

Name: Parveen

Roll No. 2021079

Documentation:

Options handled: (Individual Details below)

cd: cd, cd ~, cd / , cd .. , cd path

pwd: pwd, pwd -L , pwd -P

echo: echo simple, echo -n , echo -help

ls : ls, ls -a , ls -m

date: date, date -u, date -r filename

cat : cat simple, cat -n, cat -e

rm : rm simple, rm -i , rm -v

mkdir : mkdir simple, mkdir -p , mkdir -v

Assumptions:

1. Command length less than 1024
2. Echo content less than 400
3. Cd command user not permanently fixing the directory
4. Directories name with spaces between them such as (New Folder, VS T) may not work with thread as system call treat the path as two different arguments.
5. Path length , filename and directory names should be less than 250
6. Continuous superuser access is required while testing (else if on changing directories, external command may not work properly).

General error and bugs handled:

Handling of stack smashing if happen in thread call

Handling of command length buffer overflow

Handling of general execution error in thread

Handling of fork call failure

Handling of invalid garbage input to every command(ls,cat..)

Handling of blank line

Handling of getpwd buffer overflow

Detailed Description (Errors, bugs, corner cases, normal commands):

pwd: corner cases and attacks

Handling of pwd garbage command

cd: corner cases and attacks

Handling of cd garbage (Error reported in this case)

Handling of cd more than 1 arguments (Error reported in this case)

Handling of cd /pathname/

Handling of cd pathname/

Handling of cd ..
Handling of cd ~
Handling of cd /
Handling of only cd
Handling of all above cd commands with &t also

echo: corner cases and attacks

Handling of echo val val2 val3 (input having different spaces in between)
Handling of echo *
Handling of echo -n option (if operand missing also handled no error)
Handling of -help option
Handling of only echo

ls: corner cases and attacks

Handling of ls garbage (Error reported in this case)
Handling of ls /pathname/ (with -a, -m options) (if directory not found error reported)
Handling of ls /pathname1/ /pathname2/ (case where multiple paths are given (more than 1) to show their combined list) (if any error in one(or more) path error is reported respectively)
Handling ls multiple paths with -a and -m option
Handling of case where delimiters such as \n \t are appended in command (error is reported)
Handling of input having different spaces in between (for eg: ls -a /path/ /path2/)
Handling of ls -a only
Handling of ls -m only
Handling of ls only
Handling of all above commands with &t (with respected errors)

date: corner cases and attacks

Handling of date garbage (Error reported in this case)
Handling of case where arguments order is changed (Error reported in this case)
Handling of input having different spaces in between
Handling of operand missing in date -r (Error is reported)
Handling of case where path is given for file and not directly filename to show in -r case (if any error found in path error is reported respectively)
Handling of date -u
Handling of date only
Handling of all above commands with &t (with respected errors)

cat: corner cases and attacks

Handling of cat garbage (Error reported in this case)
Handling of cat name1 (with -n, -e options) (if not found error reported)
Handling of cat name1 name2 (case where multiple files are given (more than 1) to show their combined data) (if any error in one(or more) path error is reported respectively)
Handling with multiple files with -n and -e option

Handling of case where paths are given for files and not directly filename to show their content (if any error in one(or more) path error is reported respectively)

Handling the above path case with multiple paths or combination of both

Handling of case where delimiters such as \n \t are appended in command (Error reported)

Handling of input having different spaces in between.

Handling of operand missing in -n and -e case (Error is reported)

Handling of cat only (Error reported in this case)

Handling of all above commands with &t (with respected errors)

rm: corner cases and attacks

Handling of rm garbage (Error reported in this case)

Handling of rm name1 (with -i, -v options) (if not found error reported)

Handling of rm name1 name2 (case where multiple files are given (more than 1)) (if any error in one(or more) filename error is reported respectively)

Handling with multiple files with -i and -v option

Handling of case where paths are given for files and not directly filename (if any error in one(or more) path error is reported respectively)

Handling the above path case with multiple paths or combination of both

Handling of case where delimiters such as \n \t are appended in command (Error reported)

Handling of input having different spaces in between.

Handling of operand missing in -i and -v case (Error is reported)

Handling of rm only (Error reported in this case)

Handling of all above commands with &t (with respected errors)

mkdir: corner cases and attacks

Handling of mkdir garbage (Error reported in this case)

Handling of mkdir name1 (with -p, -v options)

Handling of mkdir name1 name2 (case where multiple folders are given (more than 1) to create) (if any error in one(or more) error is reported respectively)

Handling with multiple items with -p and -v option

Handling of case where paths are given for folders and not directly folder name to create in folder in the given path (if any error in one(or more) path error is reported respectively)(not for -p case , path cannot work for default directory such as /home/...)

Handling the above path case with multiple paths or combination of both

Handling of input having different spaces in between.

Handling of mkdir only (Error reported in this case)

Handling of all above commands with &t (with respected errors)

Test cases: (Note: ensure existence of directories and files in system mentioned here)

1. Blank input
2. pwd , pwd -L , pwd -P
3. cd , cd ~ , cd / , cd /home/parveen/Downloads
4. echo , echo -n hi how are you , echo -help , echo hi how you
5. ls , ls -a , ls -l , ls /boot/

6. cat file1, cat -n file1 , cat -e file1
7. rm file1, rm -i file1 , rm -v file1
8. mkdir dir1 , mkdir -p dir1/dir2 , mkdir -v dir1
9. help (to see help)

System and commands description:

1. Fork call is used to create new process, return value is checked and then process is executed.
2. wait() call is used to ensure that parent process run only after child process termination.
3. System call is used to execute command for threads. (command is passed as argument to execute another binary file).
4. Execvp call is used to execute command in fork process in child process. (as new process is created in place of existing by execvp command).
5. Pthread_create call is used for creation of threads and working function, and its arguments are passed in it.
6. Pthread_join call is used to ensure to wait till the termination of newly created thread.
7. Chdir call is used to execute cd commands directly in shell.
8. Getcwd call is used to execute pwd commands directly in shell.
9. Respected header files are included directly in main file.
10. Special variable type pid_t is used to store process id.
11. Return values are checked for every syscall to know that execution is failed or passed, custom args structure is made to pass multiple argument to function.

Commands description:

cd: to go to home directory

cd ~ : to go home directory

cd .. : to go one directory back

cd pathname: to change current directory to specified path

pwd: to get current working directory

pwd -L : Prints the symbolic path.

pwd -P : Prints the actual path.

echo: to print text

echo -n: to print text without /n as termination

echo --help: to get echo help option of bash

ls: to list items in current directory

ls -a: list all files including hidden files starting with '.'

ls -m: list items separated with comma

cat: to display file content on screen

cat -n : display file content with lines numbers

cat -e: display file content with appending \$ to end of every line

date: to display current system date

date -u: to display date in UTC format

date -r filename: to display last modified date of that file

rm: to remove files

`rm -i` : removes files with interrupt asking confirmation for every file

`rm -v` : to remove files and display a status message for each(for success or failure)

`mkdir`: to create new directories

`mkdir -p`: create new directories, if not present already also used to create directory inside directory(concurrent directories)

`mkdir -v`: creates new directories and display a status message for each(for success or failure)