ReadMe File of (Basic Shell Designing)

The main function contain simple infinte while loop where it calls **simple_shell() function** creating basic terminal kind space where it takes input and shows output and the infinte while loop is used only to take the input command and ouput the certain action repeatively till exit command is called.

Inside the simple_shell() there are two variables commands which is a pointer to char (a string) and parsedcommads is the another variable which is a pointer to pointer pointing a char (so basically a 2d array) and we allocate space for commands and read the input into it by using **getline function** and immediate next if statements are for error condition in case if character read (noc) is -1 the shows the error.

The input taken is a single line string and we need to parse it inorder to send it as an arrgument for execvp function and to do it we are using the **tok()** where the parameter passed is the commands or the string and to parse it and store it into a 2d array or array of strings. We need to use **strtok() function** it takes in first call the string or the address pointing to the first character of string and a delimter (a 1d array and address of first element is sent) as parameter and delimter should contain " \n\t\r"" where gap is space and this delimter specify that the parsing is to done based on these character. After the string is parsed by strtok in the loop where NULL is the first parameter in the strtok and in each iteration its address is stored in ret[i]. Then the address stored in ret is returned.

Back in simple_shell() we have 2d array named parsedcommads and parsedcommands[0] contains the argument name and subsequent parsedcommads[i] contains either options or filenames or plain text. Now, we need to call command_initialiser function.

Inside the function depending upon the different commands contained in args[0] different function is called.

If the args[0] is echo: then the funtion named echo is called and args (2d array containing the parsed string) is sent as parameter and using printf the text is displayed on to the terminal.

If the args[0] is pwd: then the function named PWD is called and using the inbulit function getcwd the current working directory is stored in buf as shown in code and if the return value of getcwd is NULL then the error is printed and for specific error where the size of buf is not sufficient for the storage of the curremt working directory value the we used statment like errno == ERANGE and error message is printed else the current directory's path is printed

If the args[0] is cd: then the function named cd is called with args as parameter and if only cd is entered as input without option the directory is changed using chdir() and parameter are getenv("HOME") because it return path to home directory and it function chdir is not executed then perror is used to print error and if cd is passed with option like (.. or folder name) then using chdir(args[1]) is executed.

If the args[0] is rest of commands then the function execute is called and args is passed as parameter and inside the function a child process is created using fork function and the parent process is put in wait by wait function and inside the child process execvp function is used to execute the commands by passing args[0] and args.

If the args[0] is exit: then if calls the exitfunc() and this function contains only a line exit (0) where it exits the program to terminal.

Commands mkdir and ls

```
darsh@darsh-VirtualBox:~/Desktop/OS2022_5sem$ ./test
simpleshell>>mkdir new
mkdir: cannot create directory ‘new’: File exists
simpleshell>>mkdir hello
simpleshell>>ls
assignment1.pdf  hello  new  shell.c  test  test.c
simpleshell>>ls -l
total 52
-rw-rw-r-- 1 darsh darsh 14051 Nov 16 11:40 assignment1.pdf
drwxrwxr-x 2 darsh darsh  4096 Nov 16 21:49 hello
drwxrwxr-x 2 darsh darsh  4096 Nov 16 21:46 new
-rw-rw-r-- 1 darsh darsh  3555 Nov 15 13:04 shell.c
-rwxrwxr-x 1 darsh darsh 17032 Nov 16 21:43 test
-rw-rw-r-- 1 darsh darsh  2629 Nov 16 21:43 test.c
simpleshell>>ls -ls
total 52
16 -rw-rw-r-- 1 darsh darsh 14051 Nov 16 11:40 assignment1.pdf
 4 drwxrwxr-x 2 darsh darsh  4096 Nov 16 21:49 hello
 4 drwxrwxr-x 2 darsh darsh  4096 Nov 16 21:46 new
 4 -rw-rw-r-- 1 darsh darsh  3555 Nov 15 13:04 shell.c
20 -rwxrwxr-x 1 darsh darsh 17032 Nov 16 21:43 test
 4 -rw-rw-r-- 1 darsh darsh  2629 Nov 16 21:43 test.c
simpleshell>>sdcsv
command is invalid :: No such file or directory
simpleshell>>ls -svw
ls: option requires an argument -- 'w'
Try 'ls --help' for more information.
```

Cat command

```
simpleshell>>cat test.c
#include<stdio.h>
#include<string.h>
#include<unistd.h>
#include<stdlib.h>
#include<sys/wait.h>
#include <errno.h>

#define TEST_BUFFER_SIZE 100

void command_initialiser (char **args);

void exitfunc ();
void echo (char **args);
void PWD ();
void cd (char **args);

void execute(char **args){
        int pid;
        pid = fork();
        if (pid == 0){

                if (execvp(args[0],args) == -1){
                        perror("command is invalid :");
                }

        }
        else{
```

Commands pwd, echo,date,rm,exit

```
}simpleshell>>pwd
/home/darsh/Desktop/OS2022_5sem
simpleshell>>echo csc  erve "" dvvev "dcdver      evee
csc erve dvvev dcdver evee
simpleshell>>date
Wednesday 16 November 2022 09:51:08 PM IST
simpleshell>>rm -rf hello
simpleshell>>ls
assignment1.pdf  new  shell.c  test  test.c
simpleshell>>exit
simpleshell>>exit
darsh@darsh-VirtualBox:~/Desktop/OS2022_5sem$
```

cd command

```
simpleshell>>cd
simpleshell>>pwd
/home/darsh
simpleshell>>cd ..
simpleshell>>pwd
/home
```