

Лабораторная работа №2

Тема: Мини-оболочка с файловыми командами

Цель и результаты

- Освоить работу с файловой системой средствами Python.
- Реализовать основные команды управления файлами и каталогами.
- Научиться вести журнал действий пользователя.
- (Medium) Освоить расширение функциональности за счёт дополнительных модулей-плагинов.

Предварительные знания

- Работа с файлами и каталогами в Python (`os`, `os.path`, `pathlib`, `shutil`).
- Работа с вводом/выводом (`input`, `print`, чтение и запись файлов).
- Исключения (`try/except`).
- Основы логирования (`logging`).
- (Medium) Работа с архивами (`zipfile`, `tarfile`), регулярными выражениями (`re`).

Ограничения

- Запрещено использовать внешние оболочки или запуск системных команд (`subprocess`).
- Все операции должны выполняться средствами Python API.
- Поддержка **относительных и абсолютных путей**.
- Логирование обязательно: все команды и ошибки должны фиксироваться в файле `shell.log`.

Варианты заданий

Easy (обязательная часть)

1. Команда `ls`

- Отображение списка файлов и каталогов в текущем рабочем каталоге.
- Поддержка аргумента `path` (относительный или абсолютный путь).
- Поддержка опции `-l` для подробного отображения (имя, размер, дата изменения, права доступа).

2. Команда `cd`

- Переход в указанный каталог.
- Поддержка `..` для перехода на уровень выше.
- Поддержка `~` для перехода в домашний каталог.
- В случае ошибки (каталог не существует) — вывод сообщения и запись в журнал.

3. Команда `cat`

- Вывод содержимого указанного файла в консоль.
- В случае передачи каталога вместо файла или отсутствия файла — ошибка и запись в журнал.

4. Команда `cp`

- Копирование файла из источника в назначение.
- Поддержка опции `-r` для рекурсивного копирования каталогов вместе с содержимым.
- В случае невозможности копирования (например, недостаточно прав) — сообщение об ошибке.

5. Команда `mv`

- Перемещение или переименование файла/каталога.
- Поддержка перемещения в существующий каталог.
- Проверка на существование источника и прав доступа.

6. Команда `rm`

- Удаление указанного файла.
- Поддержка опции `-r` для удаления каталога рекурсивно со всем содержимым.
- При удалении каталога требуется подтверждение (`y/n`).
- Ограничение: запрещено удалять корневой каталог (`/`) или родительский (`..`).

7. Логирование

- Все команды, выполненные пользователем, должны записываться в файл `shell.log`.
- В лог пишется: дата/время, введенная команда, результат (успех/ошибка).
- Формат строки лога:

```
[2025-10-01 13:45:22] ls -l /home/user  
[2025-10-01 13:45:23] ERROR: No such file or directory
```

Medium (плагины, дополнительные баллы)

1. Поддержка архивов (zip, tar)

- Команда `zip <folder> <archive.zip>` — создание архива формата ZIP из каталога.
- Команда `unzip <archive.zip>` — распаковка архива ZIP в текущий каталог.
- Команда `tar <folder> <archive.tar.gz>` — создание архива формата TAR.GZ.
- Команда `untar <archive.tar.gz>` — распаковка архива TAR.GZ.
- Все операции должны логироваться.

2. Поиск файлов по содержимому

- Команда `grep <pattern> <path>` — поиск строк, соответствующих шаблону `pattern` в файлах.
- Поддержка опции `-r` — рекурсивный поиск в подкаталогах.
- Поддержка опции `-i` — поиск без учёта регистра.
- Вывод: имя файла, номер строки и найденный фрагмент.

3. История команд и undo

- Команда `history` — вывод последних N введенных команд с их номерами.
- Команда `undo` — отмена последней команды из списка `cp`, `mv`, `rm`.
- Для `cp`: удаление скопированного файла/каталога.
- Для `mv`: возврат объекта в исходное место.
- Для `rm`: восстановление из временного каталога `.trash`.
- История хранится в файле `.history` и сохраняется между запусками.

Критерии оценивания (100 баллов)

- **Easy (обязательная часть)** — 70

- `ls`, `cd`, `cat` — 20

- `cp`, `mv`, `rm` — 30

- Логирование — 20

- **Medium (плагины)** — до 30

- Архивы (zip/tar) — 10

- Поиск по содержимому — 10

- История и undo — 10