Operating System Project

Simple shell

Member list

MSSV	Full Name	Email	Contact
1712045	Nguyễn Thái Hoc	nguyenthaihocxm1999@gmail.com	0974748525
<MSSV2>			
<MSSV3>			

1 Description

Some relevant function

void cd_command(char *args[], int num_args);
--> The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory.
int check_exclamation_mark(char *command, HistoryCommands *history);
--> Check exclamation mark exists or not, and catch some errors.
void copy_command(char *command, HistoryCommands *history);
--> Print the newest command from history.
int split_token(char *command, char *args[]);
--> This will require parsing what the user has entered into separate tokens and storing the tokens in an array of character strings.
void store_command(char *command, HistoryCommands *history);
--> Append command into history.
int check_pipe(int num_args, char *args[]);
--> Check pipe in command.

fork(): Fork system call is used for creating a new process, which is called ***child process***, which runs concurrently with the process that makes the fork() call (parent process). (read more : https://www.geeksforgeeks.org/fork-system-call/)
gets(char* s): Reads characters from the standard input (stdin) and stores them as a C string into str until a newline character or the end-of-file is reached
fgets(char *s, int n, stream): It reads a line from the specified stream and stores it into the string pointed to by str. It stops when either (n-1) characters are read, the newline character is read, or the end-of-file is reached, whichever comes first.
strcmp(const char *str1, const char *str2): The strcmp() function is used to compare two strings two strings str1 and str2. If two strings are same then strcmp() returns 0, otherwise, it returns a non-zero value.
excevp(const char *file, char *const argv[]): Using this command, the created child process does not have to run the same program as the parent process does.
dup2(fildes, fildes2): System call creates a copy of a file descriptor. it uses the descriptor number specified by the user.
open(const char *path, int oflags): is a system call that is used to open a new file and obtain its file descriptor.

strcpy(char * destination, const char * source): Copies the C string pointed by *source* into the array pointed by *destination*, including the terminating null character (and stopping at that point).
pipe(int fds[2]): a pipe is a connection between two processes, such that the standard output from one process becomes the standard input of the other process. (Read more : https://www.geeksforgeeks.org/pipe-system-call/)

2 Assignment:

Tasks	Complete (%)
1. Creating the child process and executing the command in the child	100%
2. Providing a history feature	100%
3. Adding support of input and output redirection	100%
4. Allowing the parent and child processes to communicate via a pipe	100%