

# Лебедев Антон ИКБО-11-24

Проект представляет собой пошаговую разработку эмулятора командной строки, имитирующего работу в UNIX-подобной операционной системе. Эмулятор работает с виртуальной файловой системой (VFS), загружаемой из XML-файла, и поддерживает выполнение команд как в интерактивном режиме, так и через стартовые скрипты.

## Реализованные этапы

### Этап 1: REPL (Read-Eval-Print Loop)

Файл: `1_stage.py`

Минимальный прототип

Возможности:

- Графический интерфейс с текстовым полем для вывода и строкой ввода команд
- Приглашение к вводу в формате `username@hostname:~$` (на основе реальных данных ОС)
- Парсер команд с поддержкой аргументов в кавычках (через `shlex`)
- Команды-заглушки: `ls`, `cd`
- Команда выхода: `exit`
- Обработка ошибок парсинга

### Этап 2: Конфигурация

Файл: `2_stage.py`

Добавлена поддержка параметров командной строки и стартовых скриптов.

Возможности:

- Параметры командной строки:
  - `--vfs <путь>` - путь к VFS
  - `--startup <путь>` - путь к стартовому скрипту
- Выполнение стартового скрипта при запуске
- Обработка ошибок в скриптах (продолжение выполнения при ошибках)
- Служебная команда `conf-dump` для вывода параметров конфигурации

### Этап 3: VFS (Virtual File System)

Файл: `3_stage.py`

Интеграция виртуальной файловой системы из XML.

Возможности:

- Загрузка VFS из XML-файла
- Поддержка base64-кодирования для содержимого файлов
- Навигация по директориям (команда `cd`)
- Просмотр содержимого директорий (команда `ls`)
- Все операции выполняются в памяти
- Обработка ошибок загрузки VFS

### Этап 4: Основные команды

Файл: `4_stage.py`

Добавлены функциональные команды для работы с VFS.

Реализованные команды:

- `ls [путь]` - была реализована в 3 этапе
- `cd [путь]` - была реализована в 3 этапе
- `wc <файл>` - подсчёт строк, слов и символов в файле
- `history` - история введённых команд
- `conf-dump` - вывод параметров конфигурации
- `exit` - выход из эмулятора

История команд: Все выполненные команды сохраняются в памяти и доступны через `history`.

### Этап 5: Дополнительные команды

Файл: `5_stage.py` ¶ **Финальная версия**

Финальная версия с консольным интерфейсом (CLI) и расширенным функционалом.

Новые возможности:

- `chown <владелец>[:группа] <файл>` - смена владельца файла/директории

- `vfs-load <путь>` - загрузка новой VFS без перезапуска эмулятора
- Поддержка комментариев в стартовых скриптах (строки начинающиеся с `#` )
- Атрибуты `owner/group` для файлов и директорий в VFS

## Тестирование

---

Для каждого этапа созданы тестовые скрипты и примеры VFS, демонстрирующие:

- Корректную обработку команд
- Работу с различными структурами VFS
- Обработку ошибок
- Граничные случаи