

CSE231 - Operating Systems

Assignment-1.2

Name: Ananya Lohani
Roll Number: 2019018
Branch: CSE

Organisation:

The structure of the /Part-2 directory is given below:

```
.
├── Makefile
├── external
│   ├── cat_cmd.c
│   ├── date_cmd.c
│   ├── ls_cmd.c
│   ├── mkdir_cmd.c
│   └── rm_cmd.c
├── internal
│   ├── internal.c
│   └── utils.c
├── main.c
└── text_files
    ├── cd_help.txt
    ├── history.txt
    └── pwd_help.txt
```

- All internal commands (except exit and pwd) are written as functions in internal.c placed in /internal directory.
- The file utils.c inside /internal contains those functions which are used to perform miscellaneous operations in main.c.
- All external commands are written as individual programs and their source codes are placed in /external directory.
- The /text_files directory contains the text files required in the program.
- main.c is the main. The files internal.c and utils.c are included in this file along with other header files and ANSI escape codes for the formatted output on the terminal. main.c also calls fork(), execv() and wait() system calls for executing the external commands.

- Makefile generates the executable file called shell after compiling main.c and all the external command files. The executables of the external commands are stored in /external directory, while shell is stored in the current directory itself.

Working of the program:

The main() function in main.c runs an infinite loop which prompts the user to enter a command and checks if the command is valid. If it's invalid then graceful degradation occurs. If it's an internal command the program runs the internal commands handled by the shell itself. If it's an external command then the program calls fork() and spawns a child process which calls execv() and executes the external command's corresponding program and then terminates the child process. If there's any error in reading the text files in the directory /text_files, then complete termination of the program occurs with an exit status of -1. Additionally, if any error is encountered in the execv() system call then complete termination occurs with an exit status of -1.

Commands:

Internal

cd

The cd command is used to change the current working directory. I have implemented cd to support all types of valid arguments that the cd command in Linux can take.

Options:

- -P : use the physical directory structure without following symbolic links: resolve symbolic links in DIR before processing instances of `..'
- --help : displays this help page of cd, as stored in the file `text_files/cd_help.txt'; displays all the options available.

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Too many arguments: If more than 2 arguments are passed
- Invalid command: If an invalid option is used
- Could not open `text_files/cd_help.txt': If there was a problem opening cd_help.txt, exit with status -1

- No such file or directory: If the mentioned directory does not exist
- Failed to open directory: If the mentioned argument is not a directory or if there was a problem opening it

Assumptions:

- The user does not try to open a directory whose name begins with '-'.
- If no option is passed, the user does not pass more than 1 argument.
- By default, cd is implemented as if '-P' were specified.

echo

The echo command is used to write commands to stdout.

I have implemented echo to support all types of valid arguments that the echo command in Linux can take.

Options:

- -n : Do not print the trailing newline character.
- -E : disable interpretation of backslash escapes (default).

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Missing argument: If no arguments are passed
- Invalid command: If an invalid option is used

Assumptions:

- If the argument contains double quotes('\"'), it's always used in pairs.

history

Display the history list with line numbers. An argument of N lists only the last N entries.

Options:

- -c : clear the history by deleting all of its entries
- -d [OFFSET] : delete the entry at position OFFSET.

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Too many arguments: If more than the required arguments are passed
- Missing argument: If no value of offset is passed with -d
- Invalid command: If an invalid option is used
- Could not open `text_files/history.txt': If there was a problem opening history.txt, exit with status 1
- Invalid value of offset: If a non-numeric value is passed as offset

Assumptions:

- The history is always stored in the file text_files/history.txt

pwd

Print the name of the current working directory.

Options:

- -P : print the physical directory, without any symbolic links
- --help : displays the help page of pwd, as stored in the file `pwd_help.txt'; displays all the options available.

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Too many arguments: If more than 2 arguments are passed
- Invalid command: If an invalid option is used
- Could not open `text_files/pwd_help.txt': If there was a problem opening pwd_help.txt, exit with status -1

Assumptions:

- By default, pwd is implemented as if `-P' were specified.

exit

Exit the shell with a status of the least significant byte of N specified by an argument

Options:

None

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Too many arguments: If more than 2 arguments are passed

- Numeric argument required: If a non-numeric value is passed the argument

Assumptions:

- By default, the shell exits with status 0.

External

ls

List all the non-dot files and directories present in the mentioned as argument(s).

Options:

- -m : list all the entries separated by a comma
- -D : list all the entries enclosed in double quotes

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Invalid command: If an invalid option is used
- No such file or directory: If the mentioned directory does not exist

Assumptions:

- The user does not try to pass a directory whose name starts with '-' as an argument.

cat

Concatenate file(s) to standard output. With no file, or when file is -, read standard input until eof character is encountered(ctrl+D on the keyboard).

Options:

- -n : number all output lines
- -E : display '\$' at the end of each line

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Invalid command: If an invalid option is used

- Could not open file: If the filename given doesn't exist or cannot be opened

Assumptions:

- The user does not try to pass a file whose name starts with '-' as an argument.

date

Display the current time in the given format. I have implemented date to support all types of valid formats that the date command in Linux can take.

Options:

- -u : display UTC in the given format
- -R : display the date and time in the RFC 5322 format

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Too many arguments: If more than 2 arguments are passed
- Invalid format: If an invalid format is passed(All formats must begin with '+' e.g. "+%c")
- Invalid command: If an invalid option is used

Assumptions:

None

rm

Remove (unlink) the file(s).

Options:

- -v : print a message for each removed file
- -i : prompt before every removal

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Invalid command: If an invalid option is used
- Permission denied: If the shell does not have permission to delete the file

- Is a directory: If the mentioned argument is a directory
- No such file or directory: If the mentioned file does not exist
- Missing argument: If no arguments are passed

Assumptions:

- The user does not try to pass a file whose name starts with '-' as an argument.

mkdir

Create the directory(ies), if they do not already exist.

Options:

- -v : print a message for each created directory
- -p : no error message displayed if directory already exists

Corner Cases/Errors Handled:

In all these cases, an error message is displayed with graceful degradation(unless stated otherwise):

- Invalid command: If an invalid option is used
- Permission denied: If the shell does not have permission to create a directory
- Is a directory: If the mentioned directory already exists
- No such file or directory: If the mentioned path is not valid
- Missing argument: If no arguments are passed

Assumptions:

- The user does not try to pass a directory whose name starts with '-' as an argument.

Test Cases:

```
Welcome to lohanish!
      Commands: cd, echo, history, pwd, exit, ls, cat, date, rm, mkdir
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ cd ..
lohanish: /home/lohanis/Downloads/Assignment-1$ ls
Part-2      Part-1
lohanish: /home/lohanis/Downloads/Assignment-1$ mkdir test case "test case"
lohanish: /home/lohanis/Downloads/Assignment-1$ ls
Part-2      test case      case      test      Part-1
```

```
lohanish: /home/lohanis/Downloads/Assignment-1$ cd Part-2
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ ls
test.txt      internal      main.c       Makefile     external     text_files   shell
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ rm -v test.txt
removed test.txt
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ ls -D
"internal"    "main.c"     "Makefile"   "external"   "text_files" "shell"
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ date
Wed Sep 30 20:00:40 2020
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ date -u
Wed Sep 30 14:30:47 2020
```

```
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ date "+%A"
Wednesday
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ history -c
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ history
1 history
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ pwd
/home/lohanis/Downloads/Assignment-1/Part-2
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ cd ~
lohanish: /home/lohanis$ pwd
/home/lohanis
lohanish: /home/lohanis$ history
1 history
2 pwd
3 cd ~
4 pwd
5 history
```

```
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ echo abc def ghi
abc def ghi
lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ cat
hello
hello
this
this
is
is
a
a
test
test
case
case

lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ cat -n
demo
      1 demo
with
      2 with
-n
      3 -n

lohanish: /home/lohanis/Downloads/Assignment-1/Part-2$ exit
```