CSE 231: Operating System 26/08/2024 **Assignment-2 (Section-A)**

SimpleShell: A Unix Shell in C from Scratch

Group Details

Group Member 1 :

Name : Himanshu
Roll no: 2023241
Gmail ID : himanshu23241@iiitd.ac.in

Group Member 2 :

Name : Harsh Gupta
Roll no : 2023229
Gmail ID : harsh23229@iiitd.ac.in

Contribution

Himanshu :

1) Main function
2) Execute background command
3) Shell interpreter
4) Control c handler
5) Shell interpreter function

Harsh Gupta :

1) launch shell function
2) Function for clearing screen
3) Function for executing the command
4) Function for executing pip

Code Documentation

=> MainFunctions

First we launch the shell and declare the necessary variable .This will print a welcome screen for the shell name dam_shell with the current username.Also clear the terminal using the clear screen function.

- Error Handling : If username is not found the program will exit.

=> Function for clearing system

Clear command will clear the current screen

=> launch function

This function creates a new process to run a command and manages pipes when needed. It also logs the command's details, such as the process ID (PID), start time, and execution duration, into the command history.we use pipes for communication between processes if it_pip is true.

- Error Handling : we manage errors that occur during fork() and execvp() , and also verifies if memory allocation fails.

=> Control c handler function

It processes the SIGINT signal, typically triggered by Ctrl+C, by displaying the command history and then shutting down the shell.It goes through the saved command history, displaying information such as the PID, start time, and execution duration.

=> Function for executing the commands

This function splits the input command into individual tokens (arguments), checks if it's a built-in command (like exit), and hands over the command execution to the launch() function.

- Error Handling : It verifies errors that may occur during memory allocation and tokenization, specifically when using strdup() and realloc(). If an error is detected, appropriate actions are taken to handle the failure.

=> Function for executing pip
This function processes piped (|) commands by splitting them into sub-commands, trimming spaces, and executing them in a pipeline. It updates the command history and returns the status of the last command.

- Error Handling : The function includes error handling to check for memory allocation failures while splitting and processing commands. In case of a failure, it manages the error to avoid crashes or data loss.

=> Function for executing background command (bonus)
This function handles background commands marked by an ampersand (&). It splits the command into sub-commands, executing them sequentially. The command history is updated after each execution, and the function returns the status of the last background command.

- Error Handling : The function handles memory allocation errors during sub-command splitting and execution. It mitigates failures to avoid disruptions, though true background execution using fork() without wait() is not implemented.

=> Shell interpreter function

This function reads a shell script file line by line and executes each line as a command. It updates the command history and returns 1 on success. If the file can't be opened, it prints an error message and returns 1. Each line is executed using execute_background_commands().

Github private repository link:

https://github.com/himanshu23241/Operating_System-CSE231