



Конспект: Основы Shell (Woche 3)

1. "Большая Картина": Как всё связано (Архитектура)

Чтобы понять, *что* такое Shell, нужно видеть всю "цепочку" от вас до "железа":

1. **Вы (Пользователь)**
 - Работаете в **Приложениях (Anwendungen)** (напр., Chrome, VS Code).
2. **Shell (Оболочка)**
 - Ваш "**переводчик**" (интерпретатор) на язык Ядра.
 - *Примеры:* **bash** (стандарт Linux), **zsh** (стандарт macOS), **PowerShell** (Windows).
3. **Ядро (Kernel)**
 - "**Мозг**" операционной системы (OS). Главный менеджер.
 - *Задачи:* Управляет памятью, файлами, безопасностью и оборудованием.
4. **Драйверы (Treiber) / HAL**
 - "**Посредники**", которые переводят команды Ядра для *конкретного* "железа".
5. **Оборудование (Hardware)**
 - Физические компоненты: **CPU** (процессор), **RAM** (память), **SSD/HDD** (диски).

Схема работы:

Вы → пишете команду в [Shell] → Shell переводит ее для [Ядра] → Ядро выполняет задачу → Shell показывает вам результат.

Интерпретатор (Bash) vs. Компилятор:

- **Shell (Интерпретатор):** Читает и выполняет команды **строка за строкой**.
- **Компилятор (C++, Java):** *Сначала* переводит всю программу в 1 файл, *потом* запускает.

2. Навигация: Как передвигаться в системе

Команда	Вопрос / Действие	Пример
<code>pwd</code>	"Где я?" (Print Working Directory)	<code>pwd</code> (покажет <code>/home/user</code>)
<code>cd</code>	"Идти в..." (Change Directory)	<code>cd /var/log</code> (идти в <code>log</code>)

<code>ls</code>	"Что вокруг?" (List)	<code>ls</code> (покажет файлы в папке)
<code>ls -l</code>	"Покажи подробно" (Long format)	<code>ls -l</code> (покажет права, владельца)

Ключевые символы пути:

- `/` (Корень): Самое начало всей файловой системы.
- `.` (Точка): **Текущий** каталог (где вы сейчас).
- `..` (Две точки): **Родительский** каталог (на один уровень вверх).

3. Схема: Как читать права доступа

Команда `ls -l` показывает права в виде `d rwx r-x r-x`:

d	rwX	r-X	r-X
Тип	Владелец (User)	Группа (Group)	Остальные (Other)
<code>d</code> = Директория	<code>r</code> = Чтение (Read)	<code>r</code> = Чтение	<code>r</code> = Чтение
<code>-</code> = Файл	<code>w</code> = Запись (Write)	<code>-</code> = (Нет права)	<code>-</code> = (Нет права)
	<code>x</code> = Выполнение (Execute)	<code>x</code> = Выполнение	<code>x</code> = Выполнение

4. Управление (CRUD-операции)

CRUD (Create, Read, Update, Delete) — это 4 основных действия с файлами.

C: Create (Создать)

- `mkdir <имя_папки>`: Создать **папку** (Make Directory).
- `touch <имя_файла>`: Создать пустой **файл**.

R: Read (Прочитать)

- `cat <файл>`: Показать *всё* содержимое (для маленьких файлов).
- `less <файл>`: Интерактивный просмотр (для больших файлов, выход - q).
- `head / tail <файл>`: Показать *начало* / *конец* файла.

U: Update (Обновить/Изменить)

- `cp <откуда> <куда>`: **Копировать** (Copy).
- `mv <старое> <новое>`: **Переместить** или **Переименовать** (Move).

D: Delete (Удалить)

- `rm <файл>`: Удалить **файл** (Remove).
- `rm -r <папка>`: Удалить **папку** и всё внутри (Recursive).

Нужна помощь?

- `man <команда>`: Показать **инструкцию** (Manual) для любой команды (напр., `man ls`).

5. Режим "Суперпользователя" (Sudo)

- **root**: Это "суперпользователь" (администратор). Он может делать *всё*, включая удаление системы. Работать под ним постоянно **опасно**.
- **sudo**: Команда "**Выполнить как root**" (Super User Do). Позволяет *временно* получить права **root** для *одной* конкретной команды.
 - *Пример*: `sudo rm /защищенный/файл` (Запросит ваш пароль).