NovaShell: A Safe and Monitored Command-Line Shell

Om Srivastava

23BDS1006

December 13, 2025

Abstract (Easy Language)

NovaShell is a custom command-line shell developed in C to understand and demonstrate important operating system concepts. It allows users to execute commands like a normal Unix shell, but also adds extra features such as safety checks, command suggestions, history, and visual monitoring of CPU and memory usage. The goal of NovaShell is to make command execution safer, more informative, and easier to understand.

1 Introduction

A shell is a program that acts as a bridge between the user and the operating system. Most shells only execute commands, but NovaShell goes a step further by providing feedback about how commands affect system resources. This project helps in understanding how processes are created, how system resources are used, and how commands interact with the operating system.

2 Objectives of the Project

The main objectives of NovaShell are:

- To understand process creation using `fork()` and `execvp()`
- To learn inter-process communication using pipes
- To monitor CPU and memory usage of commands
- To add safety mechanisms for dangerous commands
- To create a modular and easy-to-extend shell

3 Main Features

3.1 Basic Shell Features

- Executes normal Linux commands

- Supports built-in commands such as `cd`, `history`, and `exit`

- Supports background execution using `&`

3.2   Pipe and Redirection Support

- Supports piping using `|` (example: `ls | wc -l`)

- Supports output redirection using `>`

3.3   Command History

- Stores previously executed commands

- Allows users to view past commands using the `history` command

3.4   Safety Layer

- Blocks dangerous commands like `rm -rf`

- Prevents accidental damage to the system

3.5   Command Suggestion System

- Detects typing mistakes in commands

- Suggests correct commands (example: `lss` → `ls`)

4   Signature Feature: Visual Resource Monitoring

This is the most unique feature of NovaShell.

4.1   What It Does

After executing each command, NovaShell shows:

- Total execution time

- CPU usage in the form of an ASCII bar

- Memory usage in the form of an ASCII bar

- Performance badge (FAST / NORMAL / SLOW)

- Warning if a command uses too many resources
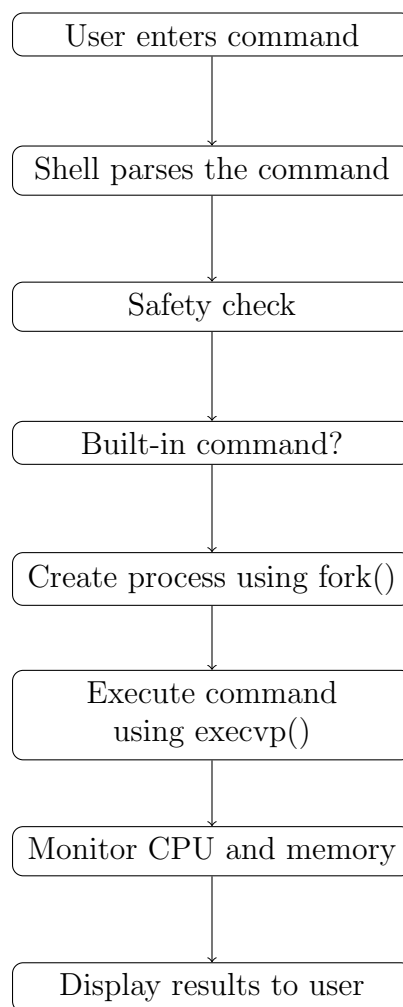
4.2   Why This Feature Is Important

- Helps users understand how commands affect system performance

- Makes the shell more interactive and educational

- Helps identify slow or resource-heavy commands

5 How NovaShell Is Different from Normal Shells

Feature	Normal Shell	NovaShell
Basic command execution	Yes	Yes
Safety checks	No	Yes
Command suggestions	No	Yes
Visual CPU/Memory monitoring	No	Yes
Performance feedback	No	Yes
Educational design	No	Yes

6 Workflow of NovaShell

Command Execution Flow

User enters command

Shell parses the command

Safety check

Built-in command?

Create process using fork()

Execute command
using execvp()

Monitor CPU and memory

Display results to user

7 Modular Design

NovaShell is divided into multiple modules:

- **main.c** – Main shell loop

- **pipe.c** – Pipe handling

- **monitor.c** – Resource monitoring

- **safety.c** – Safety checks

- **history.c** – Command history

- **suggest.c** – Command suggestions

8   Advantages of NovaShell

- Safer than normal shells

- Helps users understand system resource usage

- Easy to extend with new features

- Good learning tool for operating systems

9   Conclusion

NovaShell is a feature-rich and educational shell that demonstrates important operating system concepts in a simple and interactive way. By adding safety checks and visual monitoring, it provides better feedback to users compared to traditional shells. This project helped in understanding how shells work internally and how system resources are managed.