

# Первые команды



# Введение

- Команды линукса
- Терминалы
- Вход на сервер
- Новые команды
- Как пользоваться командами
- Корневой раздел
- Работа с терминалом
- Флаги и аргументы
- Полный и относительные пути
- Перемещение и переименование
- История



# Терминал: песочница

- Терминал = песочница
- Терминал для ввода команд открывается для каждой браузерной сессии.
- При работе в этих терминалах нельзя обновлять страницу браузера или двигаться вперед и назад стрелками, чтобы не потерять историю введенных команд.
- Нет необходимости знать все команды наизусть.
- Можно воспользоваться подсказками или ввести в терминале:
  - КОМАНДА `-help` - `-help`(иногда `-h`)



# Терминал: консоль

- Терминал, оболочка для ввода команд есть в любой ОС под управлением ядра Linux.
- Оболочка для ввода команд присутствуют в MacOS - базовые команды идентичны и выполняют такие же функции.
- Терминалы служат для общения пользователя с системой.
- Вводим команды в терминале в виде текста и получаем результат выполнения в виде текста как результат работы программ.



# Начало работы с терминалом

Что мы видим, когда входим на сервер или начинаем работу в терминале:

```
localhost:~#
```

- **localhost** - это название сервера, его внутреннее имя. Вместо него может быть IP адрес.
- **~** говорит нам о том, что мы работаем из домашней директории.
  - По умолчанию все пользователи начинают работу из домашней директории
- **#** - символ суперадминистратора или суперпользователя.
  - Это тот, кто является самым главным на сервере и может создавать и удалять все файлы и папки, запускать процессы и администрировать все ресурсы и модули, подключенные на удаленном устройстве.
  - Для привилегированного пользователя вместо **#** будет **%**. Этот пользователь может поднять уровень привилегий до суперпользователя, но по умолчанию им не является.

# Вход на сервер

Давайте перейдем по ссылке в песочницу и начнем занятие:

[JSLinux](#)



# Команды Linux

Команда	Пояснение
<b>mkdir</b> (make directory)	создание папки
<b>touch</b>	создание файла
<b>cp</b> (copy)	копирование
<b>mv</b> (move)	перемещение
<b>rm</b> (remove)	удаление файла/папки  *Иногда может помогать нам удалять целые каталоги, но для этого нужен флаг <b>-rf</b>
Полный и относительный путь	

Работа в терминале не имеет функции отмены.  
"Откатить" удаленный файл или папку невозможно!

# Команды Linux

Команда	Пояснение
<b>whoami</b>	узнать, под каким именем мы работаем в системе
<b>mv</b>	копирование файлов и папок
<b>rm</b>	команда удаления
<b>tree</b>	просмотр дерева каталогов
<b>history</b>	вывод истории
<b>pwd</b>	просмотр рабочей директории
<b>export_file</b>	скачать файл из песочницы
<b>cd</b>	команда навигации по каталогам
<b>mkdir</b>	создание папок
<b>cp</b>	команда копирования
<b>touch</b>	создание пустых файлов
<b>ls</b>	просмотр содержимого каталога



# Первые команды

- Терминал (командная оболочка) служит для обработки команд и общения пользователя с операционной системой.
- Все вводимые команды будут обрабатываться и результат будет виден в терминале.
- Отсутствие ответа от ОС после ввода команды - стандартное поведение. Например, при создании каталога система ничего не возвращает, а просто создает нужную директорию.
- Все команды отделяются друг от друга. Несколько команд можно ввести в одну строку, но это не популярная мера.
- Команды необходимо отделять от аргументов, ключей и прочей вводимой информации пробелами.



# Корневой раздел файловой системы

Корневой раздел - начало файловой системы.

- Он обозначается символом / и в нем находятся все папки, с которыми происходит взаимодействие в процессе работы.
- Перейти в него можно при помощи команды `cd /` или набрав `cd ..` (если курсор находится на один уровень ниже корневого раздела).

Все файлы, папки, картинки, музыка и даже устройства, система видит как обычные текстовые файлы.

Основным рабочим инструментом в системе будет текстовый редактор.

# Экспресс-опрос

- **Вопрос 1.**

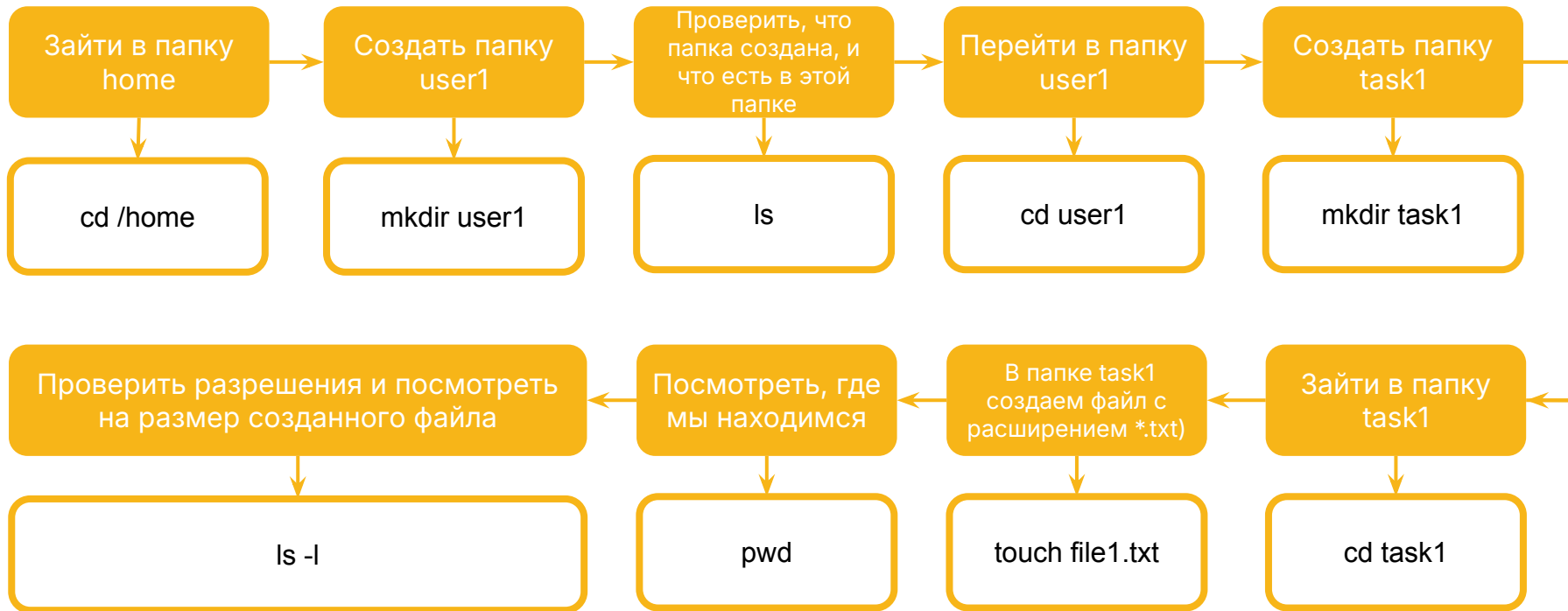
Что такое терминал и командная оболочка?

- **Вопрос 2.**

Какая логика построения команд в Linux?



# Работа с терминалом



# Флаги и аргументы

- Флаги или ключи идут с минусом -ключ меняет визуальное отображение.  
Например, команда **ls -l (l - long)** - подробная информация.
- Есть **-a** где показываются все файлы, в том числе скрытые.
- Есть псевдоним **ls -la**  
ls -a  
ls -l



# Работа с терминалом

- Если экран забит лишней информацией, то можно сделать команду **clear** и наш терминал станет чистым.

**Ctrl + L** - очистить экран и **clear**.

- Можно показать содержимое директории и субдиректорий:

Реализуются с помощью команды **tree**.

Не стоит выполнять команду **tree** в корневом каталоге, так как терминал начнет показывать абсолютно все файлы!



# Команда tree

Если использовать команду **tree** со слешем (/ИМЯ\_ПАПКИ), то она покажет ветвления от папки, которую ты запросишь в терминале.

Пример: `tree /home/`

**tree** есть не во всех дистрибутивах.

Пробовать можно во всех, но иногда результата мы не увидим.



# Работа с терминалом

- Создадим несколько папок одна в одной по относительному пути.
- В папке home должно получиться несколько папок (одна в одной).
- В терминале можно посмотреть дату и время.

Делается это командой **date**.

- Можно посмотреть, что мы вводили и в каком порядке.

Для этого нам поможет команда **history**.





# Работа с терминалом

Можно создавать файлы и папки одной строкой, не заходя каждый раз в новосозданную папку и подпапки.

- `mkdir -p /home/user2/task2/`
- `touch /home/user2/task2/file2.txt`
- `mkdir -p /home/user3/task3/`

Флаг **-p** позволяет создать путь сокращение от path, если нужно создать папку и путь.

Если нужно создать одну папку то `mkdir`, при необходимости - можно создать весь путь с субдиректориями, просто добавив флаг **-p**

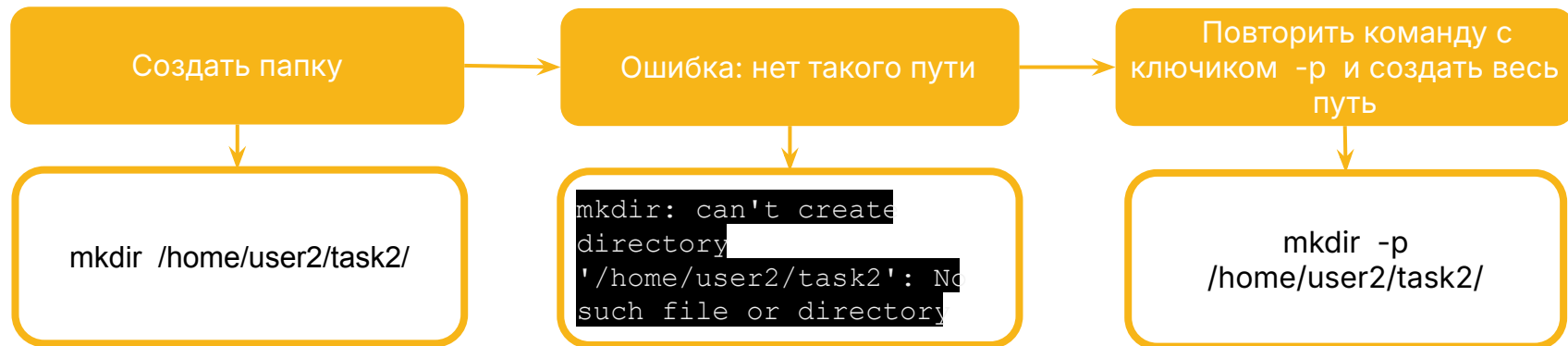
# mkdir -p

1. **mkdir -p** используется для избежания ошибок при повторных запусках команда создания папок
2. **mkdir -p** используют для скриптов, чтобы не возвращать ошибку, а убедиться в наличии путей.

Скрипт - это небольшая программа, которая содержит последовательность действий, созданных для автоматического выполнения задачи.



# mkdir -p



# Полный и относительные пути

Полный/абсолютный путь

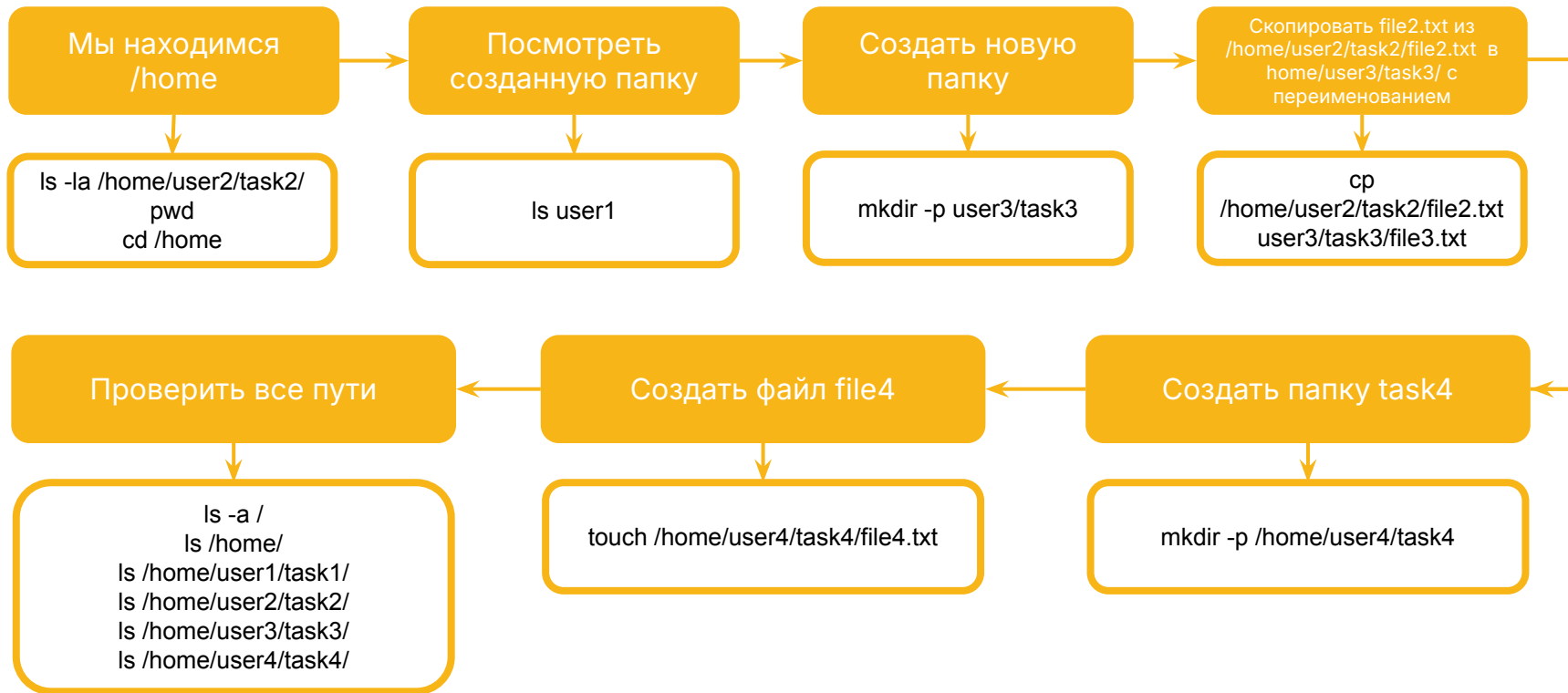
Относительный путь

Путь начинается от корня "/" и описывает весь путь к файлу.

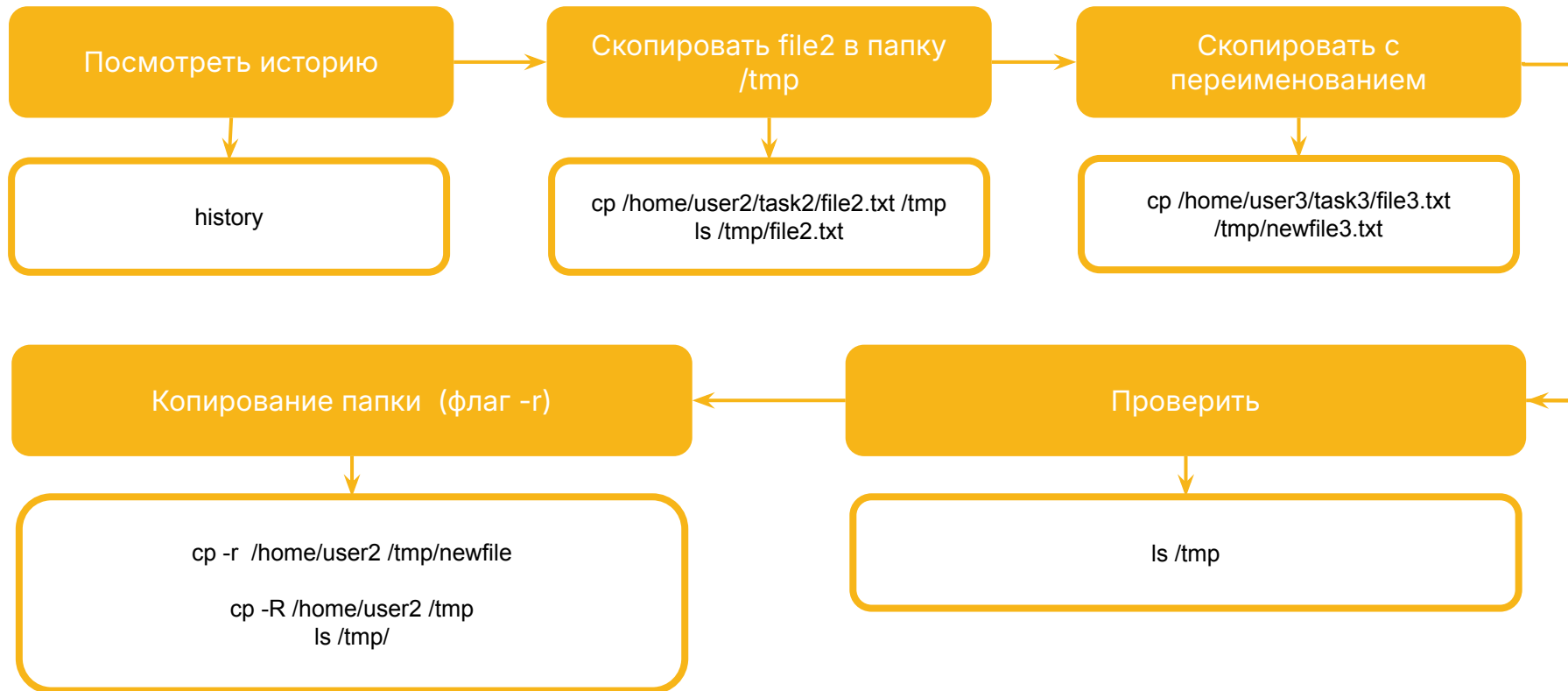
Путь к файлу относительно текущей папки, такие пути часто вызывают путаницу

Если не уверен в правильности пути - используй полный, проверяя себя кнопкой TAB каждый шаг!

# Навигация, создание файлов и папок



# История и копирование



# Перемещение и переименование

- Перемещение и переименование происходит при помощи команды mv, используя 2 аргумента: что переносим и куда переносим.

Переносить и копировать можно с переименованием:

старое название → новое название

- При переименовании файл переносится в ту же локацию, только с другим именем.

```
mv /opt/newfile /opt/newfile.txt  
mv /opt/newfile.txt /tmp/new_text  
rm /tmp/file2.txt  
ls /tmp
```

# Удаление

- Удаление пустых папок командой **rmdir**
- Удаление файлов и папок командой **rm**
- **Удаление иерархии с вложениями `rm -r`**





# История

- Посмотреть историю:

**history**

- Вывод истории и запись ее на свой компьютер:

```
history > /tmp/history.txt  
export_file /tmp/history.txt
```



# Экспресс-опрос

- **Вопрос 1.**

Как вы поняли разницу между полным и относительным путем?

- **Вопрос 2.**

Для чего нужен ключ -r?

- **Вопрос 3.**

Какой ключ мы используем для копирования папки целиком?



## ЗАДАНИЕ

Соотнесите название команды и то, что она делает:  
Ответ напишите в чат в виде: 1А, 2В...

Команда	Пояснение
1. <b>mkdir</b>	A. создание файла
2. <b>touch</b>	B. удаление файла/папки
3. <b>cp</b>	C. перемещение
4. <b>mv</b>	D. копирование
5. <b>rm</b>	F. создание папки

# Домашнее задание

1. Создать папку `animals` в `/home`
2. Создать подпапки `dogs`, `cats`, `fish` в `animals`, созданной ранее папке
3. Добавить файлы `pluto`, `nemo` и `garfield` в папках, созданных ранее (`pluto goes to dogs`, `garfield goes to cats`)
4. В `/tmp` создать файл `adam`
5. Копировать `adam` в `/home/humans` (создать директорию)
6. Создать файл `evA` в `/tmp`
7. Переместить с переименованием `evA` в `/opt/eve`
8. Переместить `eve` в `humans` ( `/home/humans`)
9. Проверить, что `adam` и `eve` находятся в папке `humans`



# Полезные ссылки

- [Основные linux-команды для новичка](#)