Operating Systems

Basic Linux Shell

The source code for the shell program is named ‘myshell.c’.
The source code and their binaries for the external commands are as follows:
ls: myls.c , binary: myls
cat: mycat.c , binary: mycat
date: mydate.c , binary: mydate
mkdir: mymkdir.c , binary: mymkdir
rm: myrm.c , binary: myrm

The Makefile targets are as follows:

- compile: to compile all the source codes .c files
- run: to execute the binary of myshell.c, i.e, to run the shell.
- clean: to remove the binary file of the shell.

Assumptions:

The name of the shell is ‘Shruti Shell’.
Assumptions taken into account in this shell are as follows:

- Maximum length of input string taken is 1000 characters.
- Maximum command length supported (command name+options+arguments) = 100.
- Directory names of maximum 1024 characters length are supported.
- If some error occurs in any system call, or an invalid command/option/arguments are passed, the appropriate error messages are thrown. Only when the user types ‘exit’ command will the shell stop execution.
- GNU readline libraries are used in the shell source code, so the machine on which this shell is run must have the readline-dev package installed. To install this package, type this on your linux terminal: sudo apt-get install libreadline-dev

About the shell:

- 5 internal commands in Shruti Shell: cd, echo, history, pwd and exit.
- 5 external commands in Shruti Shell: ls, cat, date, mkdir and rm.
- After printing all the errors for the entered command, a new shell line is printed automatically to take the next user input.
- fork(), exec() and wait() system calls are used to handle the external commands of the shell.
- Upward arrow key on the keyboard can be used to get the previous entered commands in that particular session.
- File names starting with ‘-’ are allowed by this shell.
- If a command which is neither an internal or external command is entered, then the error message saying ‘Shruti Shell: No such command exists’ is printed.

- There is a txt file named historyfile.txt in the same directory as the myshell.c file to store the history of this shell for all sessions.

Internal commands supported:

1. cd
    - To change the current working directory of the shell. It forces symlinks to be followed by default.
    - Syntax: cd [-L/--help] [dir]
    - Options: -L, --help
        - -L: This options works like the default implementation of the cd command, i.e., cd dir and cd -L dir, works the same. It forces symbolic links to be followed.
        - --help: This option displays the details about the cd command on stdout.
    - Errors handled:
        - cd can have only 1 argument at a time, thus if more than one directory names are given (with/without the option), an error saying ‘too many arguments’ is printed. Eg: cd -L sem3 sem4 or cd sem3 sem4 will give this error.
        - If the directory name entered is not valid, then the chdir() system call returns a negative value, in which case the error message saying ‘No such file or directory’ is printed.
    - Special cases handled:
        - If the user does not provide a directory name (with/without -L), i.e. just types: cd or cd -L, or types cd ~, then the directory is changed to the home directory.
        - cd / changes current directory to the root directory, while cd .. changes directory to the parent directory of current directory.
    - Any other argument starting with ‘-’ is considered a directory name argument.
2. echo
    - Display a line of text entered by the user back on the stdout.
    - Syntax: echo [-n/-E] [line to print]
    - Options: -n, -E
        - -n: does not print the trailing newline.
        - -E: disable the interpretation of backslash escapes. This option works like the default implementation of the echo command, i.e., echo hello and echo -E hello both print hello on stdout.
    - By default, echo disables the interpretation of backslash escapes. Eg: echo [-E] shruti\njha prints shrutinjha.
    - Errors/Corner cases handled:
        - If no argument is provided, print an empty line.
        - If an argument is entered with padded spaces in the front, output is given after removing the spaces. Eg: >>>echo   hello prints >>>hello

    - If input is entered in quotes, the input is printed as it is. Eg: echo “shruti\njha” prints shruti\njha.
    - Multiple arguments at a time are supported. Arguments starting with ‘-’ are allowed.

3. history
    - Display the history of ALL the sessions of this shell on stdout, along with line numbers.
    - Syntax: history [-c/--help] [n]
    - Options: -c, --help
        - -c: clear the entire history
        - --help: display details of history command on stdout.
    - An argument n (integer) can be passed to see only the last n history entries.
    - Errors handled:
        - If the history file, where all the commands are appended, cannot be opened using the fopen() function, then the error message saying ‘cannot open history file’ is printed and the shell line is displayed to take the next input.
        - If an argument is entered which is not the options -c or --help, and it is not an integer as well, then an error message saying ‘invalid option’ is printed. Eg: history hello prints this error.
        - If more than 1 integer argument is passed to history or any argument is passed along with -c option, then an error message saying ‘invalid option’ is printed. Eg: history -c 5 or history 5 10 will throw this error.
    - Special cases handled:
        - If the argument n is greater than the number of history entries, then the entire history list is printed.
4. pwd
    - Prints full path name of the current working directory. It avoids symlinks by default.
    - Syntax: pwd [-P/--help]
    - Options: -P, --help
    - -P: This option works like the default implementation of pwd command, i.e. pwd and pwd -P works the same.
    - --help: this prints details about the pwd command on stdout.
    - Errors handled:
        - If an argument is passed along with the pwd command, an error message saying ‘invalid option’ is printed.
        - If an argument is passed along with the -P option, then an error message saying ‘invalid option’ is printed.
        - If there is an error getting the cwd because of the getcwd() system call returns NULL pointer, then an perror() function prints the error returned by getcwd.
5. exit
    - To exit the Shruti Shell.

- Exits with status 0 by default.
- Syntax: exit [--help] [n]
- Options: --help
    - --help: prints details about the exit command on stdout.
- Special cases handled:
    - If the argument n passed is an integer, then exit is made using n status. Else if the argument is non-integer or no argument is passed, then exit is made using 0 status.
    - If more than one argument is passed, exit made with the first argument.

External command supported:

1. ls
    - To print the directory contents (current directory by default).
    - By default, entries starting with "." are ignored and entries printed in alphabetical order.
    - Syntax: ls [-a/-r] [file 1] [file 2] …...
    - Options: -a, -r
        - -a: do not ignore entries starting with "."
        - -r: list the entries in reverse alphabetical order.
    - Errors handled:
        - Scandir() system call is used to scan and store the contents of the entered directory. In case of error, for eg. the directory does not exists, it returns negative value, in which case the perror() is used to print the error returned by scandir.
        - In case an invalid option is entered, an error message saying 'invalid option' is thrown.
    - Special/Corner cases handled:
        - If the user does not provide a directory name (with/without option), i.e., just types ls or ls -a/-r, then the contents of the current directory is printed.
        - Multiple arguments at a time are supported.
        - If an argument starts with '-' and it is not one of the two options, then it is considered as a directory name.
        - Both flags/options can be used together in any order.
2. cat
    - Print file contents on stdout
    - Syntax: cat [-n/-E] [file1] [file2] ….
    - Options: -n, -E
    - -n: number all the output lines
    - -E: display $ at the end of each line
    - Errors handled:
        - fopen() function is used to open the entered file in read mode. If the pointer returned by fopen is NULL, then an error message saying 'unable to open the file' is printed.

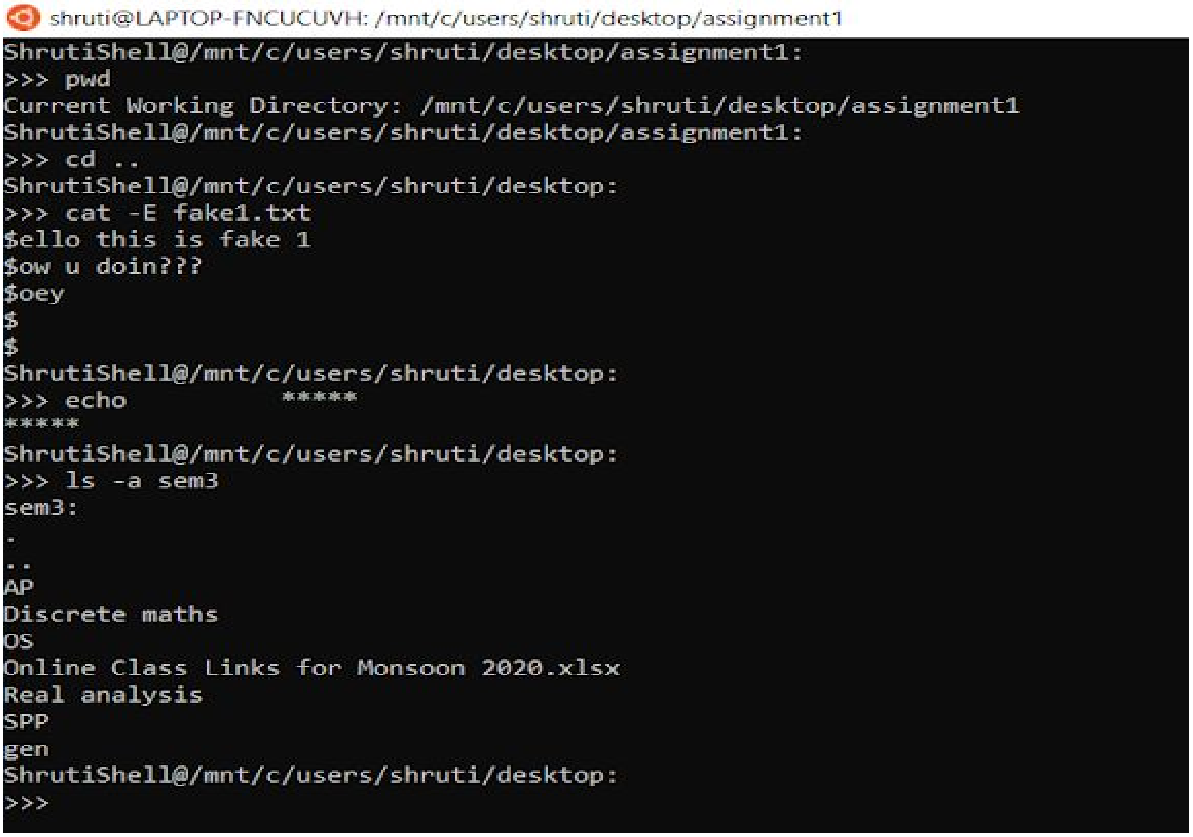
        - If fgets() returns NULL, the file reading is stopped and a new shell line is printed to get the next command input.
        - If no filename is provided as an argument to cat (with/without the options), then an error message saying ‘missing operands’ is printed.
    - Special cases handled:
        - Multiple arguments at a time are supported.
        - Both flags can be used together in any order. If this is done, first the dollar sign, then the line number and then the line read is printed in the order.
        - File names starting with ‘-’ are allowed.
3. date
    - Prints the system time and date.
    - Syntax: date [-u/-I]
    - Options: -u, -I
    - -u: print the Greenwich Mean Time (GMT), also called UTC
    - -I: print the date in ISO 8601 format (only date format, i.e. YYYY-MM-DD implemented, which is the default)
    - Errors handled:
        - time() function is used to return the time_t type variable. If it returns -1, it means the function didn’t work, and an error message saying ‘time() failed’ is printed.
        - localtime() function is used to get the system local date and time (tm struct), and if it returns NULL, it means the function didn't work, and an error message saying ‘localtime() failed’ is printed.
        - Similarly, gmtime() function is used to get the GMT time, and if it also returns NULL, then an error message saying ‘gmtime() failed’ is printed.
        - If any argument other than the options -u or -I is given, an error message ‘invalid option’ is thrown.
    - Special cases handled:
        - Both flags can be used together in any order.
4. mkdir
    - It is used to make directories. Cannot make a directory with the same name as an existing directory.
    - Syntax: mkdir [-p/-v] [dir 1] [dir 2] …..
    - Options: -p, -v
        - -p: no error if existing, make parent directories as needed. It means that it can make directories recursively, and if the directory already exists, then no error is thrown. Eg. mkdir -p sem3/sem4/sem5 would make a directory sem3 containing the directory sem4 which in turn contains the directory sem5. Without using this option, this cannot be done, i.e. mkdir sem3/sem4/sem5 will give an error.
        - -v: print a message for each created directory.
    - Errors handled:

    - mkdir() function is used to make a directory. The return value of this function is checked, and if it is not 0, then an error occurred, and this error is printed using the perror() function.
    - If no arguments are provided with mkdir (with/without the options), then an error message saying ‘missing operands’ is printed. Eg. mkdir or mkdir -v/-p would give this error.
  - Special cases handled:
    - Multiple arguments at once are supported.
    - Both flags can be used together in any order.
    - File names starting with ‘-’ are allowed.

5. rm
   - It is used to remove files and empty directories.
   - Syntax: rm [-i/-v] [file 1] [file 2] …..
   - Options: -i, -v
     - -i: prompt the user before every removal. If the user enters y, only then delete the file, otherwise not.
     - -v: print a message for each removed file.
   - Errors handled:
     - remove() function is used to remove a file or empty directory. The return value of this function is checked, and if it is not 0, then an error occurred, for eg. the file does not exist or the directory is not empty, and this error returned by remove() is printed using the perror() function.
     - If no arguments are provided with rm (with/without the options), then an error message saying ‘missing operands’ is printed. Eg. rm or rm -v/-i would give this error.
   - Special cases handled:
     - Multiple arguments at once are supported.
     - Both flags can be used together in any order.
     - File names starting with ‘-’ are allowed.

Test case to check the shell:
https://docs.google.com/document/d/1f_Fbjeb6TOC1mu927YbQzSaNmZiYp4UWyYAFIuCaG7M/edit?usp=sharing

The output of Shruti Shell looks like this:


shruti@LAPTOP-FNCUCUVH: /mnt/c/users/shruti/desktop/assignment1

```
ShrutiShell@/mnt/c/users/shruti/desktop/assignment1:
>>> pwd
Current Working Directory: /mnt/c/users/shruti/desktop/assignment1
ShrutiShell@/mnt/c/users/shruti/desktop/assignment1:
>>> cd ..
ShrutiShell@/mnt/c/users/shruti/desktop:
>>> cat -E fake1.txt
$ello this is fake 1
$ow u doin???
$oey
$
$
ShrutiShell@/mnt/c/users/shruti/desktop:
>>> echo          *****
*****
ShrutiShell@/mnt/c/users/shruti/desktop:
>>> ls -a sem3
sem3:
.
..
AP
Discrete maths
OS
Online Class Links for Monsoon 2020.xlsx
Real analysis
SPP
gen
ShrutiShell@/mnt/c/users/shruti/desktop:
>>>
```