Министерство науки и высшего образования Российской Федерации

Ульяновский государственный Технический университет

Кафедра «Вычислительная техника»

Дисциплина «Операционные системы»

**Лабораторная работа №1**

**«Разработка оболочки»**

**Выполнил**:

студент гр. ИВТАПбд-31

Сокольский Р.С.

**Проверил работу:**

Беляев К. С.

Ульяновск 2025

**Цель работы.**

Разработать простую командную оболочку (shell) для операционной системы Linux, поддерживающую базовые команды, работу с переменными окружения и выполнение внешних программ.

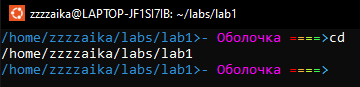
**Ход работы.**

Работа над созданием командной оболочки началась с разработки основного цикла программы, который должен бесконечно запрашивать у пользователя ввод команды, обрабатывать её и выполнять соответствующие действия. Для считывания ввода использовалась функция getline(), позволяющая получить строку произвольной длины. После считывания команда очищалась от лишних символов новой строки с помощью strcspn(), чтобы избежать ошибок при разборе аргументов.

Далее была реализована логика разбора командной строки. Использовалась функция strtok(), которая разделяла введённую строку на отдельные аргументы. В результате каждая команда представляла собой массив строк, где первый элемент был названием команды, а последующие элементы – её аргументами. Если пользователь не вводил команду (например, просто нажимал Enter), программа пропускала выполнение и ожидала нового ввода.

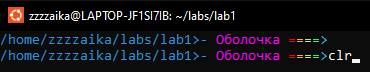
После этого были добавлены встроенные команды оболочки. В их число вошли:

* cd <directory> – смена текущего рабочего каталога. Команда использовала функцию chdir(), а в случае ошибки выводила сообщение с помощью perror(). Кроме того, после успешного изменения каталога обновлялась переменная окружения PWD через setenv().



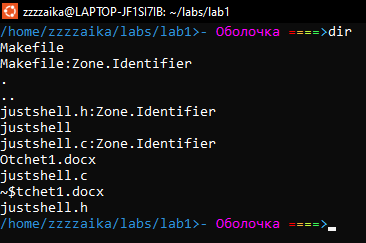
**Рис. 1 вывод команды cd.**

* clr – очистка экрана, выполняемая с помощью управляющей последовательности \033[H\033[J, которая возвращала курсор в верхний левый угол терминала и очищала его содержимое.



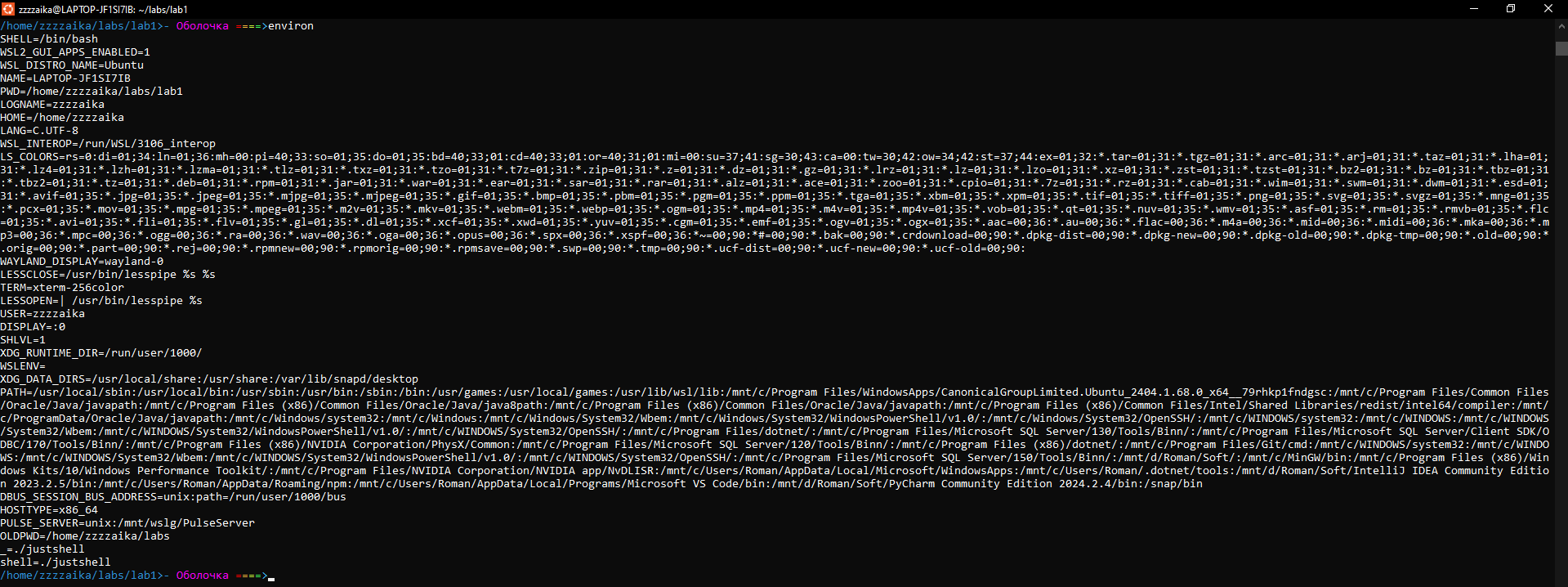
**Рис. 2 вывод команды clr.**

* dir <directory> – отображение содержимого каталога. Для этого использовалась функция opendir(), которая открывала указанный каталог, затем readdir() последовательно считывала файлы и каталоги, а closedir() освобождала ресурсы. В случае ошибки (например, если каталог не существовал) программа выводила сообщение об ошибке.



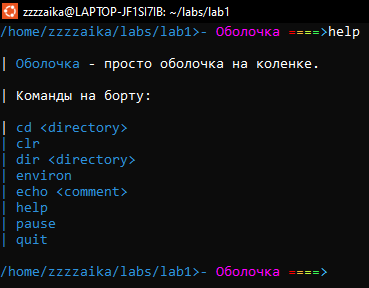
**Рис. 3 вывод команды dir.**

* environ – вывод всех переменных окружения. Использовался внешний массив environ, который хранит все переменные среды. Каждая переменная последовательно выводилась на экран.



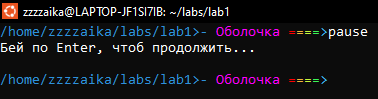
**Рис. 4 вывод команды environ.**

* echo <comment> – вывод комментария на экран. В цикле обрабатывались все аргументы команды, объединяя их в строку, которая затем выводилась через printf().
* help – отображение справочной информации о доступных командах с описанием их работы. Вывод был стилизован с использованием цветных ANSI-кодов.



**Рис. 5 вывод команды help.**

* pause – приостановка работы оболочки до нажатия клавиши Enter. Использовалась функция getchar() в цикле, чтобы программа не продолжала выполнение, пока пользователь не подтвердит продолжение работы.



**Рис. 6 вывод команды pause.**

* quit – завершение работы оболочки. Вызов exit(0) обеспечивал корректное завершение программы.

После реализации встроенных команд была добавлена поддержка выполнения внешних программ. Для этого использовался механизм создания дочернего процесса через fork(). Родительский процесс порождал новый процесс, в котором выполнялась введённая пользователем команда с её аргументами с помощью execvp(). Если команда выполнялась успешно, родитель ждал завершения дочернего процесса через waitpid(). В случае ошибки (например, если команда не существовала) программа выводила сообщение через perror().

Дополнительно было реализовано цветное приглашение командной строки, отображающее текущий рабочий каталог пользователя. Функция getcwd() получала путь текущего каталога, а затем он выводился в стилизованном виде с использованием цветных ANSI-кодов.

После завершения реализации всех функций программа была протестирована. Проверялась корректность работы встроенных команд, возможность выполнения внешних программ, обработка ошибок (например, попытка перехода в несуществующий каталог или выполнение неизвестной команды). Итоговая версия оболочки стабильно выполняла поставленные задачи и обеспечивала базовый функционал командного интерфейса.

**Вывод.**

В ходе работы была разработана и протестирована простая командная оболочка, поддерживающая базовые команды и выполнение внешних программ. Программа корректно обрабатывает пользовательский ввод, выполняет команды и выводит результаты.