CSE231: Operating Systems
Assignment 1

Name: Agamdeep Singh
Roll No: 2021306

_____

The shell is a basic C program which uses various libraries to take inputs from the user and parses them into tokens and then either call functions from within the shell program or use fork(), execl(), wait() or pthread_create(),system() to execute them and product required results

The various options a user can run in the shell are:

S.No	Command		Option 1	Option 2
1	echo		- n	- - help
2	pwd		- P	- L
3	cd		~	-
4	ls		- a	- A
5	cat		- n	- E
6	rm		- f	- v
7	date		- u	- r
8	mkdir		- v	- m

1. **echo:**

   echo <option> <string>
   It can be used to display a line of text

   Options:
   - n   allows user to remove trailing EOL character in output
   - - help   allows user to view the help

2. **pwd:**

   pwd <option>
   It can be used print current/working directory

   Options:
   - P   avoids all syslinks while printing current/working directory
   - L   uses PWD from environment, even if it contains symlinks

3. **cd:**
   cd <option> <directory>
   Can be used to change directory to the given directory address
   Options:
   -               allows user to visit previous directory
   ~   allows user to go to /home/<user> directory

4. **ls**
   ls <option>
   Can be used to list all contents of a directory
   Options:
   - a   allow user to view all directories including the ones starting with".""
   - A   displays all directories including the ones starting with "." except "." and ".."

5. **cat**
   cat <option> <filename>
   Allows user to print a file in the directory the shell is invoked in
   Options:
   - n   prints the file along with the line numbers in the file
   - E   prints the file with a '$' at the end of each line

6. **rm**
   rm <option> <filename>
   Allows user to delete a file the directory the shell is invoked in
   Options:
   - f   doesn't give an error if the file that is being deleted doesn't exist or is a directory
   - v   gives a verbose description of what is happening

7. **date**
   date <option>
   Allows user to view the current date and time
   Options:
   -R   output date and time in RFC 5322 format. Example: Mon, 14 Aug 2006 02:34:56 -0600
   -u   print Coordinated Universal Time (UTC)

8. **mkdir**
   mkdir <option 1> <optional token> <directory_name>
   Allows user to make a directory in the directory the shell is invoked in
   Options:
   - v   gives a verbose description of what is happening
   - m   set file mode a=rwx/rw

Assumptions:

1. The user will input the command in the correct format.
2. The user will input the correct number of tokens while inputting the commands.
3. The user will only use one command option at a time.
4. The user will only try to use the commands in the directory the shell is invoked in.
5. echo: A $ symbol in this shell won't necessarily function like it does in bash.
6. cd: The "cd -" command won't be used successively. A cd will occur in between each use of "cd -".
7. mkdir: mkdir command will only make 1 directory at a time.
8. date: The system has a time and an active internet connection
9. pwd: the user hasn't made a syslink(it should work but this shell doesn't have ln functionality)
10. cat: the file in the cat command isn't an executable or doesn't have any EOF character before the actual ending of the file.

Test Cases:

All test cases to begin with:
> make   should instantly run the shell
> make run(optional)   in case the shell doesn't run upon the
                                          command

    echo 1
    echo 12 3 123
    echo -n 1
    echo -n 12 12 3 123
    echo - -help

    pwd
    mkdir 1
    mkdir -v 2
    mkdir -m a=rwx 3
    (use terminal to "ls -l to check permissions")
    cd 1
    pwd
    cd -
    pwd
    cd ~
    pwd
    cd <directory of invocation>
    pwd
    pwd -L
    pwd -P (requires ln)

(use terminal to touch 1.txt & 2.txt)
rm 1.txt
ls
rm -v 2.txt
ls
rm 1
rm -f 1
ls -a
ls -A

cat test.txt
cat -n test.txt
cat -E test.txt

date
date -u
date -R
date&t
date&t -u
date&t -R

exit

Edge Cases Handled:

General:
- If the first token of a command is incorrect, it gives an error.
- If a normal enter is struck, prompts a new input from the user similar to bash.

Specific:
1. echo
   a. When given -n as input token it doesn't print the leading EOL character
   b. When given - -help as a token it opens the help information
2. ls
   a. Doesn't show hidden files without -a or -A symbol.
   b. Doesn't show the current and parent directory in the -A menu
   c. Doesn't crash when given incorrect number of tokens
3. mkdir
   a. Doesn't let you create a file if a file with that name already exists
   b. Doesn't give errors when creating a directory with a name that is already assigned to another file when using the -f token
   c. Doesn't crash when given incorrect number of tokens
4. rm
   a. doesn't let you delete directories with any flag(since -r is not included in my shell)
   b. gives an error when trying to delete a file that doesn't exist
   c. Doesn't crash when given incorrect number of tokens
5. pwd
   a. Gives physical address when -P is used
   b. Gives Logical Address when -L is used
6. Date
   a. displays the system's when no flag is used
   b. Displays the UTC when -u flag is used
   c. Display the RFC 5322 format when -R flag is used
   d. Doesn't crash when given incorrect number of tokens
7. cat
   a. Display the file with $ signs when -E is used
   b. Displays the file along with line numbers when -n is used
   c. Doesn't crash when given incorrect number of tokens
   d. Displays the entire file when no flag is used
8. cd
   a. Gives error when a non-existent directory is passed
   b. Goes to /home/<user> when ~ is used
   c. Goes to the previous directory when - is used
   d. Doesn't crash when given incorrect number of tokens