

# **CS-2006**

**“OPERATING SYSTEM”**

**“PROJECT REPORT”**

**PROJECT TITLE:**

**UNIX SHELL (Combining Commands)**

**Group Members**

**Shahryar 21k3411**

**Israr Ayoub 21k4521**

**Sanaullah 21k3412**

**Theory Instructor: Lab Instructor:**

**Mr. Faisal Ahmed M. Monis**

**Introduction:**

The Project name is **“Unix Shell”.** It’s all about creating a new terminal in Linux, there are also other shells like zsh, bash etc. In this we add Different System Commands to perform Different Tasks like in Linux terminals. combined with the power and flexibility the command line offers, means that using it may be essential when we are performing different tasks.

**Features:**

* Interactive user interface
* Animated view
* Save Your Time
* Perfect for Programmers
* Command History
* Fast & Efficient

**Tools and Techniques:**

* LINUX/ UBUNTU
* C LANGUAGE
* MACROS
* LIBRARIES
* USER-DEFINED MODULES

**Motivation:**

The motivation behind creating this UNIX shell is to provide a practical example of how a simple shell can be implemented. Understanding the internals of a shell is crucial for anyone interested in operating systems, system programming, or software development in general. This project aims to demonstrate the basics of a shell, including command parsing, execution, and handling features like input/output redirection and pipes.

**Problem Statement:**

The primary challenge addressed by this project is the implementation of a functional UNIX shell capable of executing a variety of commands, handling input/output redirection, and supporting communication between commands through pipes. The code aims to create a user-friendly interface for interacting with the underlying operating system.

**Contribution:**

The code provides a basic yet functional implementation of a UNIX shell. It contributes to the understanding of fundamental shell functionalities, such as parsing commands, handling input/output, and managing processes. The inclusion of features like command history and directory changing enhances the user experience.

**Methodology:**

The implementation follows a systematic approach to building a shell. It includes components for parsing user input, handling various command types (normal commands, commands with input/output redirection, and commands connected by pipes), and managing the execution of processes. The code also incorporates error handling and features like changing directories and maintaining a command history.

**Results and Discussion:**

The UNIX shell successfully executes a variety of commands, handles input/output redirection, and supports command piping. It allows users to change directories and provides a basic command history feature. The results demonstrate a functional shell with a set of features commonly found in Unix-like shells.

**Code Screenshots:**

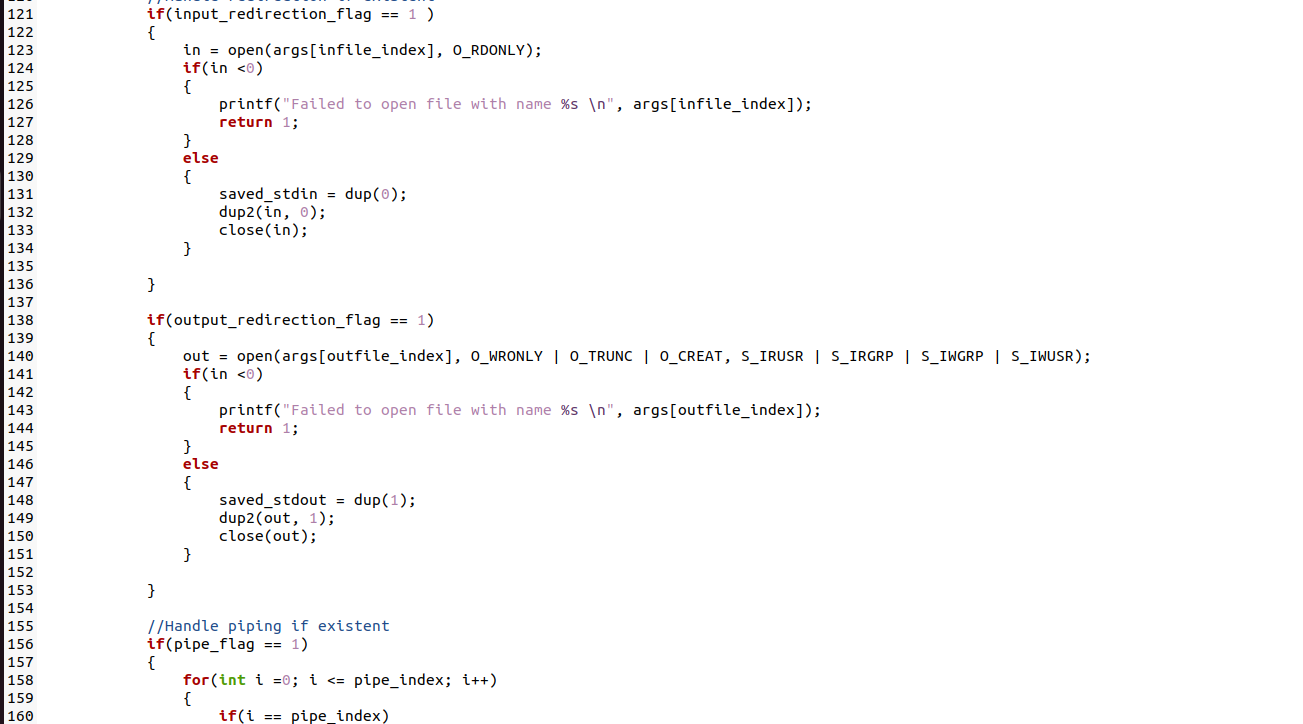
****

**A screenshot of a computer

Description automatically generated**

**A computer screen shot of a computer code

Description automatically generated**

****

**A screenshot of a computer code

Description automatically generated**

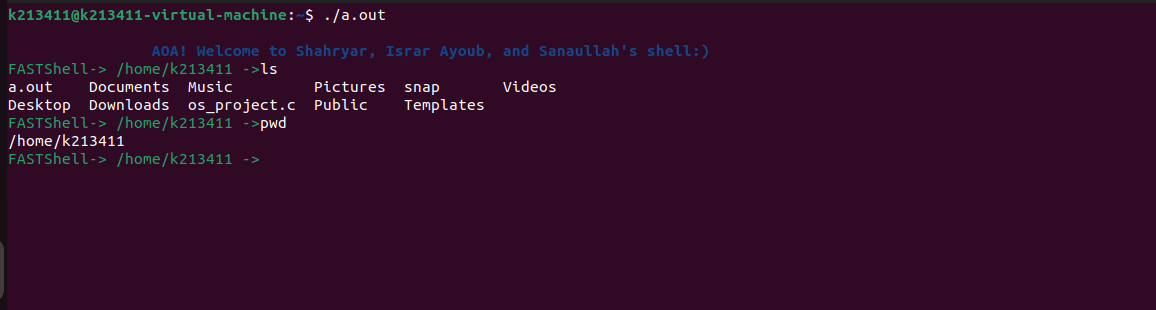
**A screenshot of a computer program

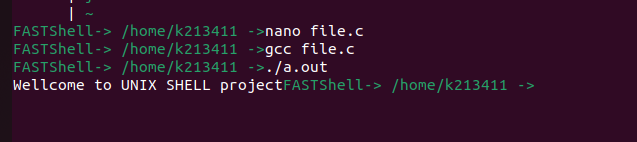
Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Output Screenshots:**





**Conclusion:**

You can see in the above screenshots how the different linux commands are running and the coloring for coloring. We have used different macros in the code part, and It is very Fast, the user interface is too good, and the initial view of the Terminal is Animated. This is just like the zsh terminal. The idea of Animation comes in our mind from zsh’s oh-my-god terminal.

**Reference:**

The development of this UNIX shell was guided by various learning resources, including the POSIX standard specifications for Unix-like operating systems. Additionally, insights from "The C Programming Language" by Kernighan and Ritchie, "Operating System Concepts" by Silberschatz, Galvin, and Gagne, and "Advanced Programming in the Unix Environment" by Stevens were instrumental in shaping the implementation. Online tutorials and documentation further enriched the understanding during the development process.