

Raghav Sakhuja
2021274

Documentation

Custom Shell made in c using in-build libraries and system calls which supports 3 internal and five external commands. It uses standard C libraries along with `fork()`, `exec()` family system calls and `wait()` to provide multiprocessor support.

It Provides Multi Thread support using `pthread_create()`, `pthread_join()` and `system()` system calls.

Internal Commands

- **pwd**

NAME

`pwd` - print name of current/working directory

SYNOPSIS

`pwd [-P/-L]`

`-L`, use `PWD` from the environment, even if it contains symlinks

`-P`, avoid all symlinks

ERRORS HANDLED

1. If extra characters are typed after `pwd`, it ignores them to print the current path
2. If a wrong flag is used, it prints the error and returns without giving the current path

- **cd**

NAME

cd - Change working directory

SYNOPSIS

cd ?dirName?

DESCRIPTION

Change the current working directory to dirName,

ERRORS HANDLED

1. If invalid directory is provided, print the error message and do not changes the directory.
2. If a wrong flag is used, it prints the error and returns without giving the current path

• echo

NAME

echo - display a line of text

SYNOPSIS

echo [OPTION] [STRING]

DESCRIPTION

Echo the STRING(s) to standard output.

-n do not output the trailing newline

-e enable interpretation of backslash escapes

If -e is in effect, the following sequences are recognized:

\\ backslash

\t horizontal tab

\c produce no further output

\n new line

ERRORS HANDLED

1. In case of unclosed quotations, prompts the user to input multiple lines until quotations are ended.
2. If a wrong flag is used, it prints the error and returns without giving the current path

External Commands

- **ls**

NAME

ls - list directory contents

SYNOPSIS

```
ls [OPTION]... [FILE]...  
ls &t [OPTION]... [FILE]...
```

DESCRIPTION

List information about the FILES (the current directory by default). Sort entries alphabetically.

-a do not ignore entries starting with .

-1 list one file per line.

ERRORS HANDLED

1. In case of directory that does not exist, gives the error and does not print any files.
2. In case of an unreadable directory, print the error message and do not print any files.
3. If a wrong flag is used, it prints the error.

- **mkdir**

NAME

`mkdir` - make directories

SYNOPSIS

```
mkdir [OPTION]... DIRECTORY...  
mkdir &t [OPTION]... DIRECTORY...
```

DESCRIPTION

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

`-p`, no error if existing, make parent directories as needed

`-v`, print a message for each created directory

ERRORS HANDLED

1. In case of pre-existing directories, do not remake them. Instead, print the error.
2. If a wrong flag is used, it prints the error and returns without creating a directory.

• **cat**

NAME

`cat` - concatenate files and print on the standard output

SYNOPSIS

```
cat [OPTION]... [FILE]...  
cat &t [OPTION]... [FILE]...
```

DESCRIPTION

Concatenate FILE(s) to standard output.

`-E`, display \$ at the end of each line

`-n`, number all output lines

ERRORS HANDLED

1. In case of an invalid file or a file that does not exist, gives the error and does not prints any files.
2. In case of a single file given, print the contents of the single file only.
3. If a wrong flag is used, it prints the error

• **date**

NAME

date - print or set the system date and time

SYNOPSIS

```
date [OPTION]... [+FORMAT]
date &t [OPTION]... [+FORMAT]
```

DESCRIPTION

Display the current time.

-I prints the only the date in YYYY-MM-DD format
-u prints UTC the date and time.

FORMAT controls the output. Interpreted sequences are:

%a	locale's abbreviated weekday name (e.g., Sun)
%b	locale's abbreviated month name (e.g., Jan)
%d	day of month (e.g., 01)
%m	month (01..12)
%Y	Year
%H	Hours(00 .. 23)
%M	Minutes(00 .. 59)

%S Seconds(00 .. 60)

ERRORS HANDLED

1. If a wrong flag is used, it prints the error and returns without creating a directory.
2. In case of wrong input format, print the error without returning the date.

- **rm**

NAME

rm - remove files or directories

SYNOPSIS

rm [OPTION]... [FILE]...
rm &t [OPTION]... [FILE]...

DESCRIPTION

rm removes each specified file. It does not remove directories. Can remove empty directories.

-i, prompt before every removal

-v, explain what is being done

ERRORS HANDLED

1. If a wrong flag is used, it prints the error and returns without creating a directory.
2. In case of an invalid file name or directory, do not delete any file and print the error.

Possible test cases

```
echo -n "Hello  
World"
```

```
Pwd -P  
cd ..
```

```
ls -a  
mkdir -p a  
rm -v -i a  
cat -n a.txt (a.txt must be created beforehand)  
date -u
```

```
Pwd -L  
ls &t -l  
mkdir &t -v a  
rm &t -i -i a  
cat &t -E a.txt b.txt (a.txt, b.txt must be created beforehand)  
date &t -I
```