Техническая документация GUI UNIX OS

1. Архитектура

GUI UNIX OS — учебное графическое приложение на pygame, эмулирующее базовые команды UNIX поверх виртуальной файловой системы, упакованной в ZIP-архив.

Поток работы:
1. Пользователь вводит /start.
2. Бот спрашивает имя и переходит в MAIN_MENU.
3. В главном меню доступны кнопки: «Кто мы», «Заповеди», «Отделы», «Словарь», «Перезапустить».
4. Выбор «Отделы» переводит в состояние DEPARTMENTS.
5. «Перезапустить» возвращает к главному меню, «/cancel» завершает диалог.

Данные (миссия, заповеди, словарь, описания отделов) хранятся в словаре TEXTS.
Клавиатуры формируются через ReplyKeyboardMarkup.

2. Основные функции

- help — показать список доступных команд.

- ls — вывести содержимое текущего каталога виртуальной ФС.

- cd <путь> — сменить каталог. Поддерживаются '/', '..' и относительные пути.

- wc <файл> — посчитать количество строк, слов и символов в файле из архива.

- mv <источник> <назначение> — переместить или переименовать файл внутри ZIP.

- clear — очистка экрана консоли.

- exit — завершить работу приложения.

Тестовая документация

1. Юз-кейсы

UC-01. Просмотр корня виртуальной ФС

- Открыть приложение с архивом systeam.zip.

- Ввести команду: ls.

- Ожидаемый результат: выведен список элементов каталога systeam/ (dir1/, so.txt, text.txt).

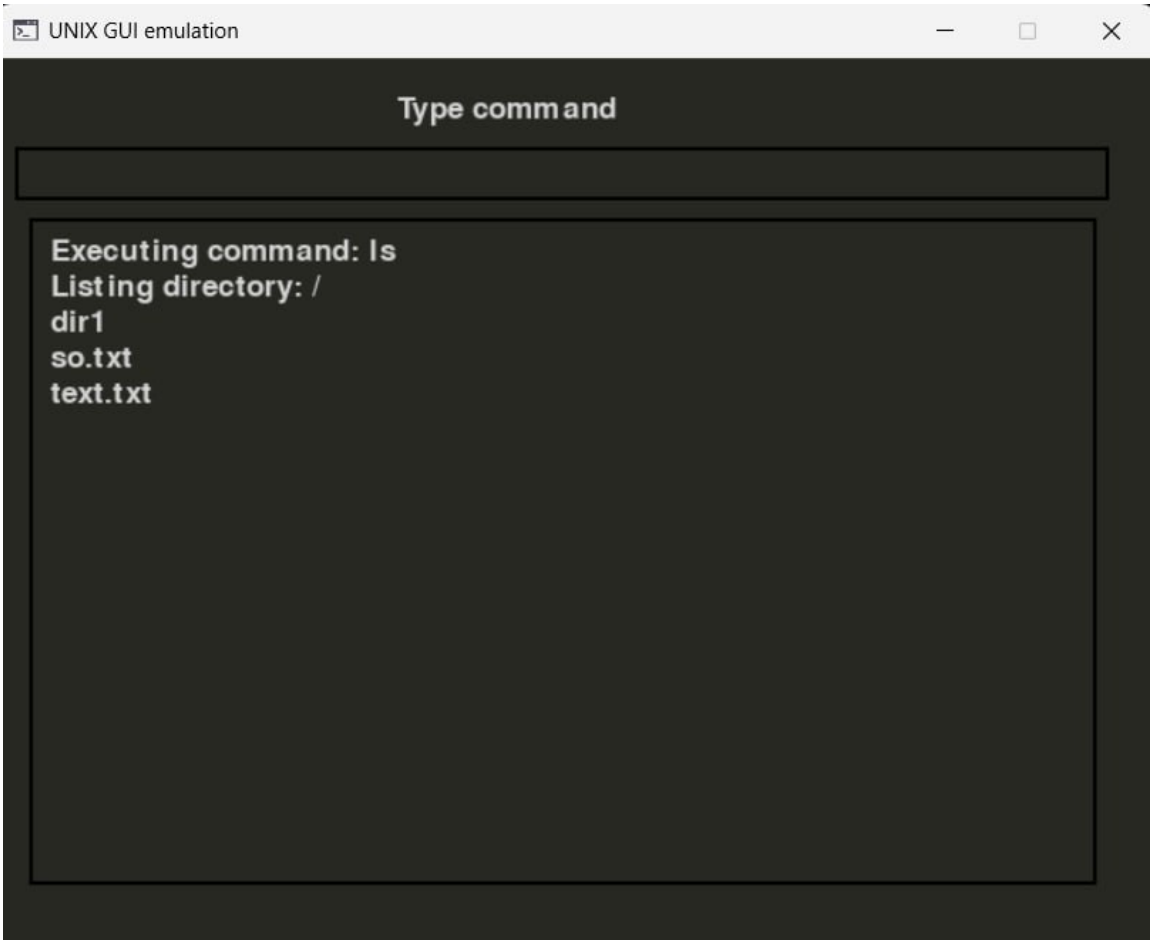

Рисунок 1 – Просмотр корня

UC-02. Переход в подкаталог и возврат

- Команда: cd dir1

- Команда: ls

- Ожидаемый результат: виден файл a.txt.

- Команда: cd ..

- Ожидаемый результат: снова каталог system/.

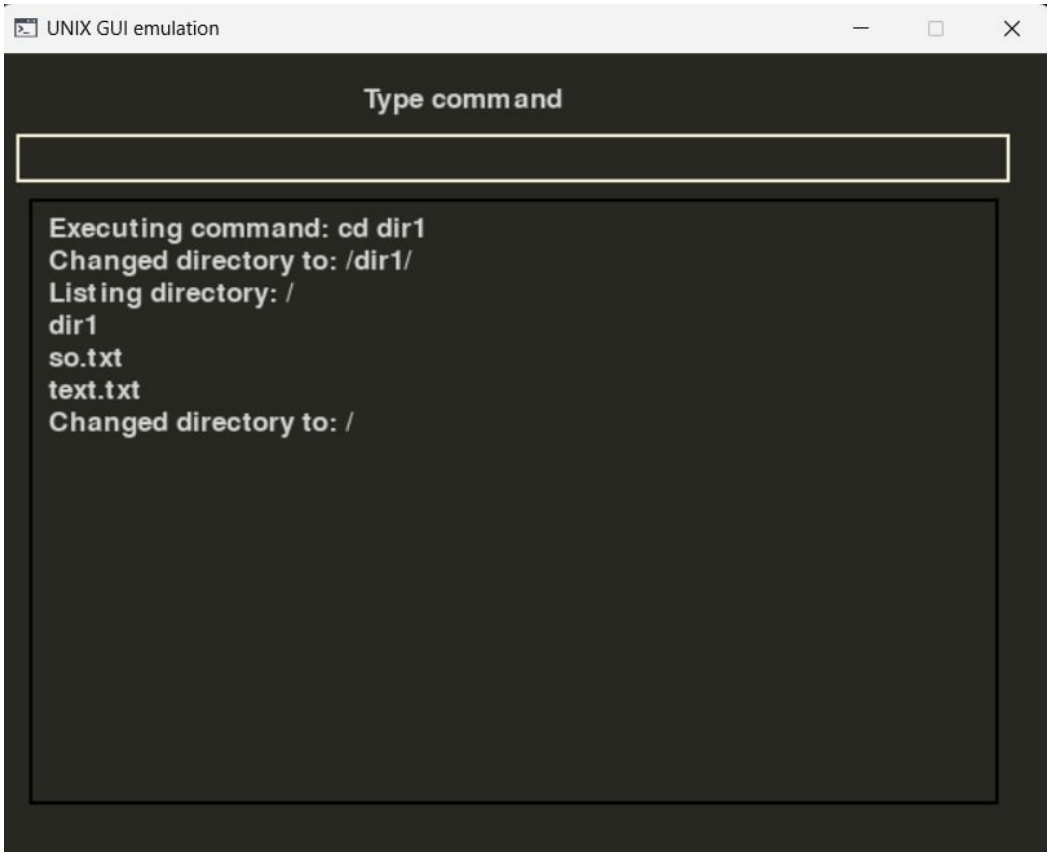

Рисунок 2– Переход в подкаталог и обратно

UC-03. Подсчёт строк/слов/символов

- Команда: wc text.txt

- Ожидаемый результат: корректные значения Lines/Words/Characters для файла в ZIP

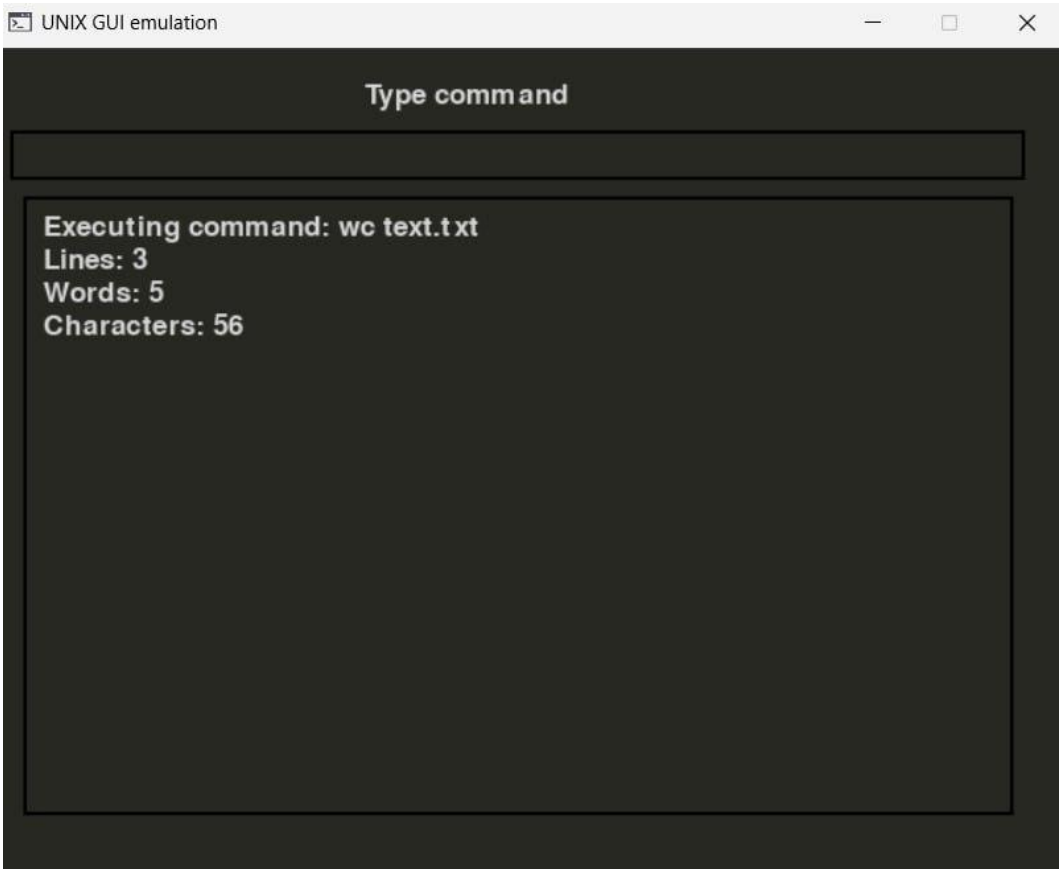

Рисунок 3– Подсчет строк и слов

UC-04. Перемещение/переименование файла

- Команда: mv text.txt dir1/text.txt

- Ожидаемый результат: файл перемещён в systeam/dir1/, подтверждающее сообщение 'Moved ...'.

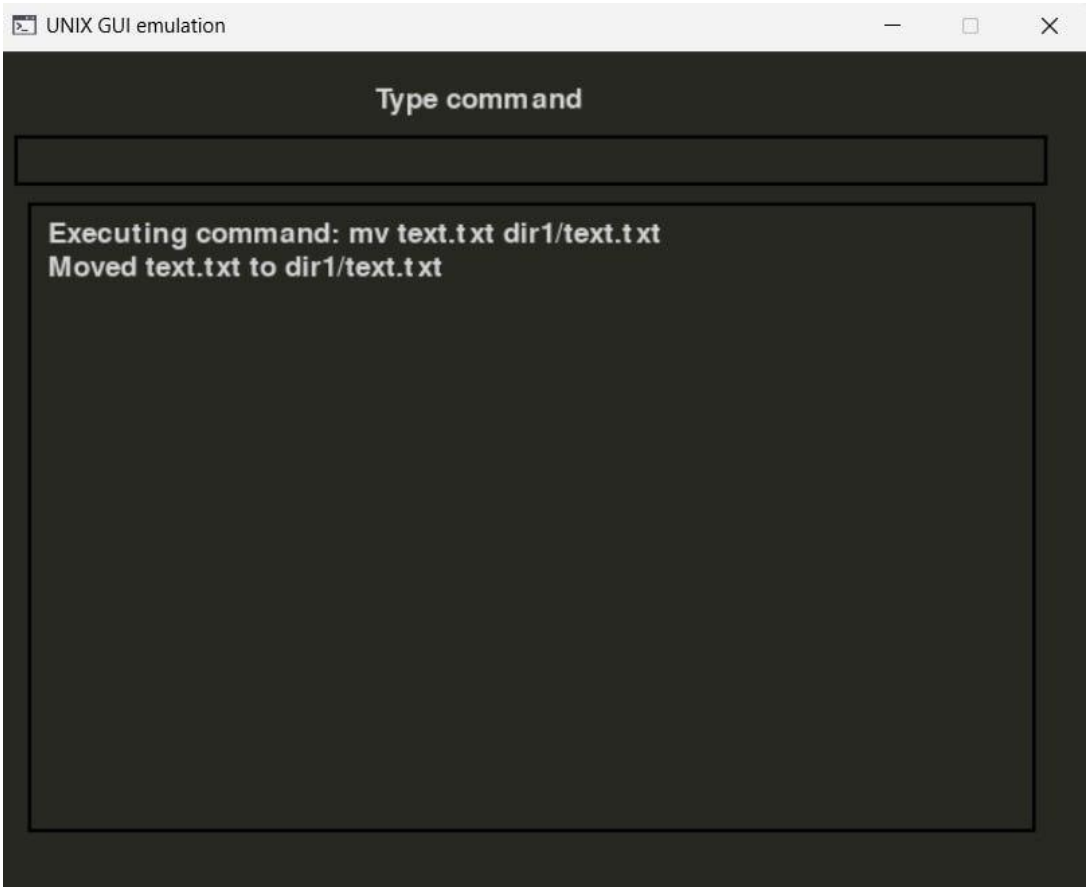

Рисунок 4– Передвижение файла

UC-05. Очистка экрана

- Команда: clear

- Ожидаемый результат: история вывода очищена.

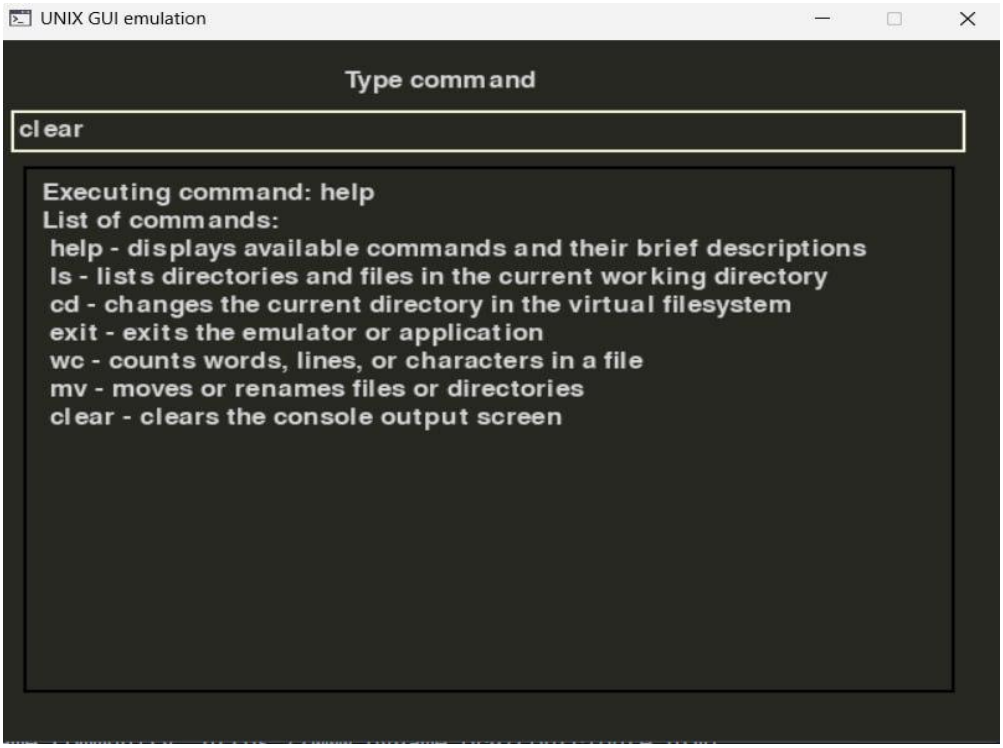

Рисунок 5– Полный экран

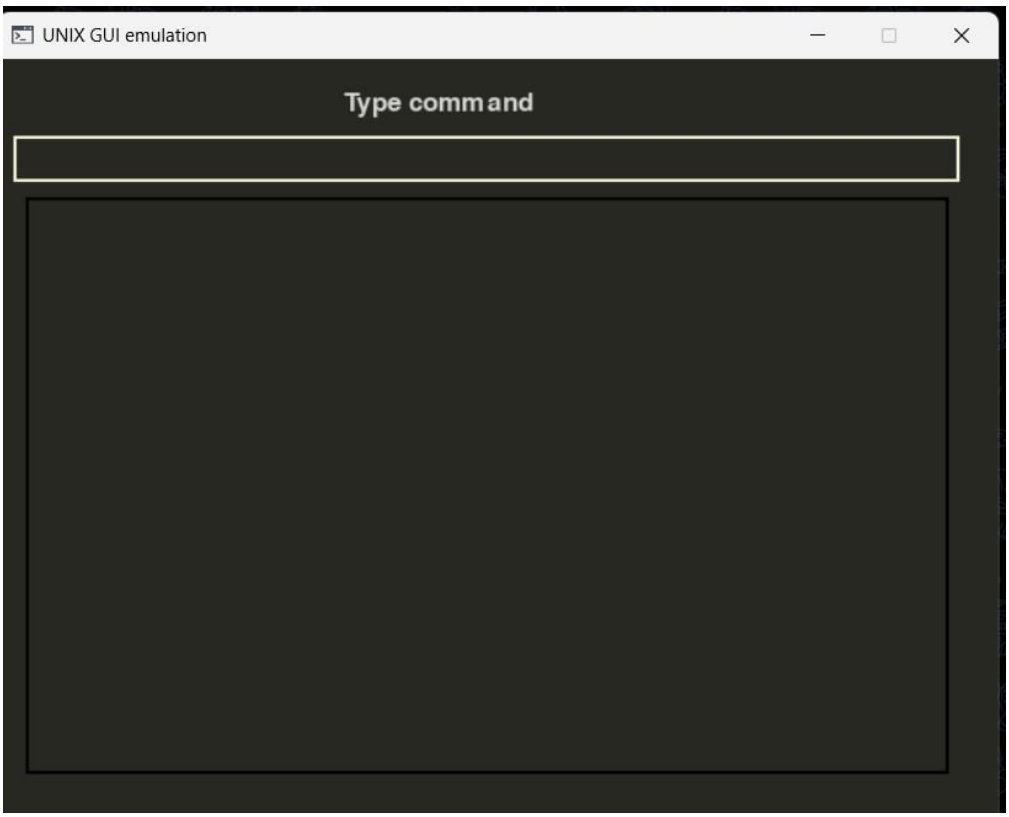

Рисунок 6– Приминение clear

UC-06. Справка

- Команда: help

- Ожидаемый результат: выведен список доступных команд.

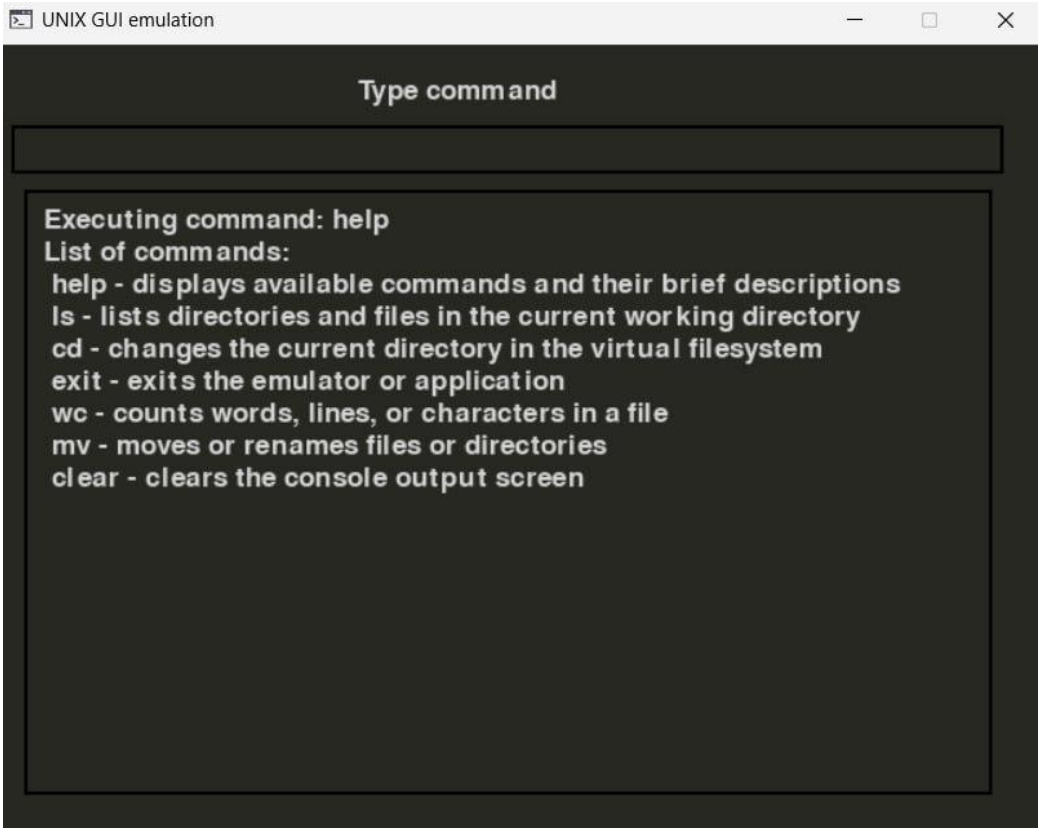

Рисунок 7– Вывод справки

UC-07. Обработка ошибок

- Команда: wc not_exist.txt → ожидается 'ERROR: File ... not found'.

- Команда: cd not_exist → ожидается 'ERROR: Directory not exist'.

- Команда: foo → ожидается 'ERROR: Invalid command'.

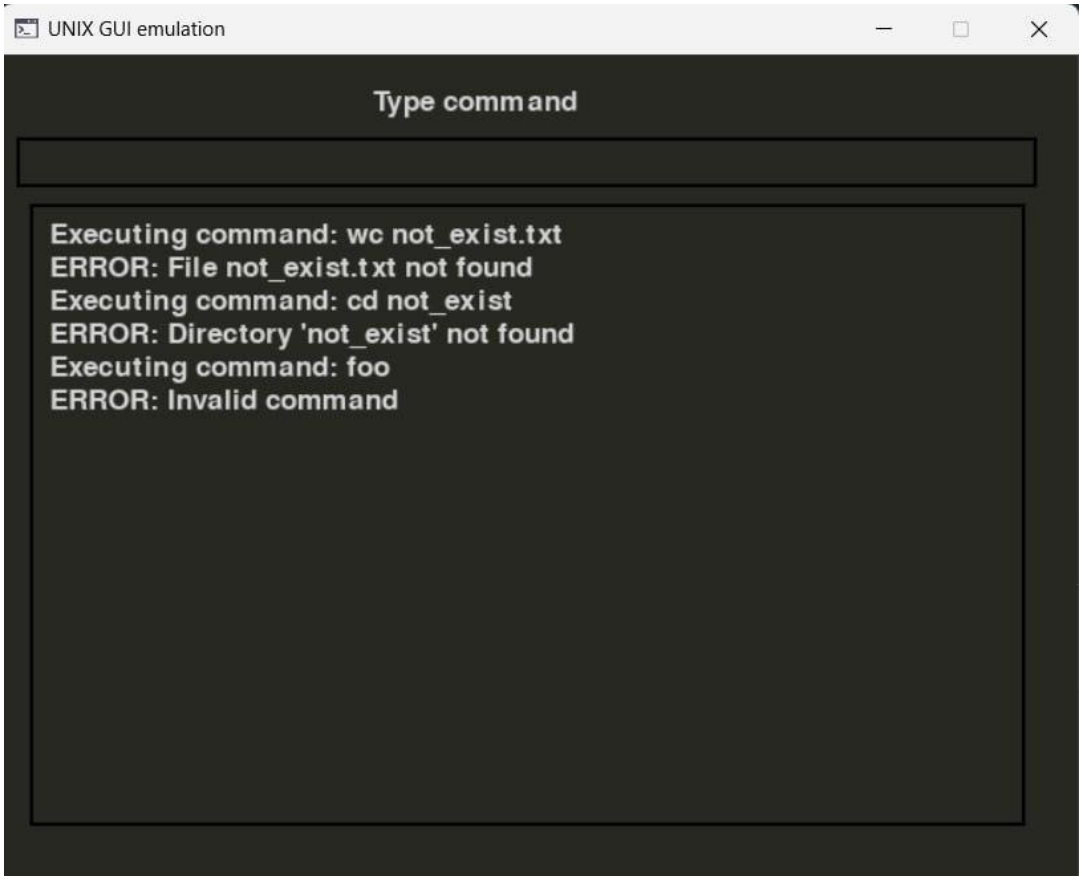

Рисунок 8– Обработка ошибок

UC-08. Автовыполнение скрипта

- Запуск с --script start.sh.

- Ожидаемый результат: в начале работы выполнены команды из файла (в поставке — 'help').

Рисунок 9– Автовыполнение скрипта

Руководство пользователя

1. Введение

GUI UNIX OS позволяет тренировочно выполнять базовые команды UNIX в изолированной среде внутри ZIP-архива; реальная файловая система не изменяется.

2. Начало работы

- 1) Установите Python 3.10+ (рекомендуется 3.11).
- 2) Создайте и активируйте виртуальное окружение:
- Windows (PowerShell):
- python -m venv .venv
- .\.venv\Scripts\Activate.ps1
- macOS/Linux (bash):
- python3 -m venv .venv
- source .venv/bin/activate
- 3) Установите зависимости:
- pip install pygame pytest
- 4) Убедитесь, что архив виртуальной ФС (например, systeam.zip) находится в корне проекта.

3. Основное меню

- Область вывода (console): история сообщений и результатов команд.
- Поле ввода (input): ввод текста команд; Enter — выполнить, Backspace — удалить символ.

4. Работа с меню отделов

- help → перечень команд.
- ls → содержимое текущего каталога.
- cd dir1 → перейти в подкаталог; cd .. → вернуться; cd / → корень systeam/.
- wc text.txt → показать Lines/Words/Characters.

- mv text.txt dir1/text.txt → перемещение/переименование.
- clear → очистка вывода.
- exit → выход из приложения

5. Решение возможных проблем

- ModuleNotFoundError: pygame — установите пакет: pip install pygame.
- Графическое окно не открывается — требуется машина с GUI/X-сервером.
- wc: файл не найден — проверьте путь и наличие в архиве.
- pytest не видит пакет core — запускайте из корня и/или задайте PYTHONPATH=.

Системная документация

1. Что вам понадобится

- Python 3.10+ (Windows/macOS/Linux).
- pygame (GUI), pytest (для тестов — опционально).
- Текстовый редактор/IDE (VS Code/PyCharm).

2. Организация файлов

- main.py — точка входа; принимает аргументы: --user, --archive, --script.
- core/ — логика приложений (console.py, emulator.py, input_box.py).
- source/ — цвета и ресурсы (img/UNIX_GUI_icon.png).
- start.sh — стартовый сценарий (по умолчанию: help).
- system.zip / test_sys.zip — примеры виртуальной ФС (корневой каталог 'system/').
- tests/test_emulator.py — модульные тесты эмулятора.

3. Работа в командной строке

Запуск приложения:

python main.py --user Alice --archive systeam.zip --script start.sh

Запуск тестов:

PYTHONPATH=. pytest -q

4. Взаимодействие с приложением

- Запустите приложение с корректным путём к ZIP и стартовому скрипту.
- После старта выполнится сценарий из --script (в комплекте — команда 'help').
- Введите команды в поле ввода и наблюдайте результаты в области вывода.