как возникли разные семейства.

«Понимание начинает путь. Проверка делает его уверенным».

ГЛАВА 1 / ОСНОВЫ LINUX

1.1. Компьютер и операционная система

Компьютер состоит из процессора, памяти, накопителей, сетевых устройств и множества контроллеров. Но прикладная программа не должна самостоятельно договариваться с каждым устройством. Операционная система создаёт общие правила: выделяет память, запускает процессы, управляет файлами, устройствами и доступом пользователей. Поэтому один редактор текста можно запускать на разных компьютерах, если для них существует **подходящая** версия программы.

Слово «Linux» строго обозначает ядро — центральную часть системы. На обычном компьютере к ядру добавляют системные утилиты, библиотеки, программу установки, диспетчер пакетов, настройки, службы и иногда графический рабочий стол. Такой комплект называют дистрибутивом Linux. Ubuntu и Astra Linux

— разные дистрибутивы, использующие ядро Linux; FreeBSD — самостоятельная операционная система с другим ядром.

Различие полезно помнить при поиске ответа. У проблемы может быть несколько уровней: сбой ядра, ошибка службы, неверные настройки пакета, запрет файловой системы или обычная опечатка в команде. Если мы не различаем слои, то и диагностика превращается в лотерею.

1.2. Короткая история: основные даты

Год	Система или событие	Что важно запомнить
1969	UNIX	Начало разработки в Bell Labs; предок множества систем и идей.
1984	Macintosh	Появление оригинального компьютера и классической Mac OS.
1985	Windows 1.0	Начало линейки графической среды Microsoft Windows.
1987	IBM/Microsoft OS/2	Отдельная ветвь для персональных компьютеров, не UNIX.
1991	Ядро Linux	Линус Торвальдс начал публичное развитие ядра.
1992	Solaris	Sun представила семейство Solaris; ныне существует Oracle Solaris.
1993	FreeBSD и Debian	Независимые проекты: BSD-система и Linux-дистрибутив.
2001	Mac OS X	Современная UNIX-основа системы Apple.
2003–2004	Fedora, Ubuntu	Развитие двух важных направлений Linux.

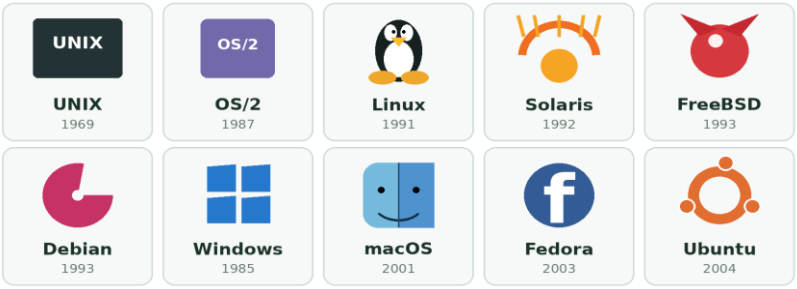


Рис. 1. Авторские учебные пиктограммы и годы начала семейств. Это не официальные логотипы производителей.

1.2.1. Откуда взялся UNIX

В конце 1960-х инженеры Bell Labs искали удобную среду для разработки программ. Так возник UNIX: сравнительно компактная система с иерархической файловой структурой и утилитами, которые удобно соединять друг с другом. Идея «маленьких инструментов, выполняющих понятную задачу» живёт до сих пор в командах, перенаправлениях и конвейерах.

Важным поворотом стало распространение UNIX в университетах и появление семейства BSD. Эти разработки не были Linux, но повлияли на способы работы, документацию и программные интерфейсы. Отдельная линия развития привела к коммерческим UNIX-системам, включая Solaris.

1.2.2. Кто такой Линус Торвальдс



Линус Торвальдс — финский программист, который в 1991 году, будучи студентом Хельсинкского университета, начал создавать собственное ядро UNIX-подобной системы для персонального компьютера. 25 августа он сообщил о проекте в группе comp.os.minix. Работа постепенно превратилась в международную разработку, к которой смогли подключаться независимые программисты.

Торвальдс не придумал все программы вокруг Linux и не создал UNIX. Важна комбинация его ядра с уже существовавшими системными утилитами и свободной моделью разработки. Потому выражение GNU/Linux можно встретить там, где подчёркивают вклад инструментов GNU. Одновременно есть системы на ядре Linux, которые не используют классический набор GNU как основной пользовательский интерфейс.

1.2.3. Разница между «открытым» и «бесплатным»

Открытый исходный код не означает, что каждое программное изделие всегда бесплатное, не имеет правообладателя или может применяться без условий. Лицензия определяет права на изучение, изменение и распространение. У одной системы могут быть открытое ядро, закрытые драйверы, платная поддержка и отдельные условия для корпоративной эксплуатации. Поэтому перед установкой или распространением образа проверяют редакцию, лицензию и источник получения.

1.3. Современные семейства: что общего, что разного

Ubuntu и Debian используют пакеты формата DEB и инструменты APT, но отличаются политикой выпусков, настройками и составом пакетов. Astra Linux опирается на Linux и инструменты Debian-подобной экосистемы, добавляя собственные компоненты и механизмы обеспечения защиты. Fedora и CentOS Stream используют RPM и DNF, но имеют разные цели: Fedora быстро внедряет новшества, а CentOS Stream связан с разработкой корпоративного направления RHEL.

FreeBSD имеет собственное ядро, базовую систему и механизм pkg для дополнительных программ. Oracle Solaris относится к традиции UNIX и использует собственные технологии, например IPS для пакетов. macOS имеет UNIX-основу и собственные компоненты Apple. OS/2 и семейство Windows демонстрируют альтернативные пути развития настольных операционных систем. PowerShell при этом не ОС, а оболочка и язык автоматизации, который может использоваться на разных платформах.

Среда	Семейство	Пакеты / особенности
Ubuntu / Debian	Linux, DEB	apt, dpkg, systemd в типичных современных выпусках
Astra Linux	Linux, Debian-подобная	apt, собственные компоненты защиты и редакции
Fedora / CentOS Stream	Linux, RPM	dnf, rpm, systemd; SELinux во многих установках
FreeBSD	BSD	pkg, ports; система и ядро развиваются вместе
Oracle Solaris	UNIX	IPS: pkg; SMF и другие собственные средства
Windows / OS/2	отдельные семейства	собственные механизмы и средства установки

1.4. Мини-глава: другие ОС — архитектура и предназначение

В одинаково выглядящих окнах могут работать системы с совершенно разным происхождением. Полезно различать три вопроса: на каком ядре построена ОС, что входит в её базовую поставку и для каких задач её обычно выбирают. Одинаковые команды вроде `ls` или `cd` ещё не делают две системы одной архитектурой.

Здесь приведены короткие технические портреты. Годы относятся к появлению конкретной системы или важного выпуска, а не ко всем её будущим версиям. Например, Macintosh 1984 года и macOS XXI века — не одно и то же семейство ядер, хотя их объединяет преемственность пользовательской платформы Apple.

1.4.1. UNIX — начало традиции: 1969

В Bell Labs система UNIX выстроила модель процессов, прав, дерева каталогов и переносимых утилит. Она стала источником технических интерфейсов и идей. Сегодня слово UNIX может

также обозначать системы, прошедшие соответствующую сертификацию; поэтому не всякая UNIX-подобная система юридически является UNIX.

UNIX оказал влияние на серверные ОС, научные вычисления, учебные лаборатории и разработку программ. Его главное наследие для читателя — принципы и переносимые инструменты.

1.4.2. IBM OS/2 — самостоятельная линия: 1987

IBM и Microsoft начали совместную ОС для ПК, рассчитанную на многозадачность и развитие платформы персональных компьютеров. OS/2 не является ни дистрибутивом Linux, ни разновидностью UNIX. В поздних выпусках у неё сформировалась собственная графическая среда и экосистема программ.

Исторически OS/2 использовали в офисах, банках, терминалах и специализированных рабочих местах. Эта система полезна для сравнения путей развития ПК, а не как обязательная учебная платформа для установки сегодня.

1.4.3. Windows — от графической среды к NT: 1985 и 1993

Windows 1.0 появилась в 1985 году как графическая среда поверх MS-DOS. В 1993 году вышла Windows NT — самостоятельная архитектурная линия с собственным ядром, моделью объектов, процессами и подсистемами. Современная Windows развивается на основе семейства NT, а не на основе UNIX.

Основные сферы применения — настольные системы, рабочие места организаций, игры, разработка и серверные задачи. В PowerShell команды возвращают объекты, и это важно отличать от распространённой UNIX-практики потоков текста.

1.4.4. Solaris — корпоративная UNIX-линия: 1992

Solaris выросла из системы Sun Microsystems и семейства SunOS, а Solaris 2 опиралась на UNIX System V Release 4. В современной линейке Oracle Solaris выделяются технологии ZFS, DTrace, Zones, SMF и IPS. Похожие имена утилит не делают пакетный менеджер Solaris идентичным менеджеру FreeBSD.

Solaris интересна для изучения корпоративной UNIX-истории, отказоустойчивых приложений, хранения и наблюдаемости. В книге это сравнительная, а не основная практическая ОС.

1.4.5. FreeBSD — отдельная система семейства BSD: 1993

FreeBSD развивает ядро и базовую пользовательскую систему как согласованный проект. Она наследует идеи BSD и имеет собственные механизмы управления системой, jails, сетевую подсистему, файловые системы UFS и ZFS. Для сторонних программ используются пакеты `pkg` и порты.

FreeBSD широко известна в сетевых сервисах, серверах, хранилищах и специализированных устройствах. Она UNIX-подобна, но не использует ядро Linux.

1.4.6. Macintosh и macOS — два этапа Apple: 1984 и 2001

Первый Macintosh вышел в 1984 году с собственной классической системной программной средой. Mac OS X, выпущенная в 2001 году, построена на другой основе: Darwin, XNU и UNIX/BSD-компонентах. Следовательно, переносить команды, схемы драйверов и механизмы защиты со старой Mac OS на современную macOS неправильно.

Эта линия востребована в творческой работе, программировании, офисных задачах и экосистеме Apple. За привычным графическим интерфейсом скрываются терминал, файловые разрешения и дополнительные политики безопасности.

1.4.7. Linux — ядро и многие дистрибутивы: 1991

Linux объединяет семейство дистрибутивов вокруг ядра с модульной монолитной архитектурой. Пакеты, рабочий стол, системные службы и политика безопасности отличаются от дистрибутива к дистрибутиву. Поэтому Ubuntu, Astra Linux, Fedora и CentOS Stream следует изучать как разные сборки и варианты эксплуатации, а не как разные ядра по происхождению.

Linux используют на серверах, рабочих станциях, в контейнерах, робототехнике, сетевом оборудовании и встроенных устройствах. Именно здесь сосредоточена основная практика серии «Карате Linux».

ОС НЕ РАВНА ОБОЛОЧКЕ

Bash, zsh и PowerShell — интерфейсы и языки управления; Ubuntu, FreeBSD, Solaris и Windows — операционные системы; apt, dnf, pkg — средства управления пакетами, а не названия ОС.

ДЛЯ ДАЛЬНЕЙШЕГО ЧТЕНИЯ

К. Негус, «Библия Linux», 10-е изд. (2022): начало главы 1 — с. 36; история и происхождение Linux — ориентировочно с. 40–50. Источник используется для сопоставления тем, а не перепечатывается.

1.5. Семейства и архитектуры: не путать происхождение с внешним видом

UNIX — не одна современная программа, а историческое семейство и набор идей: процессы, единое дерево файлов, ввод-вывод и переносимые интерфейсы. Подлинное право называться UNIX зависит также от сертификации конкретного продукта. Linux и FreeBSD — UNIX-подобные системы, но не версии Linux друг друга.

Linux использует модульное монолитное ядро: работа многих подсистем происходит в привилегированном пространстве ядра, а драйверы могут загружаться модулями. Ubuntu, Debian, Fedora и Astra Linux собирают вокруг этого ядра разный набор

пользовательских инструментов, служб, политик и пакетов. Поэтому нельзя судить об ОС только по значку терминала.

FreeBSD — самостоятельная ОС с собственным ядром, согласованной базовой системой и отдельной коллекцией пакетов и портов. Solaris — коммерческая UNIX-линия со своим ядром, средствами управления службами SMF, файловой системой ZFS и инструментами наблюдения. macOS основана на Darwin и ядре XNU, которое сочетает компоненты Mach и BSD; она не является Linux.

Windows NT развивала собственную гибридную архитектуру ядра и модель безопасности с токенами, объектами и ACL; PowerShell — лишь одна из доступных оболочек. OS/2 — самостоятельная историческая многозадачная платформа персональных компьютеров. Совпадающие команды и каталоги не доказывают одинаковую внутреннюю архитектуру.

1.5.1. Почему нельзя просто перенести команды

К примеру, просмотр справки `man ls` характерен для UNIX-подобных систем; в Windows PowerShell стандартный способ изучить командлет — `Get-Help Get-ChildItem`. В FreeBSD можно использовать `pkg`, в Ubuntu — `apt`, в Fedora — `dnf`, в Solaris — системный `pkg` (IPS). При чтении чужой инструкции всегда выясняйте систему, версию, используемую оболочку и права пользователя.

ЛАБОРАТОРНАЯ РАБОТА 01 / Различаем систему и оболочку

ЦЕЛЬ

Составить краткую карточку своей текущей ОС, не устанавливая ничего нового.

1. Определите, какая ОС запущена на компьютере.
2. Назовите ядро, графическую оболочку и терминал, если они есть.
3. Запишите одну особенность лицензирования и один официальный источник документации.

ПРОВЕРКА РЕЗУЛЬТАТА

В карточке ясно различаются ОС, ядро, терминал и оболочка. Системы семейства BSD не называются дистрибутивами Linux.

ФИНИШНЫЙ ПРИЁМ / МИКРО-КАТА ИНЖЕНЕРА

Микро-ката: первая стойка — киба-дачи

Короткая разминка без растяжки и резких ударов. Выполняйте движения плавно, рядом с опорой; при боли или головокружении остановитесь. Это перерыв от сидения, а не лечение или обещание улучшения памяти.



1. Исходная
Стопы на ширине плеч; вдох 3–4 секунды.



2. Шаг в стороны
Разведите стопы чуть шире плеч без рывка.



3. Киба-дачи
Слегка согните колени; держите корпус прямо 5–10 секунд.

Киба-дачи — «стойка всадника». Для короткой разминки используйте лишь неглубокий вариант: стопы слегка шире плеч, носки направлены вперёд или чуть наружу, колени смотрят туда же, куда носки. Без боли и задержки дыхания. Если трудно удерживать равновесие, встаньте рядом с устойчивой опорой. Повторите два раза по 5–10 секунд.

МЫСЛЬ СЕНСЕЯ

«Начни с понимания. Сила придёт вместе с проверкой».

МИНИ-СЛОВАРЬ / ГЛАВА 1

ОС	Среда, управляющая аппаратными ресурсами и запуском программ.
Ядро	Компонент ОС, работающий с процессами, памятью, устройствами.
Дистрибутив	Готовая система вокруг ядра: пакеты, настройки и инструменты.
UNIX-подобная система	Разделяет принципы и интерфейсы UNIX, но не обязательно сертифицирована.

PEGINELAB // КАРАТЕ LINUX // БЕЛЫЙ ПОЯС

Глава 2

Командная строка



В ЭТОЙ ГЛАВЕ

Понимаем происхождение командной строки и путь команды от клавиатуры до работы ядра; различаем терминал, shell и выполняемую программу.

«Понимание начинает путь. Проверка делает его уверенным».

ГЛАВА 2 / КОМАНДНАЯ СТРОКА

2.1. Почему командная строка старше привычных окон

Первые компьютеры не имели знакомого нам рабочего стола. Оператор передавал задания через перфокарты, позже — с помощью текстовых терминалов и телетайпов. Поэтому текстовое взаимодействие появилось задолго до массового использования мыши. Его достоинство заключалось не в красоте, а в повторяемости: короткую последовательность команд можно записать, проверить, объяснить другому человеку и выполнить снова.

С развитием UNIX текстовые терминалы стали способом работы многих пользователей с одним сервером. Приложение отправляло текст на терминальное устройство и принимало

ввод. Сейчас физический телетайп обычно заменяет терминальный эмулятор — окно программы, имитирующей текстовую консоль. Но основные идеи сохраняются: поток ввода, поток вывода, сообщения об ошибках, управляющие сочетания клавиш.

2.2. Терминал, оболочка и ядро — не одно и то же

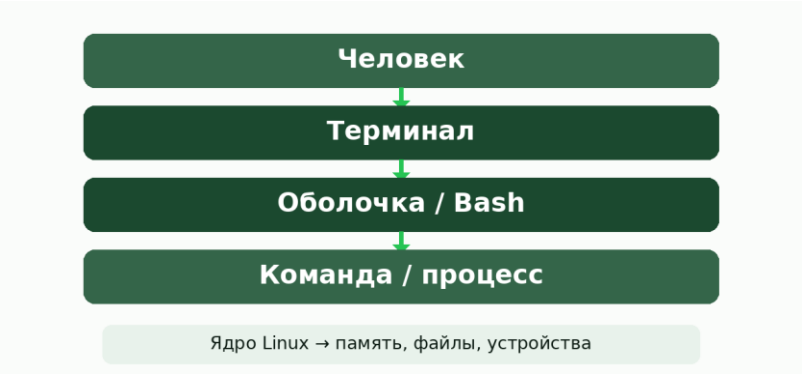


Рис. 2. Упрощённая цепочка: человек — окно терминала — оболочка — команда — системные вызовы ядра.

Терминал показывает символы, принимает нажатия клавиш и позволяет взаимодействовать с оболочкой. Bash — один из командных интерпретаторов: он читает введённую строку, выполняет подстановки и встроенные команды, а для внешних программ запускает процессы. Выполняемая программа затем обращается к ядру через системные интерфейсы, когда ей требуется прочитать файл, выделить память или работать с устройством.

Из этого следует практический вывод. Если программа `ls` отсутствует, ошибка не обязательно связана с ядром. Если Bash не находит команду, следует проверить имя и путь поиска (`PATH`). Если программа найдена, но не открывает файл, нужно проверить права и существование каталога. Эти три ситуации похожи на экране, но требуют разных действий.

2.2.1. Что такое prompt

Приглашение командной строки — строка или набор символов, которые оболочка показывает перед вводом. Она может содержать имя пользователя, имя компьютера и текущий каталог. Типовой вид `student@ubuntu:~$` означает лишь пример отображения. Символ `$` обычно подсказывает, что работа ведётся от непривилегированного пользователя; отдельные оболочки и настройки могут показывать другие обозначения.

В книге команды печатаются без лишней магии: сначала проверяем, кто мы, затем где находимся, затем что именно просим выполнить.

Первые наблюдения

```
whoami
pwd
command -v ls
echo "$SHELL"
```

`whoami` выводит имя пользователя, `pwd` — текущий каталог, `command -v ls` помогает определить расположение или тип найденной команды. Переменная `SHELL` обычно указывает настроенную оболочку входа, но не всегда доказывает, какую оболочку процесс выполняет прямо сейчас. Для текущей сессии полезно посмотреть `ps -p $$ -o comm=` в Bash.

2.3. Встроенная команда и внешняя программа

Часть действий Bash выполняет без запуска отдельного исполняемого файла. Например, `cd` меняет рабочий каталог самой оболочки, поэтому обычно является встроенной командой. А `ls` чаще запускается как отдельная программа. Чтобы выяснить разницу, Bash предоставляет `type` и `help`. Команда `type cd` расскажет о встроенном механизме, а `type ls` — о найденном исполняемом файле или псевдониме.

Не нужно запоминать набор команд наизусть. Намного полезнее научиться задавать системе вопросы: «Что сейчас

запущено?», «Где находится программа?», «Какие аргументы она принимает?» и «Какие права нужны для операции?».

Четыре вопроса оболочке

```
type cd
type ls
help cd
printf "%s\n" "$PATH"
```

2.4. Как читается команда

Строка `ls -la /home` состоит из имени программы, параметров и аргумента. `ls` — команда; `-l` и `-a` — параметры, влияющие на её поведение; `/home` — путь к каталогу, над которым выполняется действие. Запись `-la` объединяет два коротких параметра, если утилита это допускает. Однако не все программы используют одинаковые правила, поэтому справка важнее догадок.

Кавычки нужны, чтобы оболочка не разделяла имена с пробелами на разные аргументы. Сравни `mkdir Документы проекта` и `mkdir "Документы проекта"`: в первом случае получится два имени, во втором — одно. На начальном этапе научимся помещать переменные и пути с пробелами в кавычки, когда это нужно.

Безопасный пример с каталогом

```
ls -la /home
mkdir -p "$HOME/Учебный проект"
ls -ld "$HOME/Учебный проект"
```

2.4.1. Команда, вывод и код завершения

У процесса обычно есть три стандартных потока: stdin (ввод), stdout (нормальный вывод) и stderr (сообщения об ошибках). Кроме текста процесс возвращает код завершения. По соглашению код 0 обычно означает успех, а ненулевое значение — определённую ошибку или условие. В Bash его можно увидеть через `$?`, но важно сделать это сразу после интересующей команды: следующая команда изменит значение.

На практике это позволяет отличать «ничего не напечатано» от «всё успешно». Некоторые команды молчат при успехе, а некоторые печатают сообщения даже при нормальной работе.

Учимся читать код завершения

```
pwd
printf "Код: %s\n" "$?"
ls /каталог-которого-нет
printf "Код: %s\n" "$?"
```

2.5. История, автодополнение и остановка

Стрелка вверх обычно возвращает предыдущую команду. history выводит историю Bash, если она включена. Клавиша Tab помогает дополнять имя файла или команды; если вариантов несколько, оболочка может показать список. Ctrl+C передаёт сигнал прерывания выполняемой в терминале программе, Ctrl+D сообщает о конце ввода в соответствующем контексте, а Ctrl+L очищает видимую область терминала без удаления файлов.

При работе по SSH окно терминала может находиться на вашем ноутбуке, а Bash — на удалённом сервере. Перед выполнением меняющей систему команды проверяйте адрес подключения и имя компьютера. Ошибка «не тот сервер» опаснее опечатки в параметре.

Контроль контекста перед работой

```
hostname
whoami
pwd
history | tail -n 5
```

НЕ ПУТАТЬ

Открыть терминал — не значит стать администратором. Команда sudo способна повышать полномочия только согласно настройкам системы, а введённая в терминале строка не становится безопасной из-за того, что она короткая.

2.6. Как искать ранее введённую команду: история Bash

Список history хранит команды текущего сеанса и, при обычной настройке Bash, часть ранее сохранённой истории. История не является аудитом безопасности: некоторые команды не записываются, а несколько одновременно открытых сеансов могут сохранять данные с задержкой. Не полагайтесь на неё как на полный журнал.

Поиск без повторного исполнения

history 20 — последние двадцать строк истории; history | grep -F "apt" — найти строки с буквальным текстом apt; history | grep -i "tree" — поиск без учёта регистра. Если совпадений нет, grep обычно завершится с кодом 1: это не авария системы.

```
history 20
history | grep -F "apt"
history | grep -i "tree"
```

Ctrl+R открывает интерактивный обратный поиск. Начните печатать pwd; Bash предложит совпадение из истории. Повторное Ctrl+R ищет более старое совпадение. Нажмите стрелку вправо, чтобы отредактировать найденную строку, Ctrl+G или Ctrl+C, чтобы отказаться от поиска. Нажатие Enter может сразу выполнить найденную команду, поэтому всегда проверяйте строку.

history | grep -F "sudo" помогает отыскать прежние команды, но поиск не должен становиться оправданием для их слепого повторения. Не помещайте пароли, ключи и токены в параметры команд: они могут попасть в историю и журналы процессов.

2.6.1. Чем отличаются history, поиск и автодополнение

Стрелки вверх/вниз листают введённые команды. Ctrl+R ищет по истории интерактивно. grep фильтрует готовый текст истории. Tab дополняет имя программы, файла или каталога, обычно не обращаясь к истории. history -c очищает историю

текущего сеанса, но это не средство удаления секретов из журналов; в учебных заданиях оно не требуется.

2.7. Типовые ошибки терминала и как их исправлять

2.7.1. bash: команда: command not found

Причины: опечатка, отсутствующий пакет или каталог программы не указан в переменной PATH. Проверка: `type имя_команды` и `printf "%s\n" "$PATH"`. Исправление: перепроверить написание, затем через `apt search` выяснить имя пакета; не копировать случайную команду установки из сети.

2.7.2. Permission denied

Причины: нет права чтения/запуска, скрипт расположен на носителе с ограничениями, запрещает дополнительная политика безопасности. Начните с `ls -l путь` и `id`; не пытайтесь «лечить» проблему массовым `chmod 777` или запуском всего через sudo.

2.7.3. No such file or directory

Причины: неверный путь, регистр букв или отсутствие файла; в некоторых случаях отсутствует интерпретатор из строки shebang. Начните с `pwd` и `ls -la`, проверьте кавычки вокруг пути с пробелами.

ЛАБОРАТОРНАЯ РАБОТА 02 / Один сеанс — три наблюдения

ЦЕЛЬ

Определить в лабораторной системе пользователя, оболочку и расположение команды.

1. Запустите терминал и выполните `whoami`, `pwd`, `hostname`.
2. Сравните `type cd` и `type ls`.
3. Создайте учебный каталог с пробелом в имени и убедитесь, что имя не разделилось.
4. Запишите вывод и объясните, почему `cd` отличается от внешней программы.

ПРОВЕРКА РЕЗУЛЬТАТА

Вы можете назвать пользователя, каталог и компьютер, определить тип команды и привести безопасный пример правильного цитирования пути.

ФИНИШНЫЙ ПРИЁМ / МИКРО-КАТА ИНЖЕНЕРА

Микро-ката: координация без спешки

Короткая разминка без растяжки и резких ударов. Выполняйте движения плавно, рядом с опорой; при боли или головокружении остановитесь. Это перерыв от сидения, а не лечение или обещание улучшения памяти.



1. Плечи

Поднимите и свободно опустите плечи 3 раза.



2. Поворот

Медленно поверните корпус влево; таз остаётся устойчивым.



3. Возврат

Вернитесь к центру и повторите вправо.

Проверьте положение стоп и расслабьте плечи. Сделайте небольшой поворот корпуса влево, вернитесь к центру, затем повторите вправо. Без рывков и резкого скручивания. Выполните два спокойных круга. Задача — осознанно переключать действия, а не тренировать силу удара.

МЫСЛЬ СЕНСЕЯ

«Не путай окно терминала с тем, кто выполняет команду».

МИНИ-СЛОВАРЬ / ГЛАВА 2

Терминал	Программа, в которой вводят и читают текстовые команды.
Shell	Оболочка, разбирающая команды и запускающая программы.
Bash	Распространённая командная оболочка Linux.
PATH	Переменная с каталогами поиска исполняемых файлов.
History	История введённых команд в оболочке.

PEGINELAB // КАРАТЕ LINUX // БЕЛЫЙ ПОЯС

Глава 3

Документация и пакеты



В ЭТОЙ ГЛАВЕ

Учимся самостоятельно искать точное описание команды и устанавливать программы из проверенных источников, не полагаясь на случайные советы.

«Понимание начинает путь. Проверка делает его уверенным».

ГЛАВА 3 / ДОКУМЕНТАЦИЯ И ПАКЕТЫ

3.1. Документация важнее памяти

Linux часто представляют набором загадочных команд. На самом деле операционная система снабжена средствами, которые помогают понять назначение большинства инструментов. Хороший инженер не пытается воспроизвести сотню ключей из памяти: он умеет найти нужный ключ в корректном руководстве, сопоставить его с текущей версией программы и проверить эффект на тестовых данных.

Первый источник — встроенная справка `--help` или `-h`, если утилита поддерживает такой способ. Второй — справочные страницы `man`. Третий — `info` и документация пакета, часто

находящаяся в /usr/share/doc. Наконец, для особенностей дистрибутива пригодятся руководства Ubuntu, Astra Linux и документация разработчика утилиты.

Какие справки можно открыть

```
ls --help
man ls
man 1 ls
man 5 passwd
man 7 hier
```

3.2. Как устроены разделы man

man — это не одна книга, а система справочных страниц с разделами. Например, man 1 passwd описывает пользовательскую команду изменения пароля, а man 5 passwd — формат файла /etc/passwd. Одинаковое имя не означает одинаковую тему. Часто раздел 1 посвящён пользовательским командам, 2 — системным вызовам, 3 — функциям библиотек, 4 — файлам устройств, 5 — форматам файлов, 7 — общим темам и соглашениям, 8 — административным командам.

Страница обычно имеет блоки NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXAMPLES, SEE ALSO. Не пугайтесь английских заголовков: они предсказуемы. SYNOPSIS — форма записи команды, DESCRIPTION — объяснение, OPTIONS — параметры. SEE ALSO сообщает, какие соседние страницы стоит прочитать.

3.3. Как искать внутри man

В типичной системе просмотр man осуществляется через less. Нажмите /, введите слово sort и Enter, чтобы перейти к следующему совпадению. Клавиша n ищет дальше в том же направлении, N — в обратном. q закрывает просмотр. Пробел прокручивает страницу вперёд, b — назад. Если установлены другие просмотрщики, клавиши могут различаться.

Команда man -k keyword ищет по кратким описаниям страниц; apropos обычно делает примерно то же. whatis показывает

короткую сводку для заданного имени. Если поиск пустой, причина может быть в отсутствующей базе описаний или пакете manpages. Перед поиском стоит убедиться, что нужные справочные пакеты установлены.

Поиск по ключевым словам

```
man -k permissions
apropos directory
whatis tree
man tree
```

3.3.1. Русский и английский man

Не все справочные страницы переведены. В Ubuntu и Debian могут существовать отдельные пакеты русских переводов; наличие перевода зависит от версии и состава репозиториев. Откройте `locale`, проверьте наличие русской локали и выполните `apt search manpages-ru` для поиска доступных справочных пакетов. Переменная `LANG=ru_RU.UTF-8` или опция `man -L ru` помогают запросить русскую локаль, но не создают отсутствующий перевод.

Когда русская страница устарела или отсутствует, полезнее открыть актуальную английскую. Термины `owner`, `group`, `permission`, `recursive`, `mount point`, `package`, `repository` постепенно станут привычными. Мы не предлагаем механически переводить технические команды на русский язык: их синтаксис сохраняется.

Проверяем языковые варианты

```
locale
apt search manpages-ru
man -L ru ls
LANG=C man ls
```

ПРОВЕРКА

Если `man -L ru ls` всё равно показывает английский текст, это не поломка терминала. Возможно, переведённая страница отсутствует в установленном комплекте или выбранная локаль недоступна.

3.4. Как искать официальный ответ в интернете

Для системных задач поисковый запрос должен содержать не только проблему, но и семейство ОС, её версию и название команды. Например, вместо «не работает пакет» полезно искать «Ubuntu 24.04 apt repository missing Release file». Для Astra Linux отдельно указывайте редакцию, обновление и официальный раздел документации. Для FreeBSD используйте FreeBSD Handbook и конкретные man-страницы.

Случайный форум может содержать совет для устаревшей версии. Обращайте внимание на дату публикации, историю обновления, поддерживаемый выпуск и объяснение побочных эффектов. Копирование однострочных скриптов из неизвестного сайта в shell особенно рискованно, если они устанавливают пакеты или требуют прав администратора.

3.5. Что такое пакет

Программа редко состоит из одного файла. Ей могут требоваться библиотеки, конфигурация, файлы справки и другие пакеты. Менеджер пакетов ведёт учёт того, что установлено, из какого источника получено и какие зависимости необходимы. В Debian-подобных системах используются пакеты формата DEB и инструменты `dpkg` и `apt`; в Fedora и CentOS Stream — RPM и DNF; в FreeBSD — `pkg`; в Oracle Solaris — система IPS и команда `pkg`, но с иным устройством, чем в FreeBSD.

У слова «пакет» есть два связанных значения: архив с программой и единица управления для пакетного менеджера. Для новичка важнее запомнить различие между файлом, скачанным вручную, и пакетом, зарегистрированным в системе с метаданными и зависимостями.

Действие	Ubuntu / Astra	Fedora / Stream	FreeBSD
Найти	apt search NAME	dnf search NAME	pkg search NAME
Сведения	apt show NAME	dnf info NAME	pkg info NAME

Установить	sudo apt install NAME	sudo dnf install NAME	sudo pkg install NAME
Обновить индекс	sudo apt update	dnf makecache	pkg update

3.6. Что такое репозиторий

Репозиторий — это обслуживаемый источник пакетов и метаданных. Менеджер узнаёт, какие версии доступны, откуда их получать и какие зависимости разрешить. В Ubuntu используются настроенные источники APT; современная система может хранить их в файлах формата `.sources` в `/etc/apt/sources.list.d/` и других предусмотренных конфигурациях. В Fedora обычно используются файлы `.repo`. Нельзя переносить команду добавления репозитория из одной системы в другую без проверки.

Подлинность пакетов обеспечивается механизмами проверки подписей и доверенными ключами конкретного менеджера. Это не делает любой сторонний репозиторий безопасным: доверие к источнику, поддержка обновлений и совместимость остаются ответственностью администратора. Для учебной машины лучше сначала использовать официальные источники, уже настроенные дистрибутивом.

Читаем сведения о пакетах в Ubuntu

```
apt policy
apt list --installed 2>/dev/null | head
apt search tree
apt show tree
```

Команды выше выполняются без изменений системы. `apt policy` помогает узнать об источниках пакетов; `apt search tree` находит описание утилиты; `apt show tree` даёт версию, зависимости и сведения о пакете. Вывод от `apt list` может предупреждать о нестабильности командного интерфейса для скриптов — для интерактивного ознакомления это допустимо.

Перед обновлением нужно различать `apt update` и `apt upgrade`. Первая команда обновляет локальную информацию о доступных версиях, но сама не обновляет установленные

программы. Вторая, с учётом зависимостей и правил менеджера, устанавливает обновления пакетов. Для полного обновления системы существуют дополнительные правила, которые лучше изучать отдельно.

БЕЗОПАСНОСТЬ РЕПОЗИТОРИЕВ

Никогда не добавляйте неизвестный ключ, не отключайте проверку подписей и не выполняйте `curl … | sudo bash` только потому, что кто-то обещает «быстрое исправление». Сначала найдите официальную инструкцию и разберитесь в действиях.

3.7. Версии, поддержка и обновления

Установленная версия пакета — не всегда последняя версия исходного проекта. Дистрибутив может сохранять стабильную ветку и переносить в неё исправления безопасности без изменения основных чисел версии. Поэтому сравнивать безопасность по одному номеру версии неправильно. Важно смотреть список исправлений и политику поддержки выбранного выпуска.

Для учебных примеров в этой книге зафиксирована Ubuntu 24.04 LTS. Это не утверждение, что более новые выпуски отсутствуют. Цель — дать повторяемый маршрут, а перед публикацией окончательной книги сверить названия диалогов установщика и текущие пакеты. В Astra Linux дополнительно важно учитывать конкретную редакцию и установленное обновление.

3.8. Как читать техническую литературу

Технические книги полезно читать избирательно. В «Библии Linux» Кристофера Негуса (10-е издание, русский перевод 2022) подробно обсуждаются оболочка, файловая система, установка и пакеты. В нашей книге достаточно использовать её как дополнительный маршрут чтения: не нужно повторять весь материал и не следует воспринимать описания старых

выпусков как инструкции для современной Ubuntu или Astra Linux.

Для командной строки ориентируйтесь на главу 3 (с. 96–128), для файловой системы — главу 4 (с. 130–150), для установки — главу 9 (с. 238–264), для управления программами — главу 10 (с. 265–291), для учётных записей — главу 11 (с. 293–316). Перед самостоятельной практикой проверяйте справочную страницу установленной у вас версии утилиты.

3.9. Обязательная подготовка стенда «Белый пояс»

Для воспроизводимых примеров этой книги выбрана Ubuntu 24.04 LTS с Bash. Уроки можно выполнять на отдельном компьютере или в виртуальной машине. Сведения об Astra Linux и других системах идут как сравнение: имена пакетов и доступность программ проверяются по документации конкретного выпуска. Не экспериментируйте с разметкой единственного рабочего диска.

3.9.1. Что нужно до первого занятия

1. Компьютер или виртуальная машина с рабочей Ubuntu 24.04 LTS; для нового стенда ориентир — от 4 ГБ ОЗУ, 2 ядра CPU и не менее 30 ГБ свободного пространства на учебном виртуальном диске.

2. Учётная запись обычного пользователя с правом `sudo` для установки учебных пакетов; доступ в интернет для загрузки пакетов и обновлений.

3. Приложение Terminal, оболочка Bash и пакетный менеджер APT. В типичной Ubuntu они уже присутствуют, но перед началом мы проверяем это командами.

4. Средства справки и упражнений: `man-db`, `manpages`, `less`, `tree`, `nano`, `grep`, `findutils`, `coreutils`, `bash-completion`, `file`, `curl`, `ca-certificates`; переводы man-страниц на русский — по желанию и при наличии в репозиториях.

Диагностика перед установкой

cat /etc/os-release — узнать систему; bash --version — версию Bash; command -v apt — доступность APT; command -v man и command -v tree — проверка справки и дерева каталогов; df -h / — свободное место на системном разделе.

Установка учебных инструментов в Ubuntu

```
sudo apt update
sudo apt install \
    man-db manpages less tree nano grep findutils \
    coreutils bash-completion file curl ca-certificates
# Необязательно: при наличии пакета в репозиториях
sudo apt install manpages-ru
```

В Ubuntu часть перечисленных пакетов обычно уже установлена — повторная команда APT не создаёт их дубликаты. Если пакет manpages-ru недоступен, это не мешает заниматься по английским руководствам. Для поддержки man -k может понадобиться sudo mandb после установки новых справочных страниц.

3.9.2. Фиксируем минимальный чек-лист

man ls должен открыть справку, tree --version — вывести версию, nano --version — сведения о редакторе, locale — текущие языковые настройки. Если команда не найдена, возвращайтесь к диагностике ниже. Проверьте command -v git: Git полезен для следующих поясов, но не обязателен для глав 1–3.

3.10. Типовые проблемы установки пакетов

3.10.1. E: Unable to locate package

APT не нашёл имя пакета в текущем индексе. Проверьте apt search ИМЯ, затем sudo apt update. Некоторые пакеты отсутствуют в конкретном выпуске или репозитории. Не добавляйте неподписанные источники ради одного урока.

3.10.2. Temporary failure resolving ...

Это обычно проблема DNS или сетевого соединения, а не ошибка имени пакета. Проверьте соединение и адреса DNS,

например getent hosts archive.ubuntu.com; в виртуальной машине дополнительно проверьте настройки сети гипервизора.

3.10.3. Could not get lock /var/lib/dpkg/lock-frontend

Другой процесс APT или обновлений может выполнять операцию. Посмотрите процессы ps -ef | grep -E "apt|dpkg"; дождитесь окончания штатного обновления. Не удаляйте файлы блокировок вручную и не прерывайте неизвестный процесс только ради продолжения учебной работы.

3.10.4. man: No manual entry for ...

Либо такой страницы нет, либо раздел выбран неверно. Проверьте имя через command -v, список через man -k, доступность страниц пакета через dpkg -L ИМЯ_ПАКЕТА | grep /man/. Для встроенных команд Bash используйте help cd или help history.

3.10.5. Русская справка открывается по-английски

Проверьте locale -a и наличие перевода именно для команды и раздела. Пробуйте man -L ru ls, но если нужной русской страницы нет, используйте английскую: LANG=C man ls. Отсутствие перевода не означает поломку программы.

3.11. Электронные задания и контроль самостоятельности

К каждой лабораторной работе прилагается простой текстовый файл в репозитории PEGINELAB: github.com/musikfed/karate_LINUX. Для первых трёх глав это exercises/ch01.txt, exercises/ch02.txt и exercises/ch03.txt. Файлы предназначены для чтения и ручного выполнения, а не для автоматического запуска. В печатном PDF адрес указан как обычный текст, без активного веб-перехода.

Открытых готовых ответов в этих файлах нет: цель первого пояса — научиться проверять результат самостоятельно. Стандартное шифрование ответов и процедура их проверки будут разобраны в книге 3 «Зелёный пояс — Практика». Алгоритм, формат хранения и способ безопасного

предоставления ключа будут описаны и испытаны там отдельно.

ЛАБОРАТОРНАЯ РАБОТА 03 / Найди ответ без подсказки

ЦЕЛЬ

Узнать, что означает параметр `-D` у утилиты tree, не меняя настройки системы.

1. Проверьте, установлена ли tree: `command -v tree`.
2. Если утилита отсутствует, найдите пакет: `apt search tree`.
3. В официальной документации или man-странице найдите описание параметра `-D`.
4. Напишите ответ своими словами и укажите версию утилиты.

ПРОВЕРКА РЕЗУЛЬТАТА

Вы нашли описание именно параметра с прописной D и не перепутали его с `-d`.

ФИНИШНЫЙ ПРИЁМ / МИКРО-КАТА ИНЖЕНЕРА

Микро-ката: дыхание, кисти и шаг

Короткая разминка без растяжки и резких ударов. Выполняйте движения плавно, рядом с опорой; при боли или головокружении остановитесь. Это перерыв от сидения, а не лечение или обещание улучшения памяти.



1. Вдох

Станьте ровно, расслабьте плечи, выдохните.



2. Ладони

Раскройте и мягко сожмите кисти 5 раз.



3. Шаг

Переносите вес с ноги на ногу у опоры 20 секунд.

Выдохните спокойно, опустите плечи. Пять раз разожмите и снова соберите пальцы без усилия. Затем двадцать секунд медленно шагайте на месте у опоры. При неприятных ощущениях прекратите движение. Это короткий перерыв от сидячей работы, а не медицинская процедура.

МЫСЛЬ СЕНСЕЯ

«Хороший инженер не знает всё. Он умеет находить подтверждение».

МИНИ-СЛОВАРЬ / ГЛАВА 3

Пакет	Набор файлов и метаданных для установки ПО.
Репозиторий	Подписанный источник пакетов и сведений о них.
APT	Инструменты управления пакетами в Debian/Ubuntu.
MAN-страница	Справка по программе или системному интерфейсу.
Locale	Набор языковых и региональных настроек.

Глава 4

Установка Ubuntu



В ЭТОЙ ГЛАВЕ

Создаём самостоятельную учебную машину: от образа ISO и виртуального диска до первого входа и проверяемого результата.

«Не начинай с кнопки «Установить». Сначала узнай, на какой диск она указывает.»

ГЛАВА 4 / УСТАНОВКА UBUNTU

4.1. Что именно мы устанавливаем

Установка операционной системы — не мистический обряд с флешкой. Это последовательность проверяемых действий. Сначала выбирают подходящий образ, затем готовят среду загрузки, определяют диск назначения, создают файловую систему, записывают ОС и только после перезагрузки убеждаются, что машина стала работать самостоятельно.

Для нашего первого курса выбран Ubuntu Desktop 24.04 LTS. К моменту чтения могут существовать более новые выпуски.

Здесь важна не погоня за последним номером, а единый учебный стенд: читатель и автор должны разговаривать об одинаковой системе, когда ищут ошибку в командах. Если вы установили более новую Ubuntu, проверьте отличия установщика и имён пакетов по документации своего выпуска.

Desktop — вариант с графической средой, удобный для начинающего. Server обычно ставят без полноценного графического рабочего стола и администрируют через консоль либо сеть. Ядро, базовая файловая и пакетная логика у этих вариантов родственны, но выбор компонентов и первоначальная настройка различаются. В «Белом поясе» сначала осваиваем Desktop; к серверному варианту вернёмся после изучения каталога, пользователей и прав.

ГЛАВНАЯ ГРАНИЦА БЕЗОПАСНОСТИ

Практика выполняется на виртуальном диске или на отдельном пустом накопителе. Не выбирайте для стирания единственный рабочий SSD с Windows, личными документами и ключами шифрования. Название диска в установщике следует проверять по размеру и модели, а не по его месту в списке.

4.2. Три решения до начала установки

4.2.1. Отдельный компьютер, виртуальная машина или соседство с Windows

Самый спокойный способ знакомства — виртуальная машина (ВМ). Основная ОС продолжает работать, а Ubuntu получает условный компьютер: виртуальные процессор, память, сеть и накопитель. Поэтому шаг «стереть диск» внутри правильно созданной ВМ означает стереть её отдельный виртуальный диск, а не физический накопитель хозяина. Однако ошибка выбора загрузочного образа или подключение реального диска напрямую способна разрушить эту изоляцию, поэтому настройки проверяем заранее.



Рис. 4.1. Где заканчивается компьютер хозяина и начинается учебный стенд. Схема условная.

Вариант	Когда выбирать	Главный риск
Виртуальная машина	Первая практика, повторение, снимки состояния	Не хватает ОЗУ или дискового пространства у хоста
Отдельный SSD / ПК	Нужна реальная проверка оборудования	Случайное стирание не того накопителя
Две ОС на одном диске	Нужно постоянное соседство Windows и Ubuntu	Разделы, загрузчик и шифрование требуют опыта

Двойная загрузка технически возможна, но не нужна для освоения белого пояса. Мы сознательно откладываем её: там придётся учитывать загрузчик Windows, BitLocker, восстановление EFI-записей и существующие разделы. Пара десятков команд пока не стоит такого риска.

4.2.2. Ресурсы учебной машины

Для комфортного опыта с графическим рабочим столом полезно выделить ВМ две–четыре виртуальных процессорных нити, от 4 до 6 ГиБ ОЗУ при достаточном запасе у хоста и диск 60 ГиБ с динамическим выделением места. Это редакционный ориентир для курса, не официальный минимальный паспорт Ubuntu. Canonical указывает для установки Desktop не менее 25

ГБ пространства; производительность зависит от оборудования и версии.

Под динамическим диском понимается файл, который растёт по мере записи данных в виртуальную систему до заданного верхнего размера. ВМ может показывать гостевой системе «60 ГиБ», хотя на диске хозяина файл пока занимает меньше. Резерв свободного места у хоста всё равно необходим: заполненный до отказа накопитель быстро превратит лабораторную работу в поиски потерянной файловой системы.

4.3. Откуда брать образ и как проверить его подлинность

Образ ISO — файл, содержащий установочную среду. Для Ubuntu 24.04 LTS используйте официальный архив выпусков. Адрес для самостоятельного ввода в браузере: releases.ubuntu.com/24.04/. Не скачивайте «подправленные сборки», где неизвестный автор уже добавил драйверы, ускорители и якобы готовую защиту.

В имени образа встречаются номер выпуска, вариант среды и архитектура. Для большинства обычных ПК на Intel/AMD подходит архитектура amd64; это название совместимой 64-разрядной платформы, а не указание, что у вас должен быть процессор AMD. Для устройств ARM нужна другая сборка: чужой образ не запустится просто потому, что красивее называется.

4.3.1. Контрольная сумма — первая проверка

Контрольная сумма SHA-256 позволяет обнаружить отличия скачанного файла от файла, чей эталонный хеш опубликован поставщиком. Она не заменяет проверку достоверности самого источника. Не копируйте хеш из случайного форума: тогда сравнение двух совпадающих чисел ничего не докажет.

Windows PowerShell: имя ISO замените на фактически скачанное

```
Get-FileHash "C:\ISO\ubuntu-24.04.5-desktop-amd64.iso" -Algorithm SHA256
```

Linux: выполняйте в каталоге, где лежит ISO

```
sha256sum ubuntu-24.04.5-desktop-amd64.iso
```

На странице выпуска найдите файл официальных сумм SHA256SUMS и сверяйте значение для своего ISO. Если контрольная сумма не совпадает, не продолжайте установку. Перезагрузите файл из проверенного источника. В данном примере имя ISO иллюстративно: на странице выпуска число обновления и состав файлов могут меняться.

4.3.2. Если устанавливаем на физический компьютер

Для записи на флешку используйте проверенную программу, поддерживающую запись загрузочного ISO, и официальную инструкцию Ubuntu. Нужна флешка подходящего объёма (в руководстве Canonical рекомендуется 8 ГБ и больше). Перед запуском записи проверьте выбранное USB-устройство. Всё его содержимое будет уничтожено. На виртуальной машине флешка не нужна: ISO можно подключить прямо к виртуальному оптическому приводу.

КАК НЕ ПЕРЕПУТАТЬ НОСИТЕЛИ

Два устройства на 64 ГБ могут выглядеть одинаково. При записи флешки проверяют её модель и съёмный тип; при установке ОС — модель и ёмкость целевого диска. Если сомневаетесь, остановитесь и отключите посторонние накопители.

4.4. Настройка виртуальной машины

Принцип одинаков для VirtualBox, VMware Workstation и других гипервизоров, хотя имена кнопок отличаются. Создайте новую машину Linux / Ubuntu (64-bit), подключите ISO и добавьте отдельный виртуальный диск. Выберите вариант прошивки UEFI, если он доступен и совместим с выбранным установщиком. Следите, чтобы динамически выделенные 60 ГиБ не превышали доступные ресурсы компьютера хозяина.

1. Создайте ВМ с понятным именем, например KL-White-Ubuntu. Не используйте имя рабочей машины или личный пароль в названии.

2. Выделите 2–4 виртуальных потока и 4–6 ГиБ ОЗУ, оставляя достаточно ресурсов Windows или другой хост-системе.
3. Создайте новый виртуальный диск на 60 ГиБ. Не выбирайте режим подключения существующего физического диска.
4. Подключите скачанный ISO к виртуальному приводу и проверьте порядок загрузки.
5. Сохраните параметры ВМ и сделайте снимок конфигурации либо копию её описания до эксперимента.

ЕСЛИ В МЕНЮ ВМ НЕТ 64-РАЗРЯДНОЙ ГОСТЕВОЙ СИСТЕМЫ

Проверьте, поддерживает ли процессор аппаратную виртуализацию и разрешена ли она в прошивке. В Windows одновременно используемые механизмы Hyper-V/VBS могут влиять на работу конкретного гипервизора. Не отключайте системную защиту вслепую: сначала прочитайте документацию выбранной платформы.

4.5. От запуска ISO до первого входа

4.5.1. Пробный запуск и клавиатура

В окне установщика Ubuntu 24.04 возможны этапы выбора языка, доступности, раскладки клавиатуры, сети, типа установки и учётной записи. Формулировки и порядок меняются при обновлениях установщика. Поэтому следим за смыслом действия, а не ищем на экране кнопку, которая обязана называться точь-в-точь как на чужом скриншоте.

Если есть возможность сначала загрузиться в пробный сеанс, используйте её: убедитесь, что мышь и клавиатура отвечают, изображение не исчезает, сеть доступна. ВМ покажет виртуальное оборудование, поэтому успешный пробный запуск здесь ещё не доказывает совместимость реального компьютера — это разные устройства.