Kartik Gupta (2021056)

Assignment:
Implementation of a Linux shell in C, a simple shell that can handle three, internal commands – ‘cd’, ‘echo’ and ‘pwd’. These commands would be handled directly by the shell. Shell also handles five external commands – ‘ls’, ‘cat’, ‘date’, ‘rm’ and ‘mkdir’.

Commands Implemented

1.NAME
cat – concatenate and print files

SYNOPSIS
cat [-bn] [file1] ..[file3]

DESCRIPTION.
The cat command is used to display the contents of the selected files, line by line. This has been implemented as an external command, where the external c file is called as a child process using the execvp command. Flags – • -n – Numbers each line of the file being displayed. • -e – Demarcates the difference between two lines using a $ sign. Case Handling –.

The options implemented are:

-e : Demarcates the difference between two lines using a $ sign.
-n : displays lines with index

ERROR HANDLING
1. When invoked without arguments, displayed arguments expected.
2. if invalid file passed, returns error.

2.NAME
cd — change the working directory

SYNOPSIS
rm [-h|~|directory]

DESCRIPTION
The cd command is used to change directories. It follows the cd [options] [args] format, where it changes the shell’s current directory to the one given in args, if it is accessible. cd has been implemented using a combination of getcwd and chdir commands, where the strings are operated upon.

The options implemented are:

-P Instructs the shell to use the physical directory structure instead of following symbolic links.
~ change directory to home directory
.. change directory to parent directory
–help prints information about command usage

ERROR HANDLING
1. The chdir commands return values have been utilized to return the user with error if given folder doesn't exist.
2. Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.

3.NAME
date – display date and time

SYNOPSIS
date [-r filename] [-u]

DESCRIPTION
The date command is used to display the current date and time in accordance with the system's set date. It follows the date [options] format. This has been implemented as an external command, where the external c file is called as a child process using the execvp command.

The options implemented are:
-u returns time in UTC/GMT
-R returns time in RFC 2822 date and time format.

ERROR HANDLING
1. Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.
2. The exact specific syntax of the UNIX system has been imitated by using string operations.

4.NAME
echo — print to the terminal

SYNOPSIS
echo [-hn] [string]

DESCRIPTION
The echo command takes in arguments in the echo [options] [args] format and then prints out the args in the terminal.

The options implemented are:

–help displays correct usage
-n Do not print the trailing newline character.

ERROR HANDLING
1. Cases where non-flag multiple words were used have been handled and are treated as normal text.
2. When invoked without arguments, displayed arguments expected.

5.NAME
ls – list directory content

SYNOPSIS
ls [-a1]

DESCRIPTION.
ls command is used to display the content of the directory which we currently are in at a given time in the shell. It follows the ls [options] format.

The options implemented are:

-i/–inode display files with their file numbers
-Q displays files in double quotes

ERROR HANDLING
1.The opendir command's return value is used to check whether a directory can sucessfully be opened or not (NULL check).
2. Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.

6.NAME
mkdir - make directory

SYNOPSIS
mkdir [-pv] [filename]

DESCRIPTION.
The mkdir command is used to create a directory with a given name in the directory we presently are in. The format for mkdir is mkdir [options] [args], where args are the names for the new directories that need to be formed. Files are created with an rwxrwxrwx(0777) mode.

The options implemented are:

-p Create intermediate directories as required.
-v verbose; show file name and then create

ERROR HANDLING

1. The return value of mkdir is an integer 0 if the directory creation was successful. This is used to handle the directory creation related errors.
2. Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.

7.NAME

pwd — display the working directory

SYNOPSIS

pwd

DESCRIPTION

The pwd command is used to print out the current directory in which the user is in. It follows the pwd [options] format. This has been implemented using the getcwd function syscall which returns the current location to the allotted char array variable.

The options implemented are:

-L For showing the symbolic positioning of current directory.
-P For showing the physical positioning of current directory.

ERROR HANDLING

1. Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.
2. The case of pwd and -P behaving similarly and -L behaving differently has been taken into account by using different commands.

8.NAME

rm – remove directory entry

SYNOPSIS

rm [-vi] [file1] [file2]...

DESCRIPTION

The rm command is used to remove a particular file from the current directory we are in. It follows the rm [options] [arg] format.

The options implemented are:

-v verbose; show file name and then delete
-i ask for confirmation

ERROR HANDLING

1. When invoked without arguments, displays error of arguments not passed.
2. status check of remove() function returns error from errno

TEST CASE:

Shell-Simulator: make

****

Shell

****

Shell-Simulator: /home/runner/Shell$ date -R

Mon, 31 Oct 2022 17:08:30 +0530

Shell-Simulator: /home/runner$ echo -n hello world!

hello world! Shell-Simulator: /home/runner$ cd Shell

Shell-Simulator: /home/runner/Shell$ ls -i

256 . 256 .. 257 .cache 261 replit.nix 262 .breakpoints 263 .ccls-cache 316 .replit 319 Makefile 528 Makefile.txt 558 makefile 563 cat.c 564 mkdir.c 565 main.c 566 rm.c 567 date.c 568 ls.c 576 ls 577 cat 578 date 579 mkdir 580 rm 581 a.out

Shell-Simulator: /home/runner/Shell$ mkdir newlyCREATEDfile

Shell-Simulator: /home/runner/Shell$ ls -i

256 . 256 .. 257 .cache 261 replit.nix 262 .breakpoints 263 .ccls-cache 316 .replit 319 Makefile 528 Makefile.txt 558 makefile 563 cat.c 564 mkdir.c 566 rm.c 567 date.c 568 ls.c 576 ls 577 cat 578 date 579 mkdir 580 rm 581 a.out 583 main.c 796 newlyCREATEDfile

Shell-Simulator: /home/runner/Shell$ rm -i newlyCREATEDfile

Are you sure you want to remove newlyCREATEDfile? (Yes->Y/y, NO->N/n) : y

Shell-Simulator: /home/runner/Shell$ ls -i

256 . 256 .. 257 .cache 261 replit.nix 262 .breakpoints 263 .ccls-cache 316 .replit 319 Makefile 528 Makefile.txt 558 makefile 563 cat.c 564 mkdir.c 566 rm.c 567 date.c 568 ls.c 576 ls 577 cat 578 date 579 mkdir 580 rm 581 a.out 583 main.c

Shell-Simulator: /home/runner/Shell$ rm hh.txt

No such file found.

Shell-Simulator: /home/runner/Shell$ cat -n makefile

2 run:
3 gcc ls.c -o ls
4 gcc cat.c -o cat
5 gcc date.c -o date
6 gcc mkdir.c -o mkdir
7 gcc rm.c -o rm

8 gcc main.c -lpthread
9 clear
10 ./a.out
11
12 clear:
13 rm ls
14 rm a.out
15 rm cat
16 rm date
17 rm mkdir
18 rm rm
Shell-Simulator: /home/runner/Shell$ pwd -L
/home/runner/Shell
Shell-Simulator: /home/runner/Shell$ date -R
Mon, 31 Oct 2022 17:48:38 +0530