**1.1 & 1.2 Documentation**

**Basic Linux/Unix shell**

**` By: PARAS DHIMAN**

**System:-**

* The code has been written in two files containing fork() and exec() and other one containing pthread() and system() and shell has been made which is taking 2options input for each of the commands this shell is suitable for x86\_64 operating systems.

**Internal**

* **Cd(change directory)**

cd: cd [-L|[-P [-e]] [-@]] [dir]

Change the shell working directory.

Change the current directory to DIR. The default DIR is the value of the

HOME shell variable.

If the directory is not found, and the shell option `cdable\_vars' is set,

the word is assumed to be a variable name. If that variable has a value,

its value is used for DIR.

Options:

-L force symbolic links to be followed: resolve symbolic

links in DIR after processing instances of `..'

-P use the physical directory structure without following

symbolic links: resolve symbolic links in DIR before

processing instances of `..'

-e if the -P option is supplied, and the current working

directory cannot be determined successfully, exit with

a non-zero status

--help display this help and exit

The default is to follow symbolic links, as if `-L' were specified.

`..' is processed by removing the immediately previous pathname component

back to a slash or the beginning of DIR.

Exit Status:

Returns 0 if the directory is changed, and if $PWD is set successfully when

-P is used; non-zero otherwise.

ASSUMPTIONS:-

The input should be provided in ways:-

First the command has to be given :- (command)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command

-P used to change the physical directory of the cell

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of -P and –help no arguments are taken so then it shows the error of the arguments. Checks if the directory is there or not if not present then it throws the error.

/ character has been taken care of if directory not present then shows error

* **Echo**

HO(1) User Commands ECHO(1)

NAME top

echo - display a line of text

SYNOPSIS top

echo [SHORT-OPTION]... [STRING]...

echo LONG-OPTION

DESCRIPTION top

Echo the STRING(s) to standard output.

-n do not output the trailing newline

-E disable interpretation of backslash escapes (default)

--help display this help and exit

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- echo (options)

This command is same as print() command in java, C ,C++ ,python this command basically print the arguments given after the command and option

Only 2 options are given

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of -P and –help no arguments are taken so then it shows the error of the arguments.

Lines having \n as the end character has been counted.

* **History**

HISTORY(3) Library Functions Manual HISTORY(3)

NAME top

history - GNU History Library

DESCRIPTION top

-c empty the history file

Many programs read input from the user a line at a time. The GNU

History library is able to keep track of those lines, associate

arbitrary data with each line, and utilize information from

previous lines in composing new ones.

ASSUMPTIONS:-

The input should be provided in ways:-

First the command has to be given :- history

Won’t take any input but it would just show the history of all the commands entered by the user unless he doesn’t explicitly or through the -c option remove the content of the file.

ERRORS:-

It will take options specified above if other than that are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments if given less. In command of -c no arguments are taken so then it shows the error of the argument.

* **Pwd**

er Commands PWD(1)

NAME top

pwd - print name of current/working directory

SYNOPSIS top

pwd [OPTION]...

DESCRIPTION top

Print the full filename of the current working directory.

-L, --logical

use PWD from environment, even if it contains symlinks

-P, --physical

avoid all symlinks

--help display this help and exit

ASSUMPTIONS:-

The input should be provided in ways:-

First the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of -P and –help no arguments are taken so then it shows the error of the arguments.

If no options given in the options then it is by default taken to be -P.

**External:**

* **Ls**

(1) User Commands LS(1)

NAME top

ls - list directory contents

SYNOPSIS top

ls [OPTION]... [FILE]...

DESCRIPTION top

List information about the FILEs (the current directory by

default). Sort entries alphabetically if none of - nor

--sort is specified.

Mandatory arguments to long options are mandatory for short

options too.

-a, --all

do not ignore entries starting with .

-d, --directory

list directories themselves, not their contents

--help display this help and exit

Exit status:

0 if OK,

1 if minor problems (e.g., cannot access subdirectory),

2 if serious trouble (e.g., cannot access command-line

argument).

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

This assumes that the argument that is passed by the user is default given as string also doesn’t check for /.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of –help no arguments are taken so then it shows the error of the arguments.

* **Cat**

NAME top

cat - concatenate files and print on the standard output

SYNOPSIS top

cat [OPTION]... [FILE]...

DESCRIPTION top

Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

-E, --show-ends

display $ at end of each line

-n, --number

number all output lines

--help display this help and exit

EXAMPLES top

cat f - g

Output f's contents, then standard input, then g's

contents.

cat Copy standard input to standard output.

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of –help no arguments are taken so then it shows the error of the arguments.

If no file exists as the given file then it shows an error.

Checked the file to see if it is present over there or not

Checked if the file is given or not

* **Date**

DATE(1) User Commands DATE(1)

NAME top

date - print or set the system date and time

SYNOPSIS top

date [OPTION]

DESCRIPTION top

-u

print or set Coordinated Universal Time (UTC)

-R

output date and time in RFC 5322 format. Example: Mon, 14

Aug 2006 02:34:56 -0600

--help display this help and exit

Display the current time in the given FORMAT, or set the system

date.

Mandatory arguments to long options are mandatory for short

options too.

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

If no input is given then by default it will give the default local time.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of –help no arguments are taken so then it shows the error of the arguments.

If no input is given then by default it will give the default local time.

* **Rm**

M(1) User Commands RM(1)

NAME top

rm - remove files or directories

SYNOPSIS top

rm [OPTION]... [FILE]...

DESCRIPTION top

This manual page documents the GNU version of rm. rm removes

each specified file. By default, it does not remove directories.

If the -I or --interactive=once option is given, and there are

more than three files or the -r, -R, or --recursive are given,

then rm prompts the user for whether to proceed with the entire

operation. If the response is not affirmative, the entire

command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal,

and the -f or --force option is not given, or the -i or

--interactive=always option is given, rm prompts the user for

whether to remove the file. If the response is not affirmative,

the file is skipped.

OPTIONS top

Remove (unlink) the FILE(s).

-f, --force

ignore nonexistent files and arguments, never prompt

-i prompt before every removal

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of –help no arguments are taken so then it shows the error of the arguments.

* **Mkdir**

R(1) User Commands MKDIR(1)

NAME top

mkdir - make directories

SYNOPSIS top

mkdir [OPTION]... DIRECTORY...

DESCRIPTION top

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

Mandatory arguments to long options are mandatory for short

options too.

-p, --parents

no error if existing, make parent directories as needed

-v, --verbose

print a message for each created directory

--help display this help and exit

ASSUMPTIONS:-

The input should be provided in ways:-

First, the command has to be given :- (command) (options)

Second then need to give the arguments:- -(option) (file name or any other argument that is need to be passed)

Only 2 options are given for each command if there are no options given in the options then it is by default taken to be -P.

ERRORS:-

If options other than the 2 options specified above are provided then the error will be shown over there and if more arguments are given than that to be provided error will be shown as too many arguments or too few arguments. In command of –help no arguments are taken so then it shows the error of the arguments.

Shown error if the directory given is not found

Shown error if directory is already creted over there and unable to create or overwrite it again

**THE MAJOR ERRORS HANDLED BY ME ARE:-**

1. If someone enter the Wrong command
2. Wrong Command Line Options
3. If the directory Doesn,t exist
4. If too few arguments given
5. If more arguments are given than needed
6. Checked mode of creation of directory
7. History is saved if wrong commands ar given

**Test Cases Checked:-**

* ls
* ls

-l

* ls

-a

* ls

-a path

* cat

file1

* cat

file1 file2

* cat

-E file1

* cat

-T file2

* date
* date

-R

* date

-u

* rm

file1

* rm

file1 file2

* rm

-d directory1

* rm

-v file1

* mkdir

a

* mkdir

c b

* mkdir

-v d

* mkdir

-m

* cd
* cd path
* cd -P path
* cd -L path
* cd –help
* echo text
* echo -n text
* echo -E text
* echo –help
* history
* history -a 10
* history
* history -c
* history
* history –help
* pwd
* pwd -P
* pwd -L
* exit