

REPORT - Garv Makkar

Running the terminal:

In terminal first write 'make' to compile all the codes and then './myshell' to run the program. Now my scripted shell is ready to use.

NOTE: In this report, '...' is used to represent more arguments of similar type as the one written before '...'

NOTE: If you download and run these files on your machine, make sure to change the path of files used in code. (To TAs : For submission I will be renaming the folder so allow me to rename it again during demo so that i can run these...

MYSHELL :

Functionalities:

1. Internal commands like : cd, echo, pwd, exit.
2. External commands like: cat, mkdir, rm, ls, date.
External commands can be called in 2 ways:
 - a) Direct way
e.g 'mkdir fol1 fol2' (this gets executed by fork and exec).
 - b) By adding &t
e.g 'mkdir fol1 fol2 &t' (this gets executed via pthread and system call).
3. You can go to next line by pressing enter without typing any command.

Assumptions:

1. Any directory name for any command shall not have spaces.
E.g 'cd 'new folder' is not valid and users are requested to name directories without using spaces in between.

Others:

1. My prompt shows the current directory following with ':~ \$'.
2. The home directory is: '/home/garv/Desktop/Cprograms/GarvShell'.

INTERNAL COMMANDS:

CD:

Assumptions:

1. As stated, directory names must not have spaces.
2. Home directory used is “/home/garv/Desktop/Cprograms/GarvShell”.

Functionalities:

1. cd \$directory (to move to \$directory)
2. cd .. (to move to previous directory)
3. cd ~ (to go to home directory)
4. cd (to go to home directory)

Errors handled:

1. Accessing directories that are not present.
2. More or less arguments.
3. Any other format of this command.

EXIT :

Functionalities:

1. exit (the scripted shell will stop and program will end)
2. Anything written after word ‘exit’ in command will make no changes, and program will still end

ECHO:

Assumptions:

1. Special characters other than ‘\$’ is not considered, anything after echo will be printed back as it is.

Functionalities:

1. echo (to print and empty line)
2. echo xyz ... (to print xyz ...)
3. echo xyz \$abc def ... (to print xyz def ...)
4. echo -n ... (to print ... and call the prompt in the same line instead of a new line by default)

Errors handled:

Basically none because whatever you enter gets printed back.

PWD:

Assumptions:

1. Symbolic links are not being considered
2. Even if we write anything after pwd, it does not affect the command, just like what happens in actual terminal.

Functionalities:

1. pwd (to print the current working directory)
 2. pwd -L (same as pwd)
 3. pwd -P (same as pwd)
- (2 and 3 result in same because symbolic links are not being considered)

Errors handled:

Basically none because whatever you enter gets you the current directory just like The actual linux terminal.

EXTERNAL COMMANDS:

CAT:

Assumptions:

1. If the file accessed is a directory, its not an error, it just does not print anything
2. No spacing between file names too.

Functionalities:

1. cat (now whatever you enter, gets printed)
2. cat \$filename ... (content(s) of \$filename ... gets printed)
3. cat -n \$filename ... (content(s) of \$filename ... gets printed with line number before line)
4. cat -E \$filename ... (content(s) of \$filename ... gets printed with dollar sign after line)

Errors handled:

1. Trying to read a file that doesnt exist or shell is not able to access the file
2. Command in any other format

DATE:

Assumptions:

1. Using date command will give time according to IST.

Functionalities:

1. date (gives current date and time according to IST)
2. date -u (gives current date and time according to GMT/UTC)
3. date -A (gives the current weekday's name)

Errors:

1. More than required arguments
2. Any other command format

LS:

Assumptions:

1. Hidden files are not printed in any format
2. Only current directory names are printed, not what is inside them.
3. The files are printed with spaces in between unlike tabular format.

Functionalities:

1. ls -m (to print files separated by comma)
2. ls -1 (to print files in new line)

Errors:

1. More arguments
2. Any other command format

MKDIR:

Assumptions:

1. Only empty new directories are made.
2. No spaces in names of directories.

Functionalities:

1. `mkdir $dir1 ...` (making directories with names `$dir1 ...` respectively (multiple folders can be created))
2. `mkdir -v` (same as above but it tells us if the directory is made or not)

Errors:

1. If user tries to make a directory, which has the same name of already existing file or directory.
2. Less argument(s) are passed

RM:

Assumptions:

1. No spaces in names.

Functionalities:

1. `Rm $file ...` (deletes the file `$file`)
2. `Rm -d $dir ...` (deletes the directory `$dir` if `$dir` is empty)
3. `Rm -i $file ...` (asks user for conformation to delete file `$file`)

Errors:

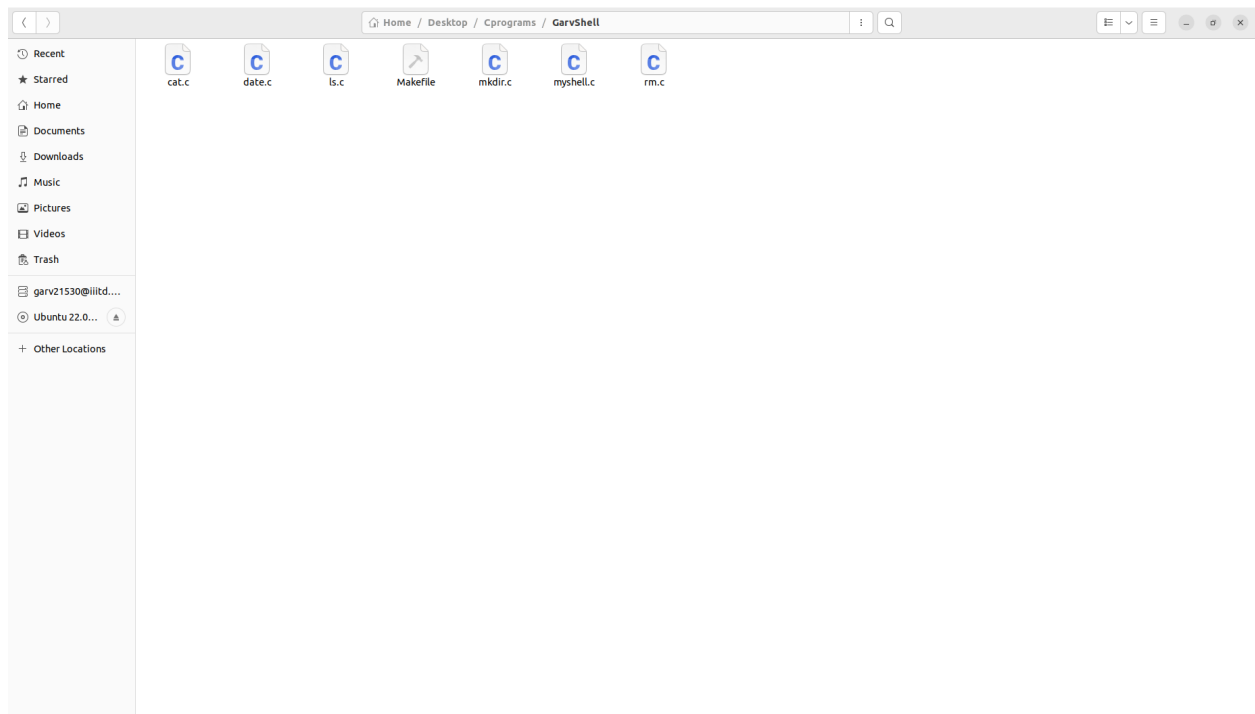
1. Deleting directory in a command that deletes file and vice a versa.
2. Deleting stuff that does not exist.

TESTCASES:

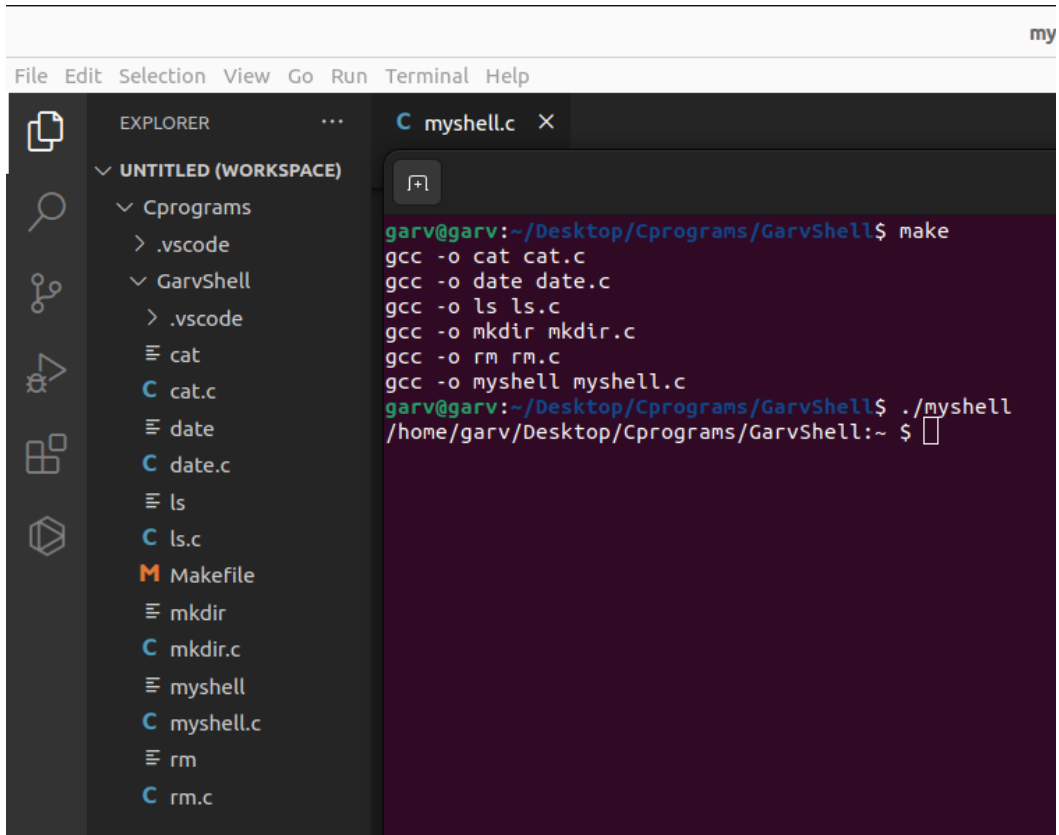
Internal commands are covered, external commands with fork/exec are also covered.

External commands ending with &t are not covered, they will execute the same way in terminal.

The folder with all programs before compilation:



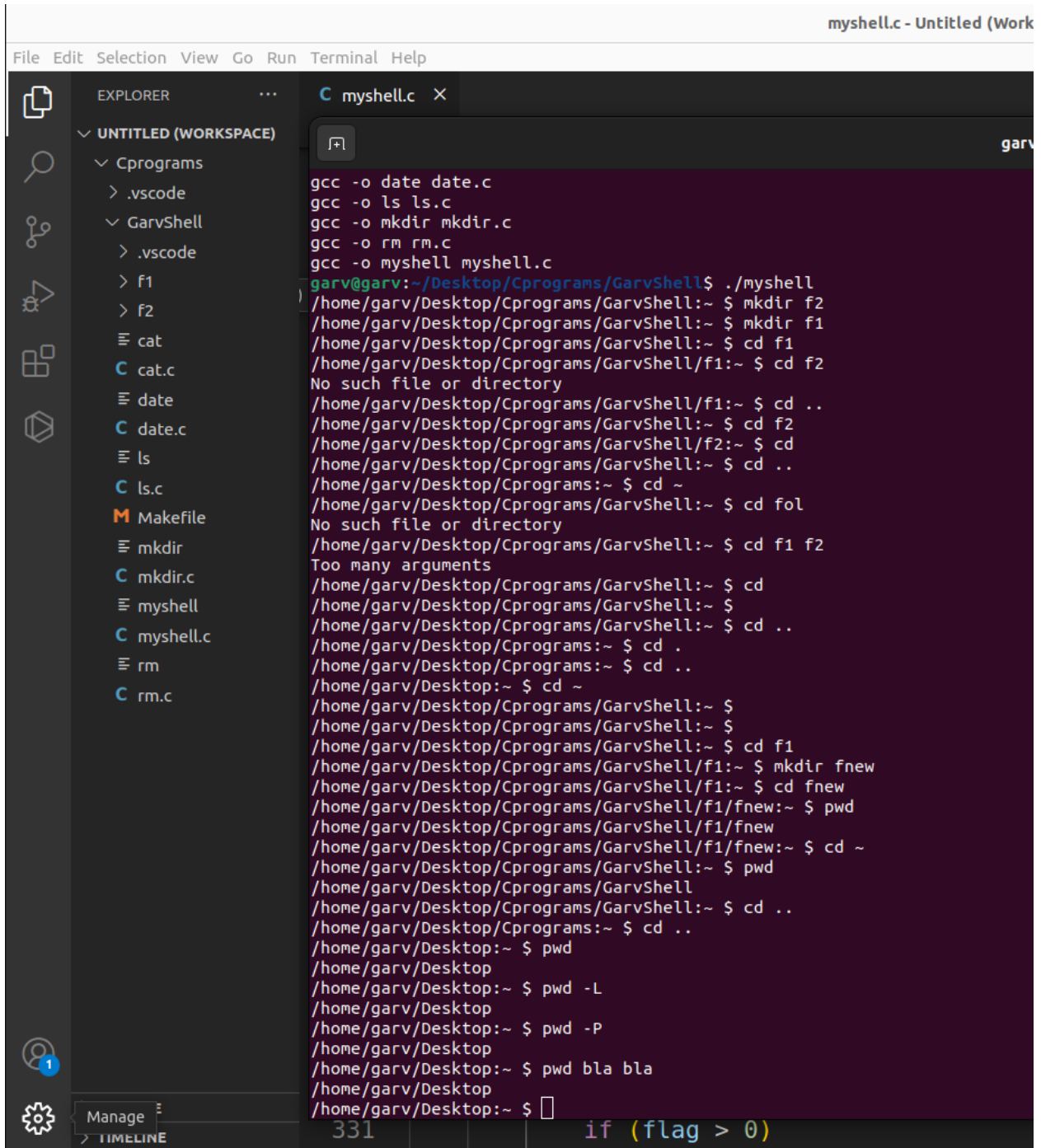
Compiling and running shell



The screenshot shows the Visual Studio Code interface with a terminal window open. The Explorer sidebar on the left displays the project structure, including a folder named 'Cprograms' containing several files: .vscode, GarvShell, .vscode, cat, cat.c, date, date.c, ls, ls.c, Makefile, mkdir, mkdir.c, myshell, myshell.c, rm, and rm.c. The terminal window shows the execution of the 'make' command, which compiles the programs. The output of the terminal is as follows:

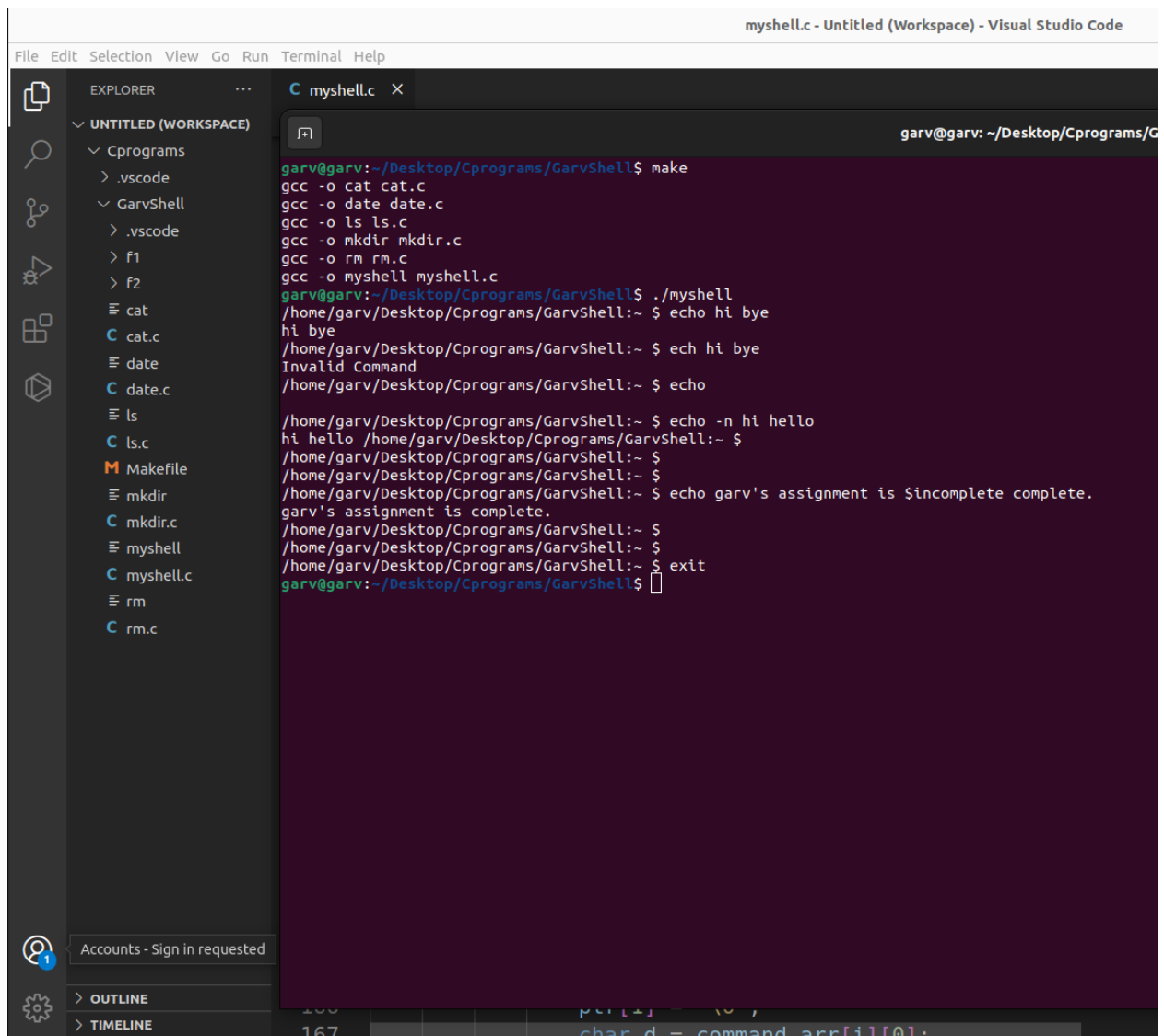
```
garv@garv:~/Desktop/Cprograms/GarvShell$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o myshell myshell.c
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell
/home/garv/Desktop/Cprograms/GarvShell:~ $
```

Working of command cd and pwd



```
myshell.c - Untitled (Work  
File Edit Selection View Go Run Terminal Help  
EXPLORER  
UNTITLED (WORKSPACE)  
Cprograms  
  .vscode  
GarvShell  
  .vscode  
  f1  
  f2  
  cat  
  cat.c  
  date  
  date.c  
  ls  
  ls.c  
  Makefile  
  mkdir  
  mkdir.c  
  myshell  
  myshell.c  
  rm  
  rm.c  
myshell.c  
gcc -o date date.c  
gcc -o ls ls.c  
gcc -o mkdir mkdir.c  
gcc -o rm rm.c  
gcc -o myshell myshell.c  
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell  
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir f2  
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir f1  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd f1  
/home/garv/Desktop/Cprograms/GarvShell/f1:~ $ cd f2  
No such file or directory  
/home/garv/Desktop/Cprograms/GarvShell/f1:~ $ cd ..  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd f2  
/home/garv/Desktop/Cprograms/GarvShell/f2:~ $ cd  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd ..  
/home/garv/Desktop/Cprograms:~ $ cd ~  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd fol  
No such file or directory  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd f1 f2  
Too many arguments  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd  
/home/garv/Desktop/Cprograms/GarvShell:~ $  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd ..  
/home/garv/Desktop/Cprograms:~ $ cd .  
/home/garv/Desktop/Cprograms:~ $ cd ..  
/home/garv/Desktop:~ $ cd ~  
/home/garv/Desktop/Cprograms/GarvShell:~ $  
/home/garv/Desktop/Cprograms/GarvShell:~ $  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd f1  
/home/garv/Desktop/Cprograms/GarvShell/f1:~ $ mkdir fnew  
/home/garv/Desktop/Cprograms/GarvShell/f1:~ $ cd fnew  
/home/garv/Desktop/Cprograms/GarvShell/f1/fnew:~ $ pwd  
/home/garv/Desktop/Cprograms/GarvShell/f1/fnew  
/home/garv/Desktop/Cprograms/GarvShell/f1/fnew:~ $ cd ~  
/home/garv/Desktop/Cprograms/GarvShell:~ $ pwd  
/home/garv/Desktop/Cprograms/GarvShell:~ $ cd ..  
/home/garv/Desktop/Cprograms:~ $ cd ..  
/home/garv/Desktop:~ $ pwd  
/home/garv/Desktop  
/home/garv/Desktop:~ $ pwd -L  
/home/garv/Desktop  
/home/garv/Desktop:~ $ pwd -P  
/home/garv/Desktop  
/home/garv/Desktop:~ $ pwd bla bla  
/home/garv/Desktop  
/home/garv/Desktop:~ $  
331 if (flag > 0)
```


Working of command echo and exit

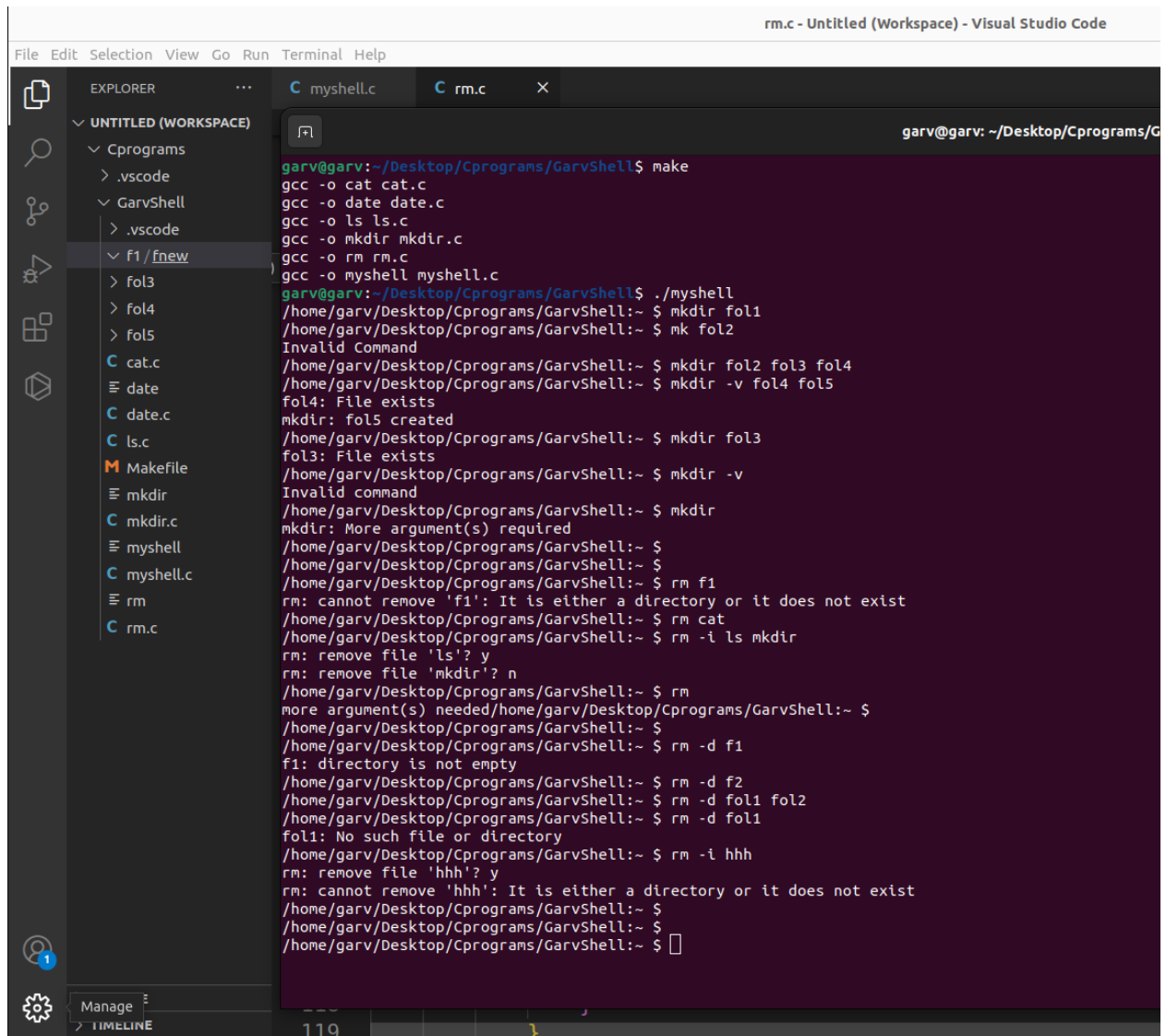


```
myshell.c - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
  UNTITLED (WORKSPACE)
  Cprograms
    .vscode
    GarvShell
      .vscode
      f1
      f2
      cat
      cat.c
      date
      date.c
      ls
      ls.c
      Makefile
      mkdir
      mkdir.c
      myshell
      myshell.c
      rm
      rm.c

garv@garv: ~/Desktop/Cprograms/GarvShell$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o myshell myshell.c
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell
/home/garv/Desktop/Cprograms/GarvShell:~ $ echo hi bye
hi bye
/home/garv/Desktop/Cprograms/GarvShell:~ $ ech hi bye
Invalid Command
/home/garv/Desktop/Cprograms/GarvShell:~ $ echo
/home/garv/Desktop/Cprograms/GarvShell:~ $ echo -n hi hello
hi hello /home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $ echo garv's assignment is $incomplete complete.
garv's assignment is complete.
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $ exit
garv@garv:~/Desktop/Cprograms/GarvShell$
```

Working of command mkdir and rm



```
rm.c - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  UNTITLED (WORKSPACE)
  Cprograms
    .vscode
  GarvShell
    .vscode
    f1/fnew
    fol3
    fol4
    fol5
    cat.c
    date
    date.c
    ls.c
    Makefile
    mkdir
    mkdir.c
    myshell
    myshell.c
    rm
    rm.c
  myshell.c
  rm.c

garv@garv: ~/Desktop/Cprograms/G
garv@garv:~/Desktop/Cprograms/GarvShell$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o myshell myshell.c
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir fol1
/home/garv/Desktop/Cprograms/GarvShell:~ $ mk fol2
Invalid Command
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir fol2 fol3 fol4
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir -v fol4 fol5
fol4: File exists
mkdir: fol5 created
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir fol3
fol3: File exists
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir -v
Invalid command
/home/garv/Desktop/Cprograms/GarvShell:~ $ mkdir
mkdir: More argument(s) required
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm f1
rm: cannot remove 'f1': It is either a directory or it does not exist
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm cat
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -i ls mkdir
rm: remove file 'ls'? y
rm: remove file 'mkdir'? n
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm
more argument(s) needed/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -d f1
f1: directory is not empty
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -d f2
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -d fol1 fol2
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -d fol1
fol1: No such file or directory
/home/garv/Desktop/Cprograms/GarvShell:~ $ rm -i hhh
rm: remove file 'hhh'? y
rm: cannot remove 'hhh': It is either a directory or it does not exist
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
```

Working of command date

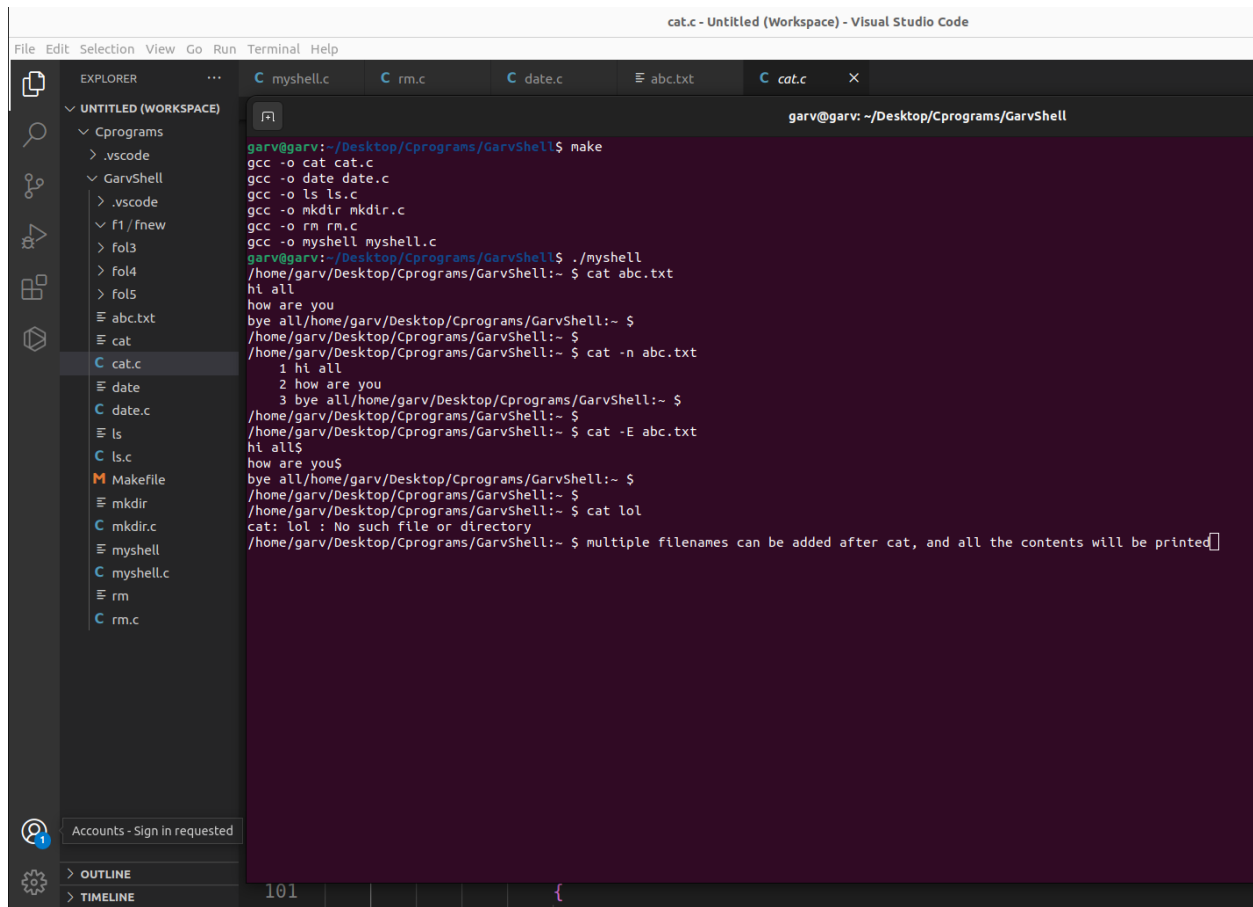
The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the execution of a custom shell named 'myshell' which implements the 'date' command. The user runs 'make' to compile the shell and then './myshell' to start it. Inside the shell, the 'date' command is tested with various options: no options (shows IST and UTC), '-u' (UTC), '-A' (Sunday), 'hi' (Invalid input), 'hi hi' (Invalid input), '-u hi' (Invalid input), '-A hi' (Invalid input), and 'jkkgf' (Invalid input). The 'dt' command is also tested and results in an 'Invalid Command' error.

```
garv@garv: ~/Desktop/Cprograms/GarvShell$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o myshell myshell.c
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell
/home/garv/Desktop/Cprograms/GarvShell:~ $ date
IST Sun Oct 30 18:23:58 2022
/home/garv/Desktop/Cprograms/GarvShell:~ $ date -u
UTC Sun Oct 30 18:24:02 2022
/home/garv/Desktop/Cprograms/GarvShell:~ $ date -A
Sunday
/home/garv/Desktop/Cprograms/GarvShell:~ $ date hi
Invalid input
/home/garv/Desktop/Cprograms/GarvShell:~ $ date hi hi
Invalid input
/home/garv/Desktop/Cprograms/GarvShell:~ $ date -u hi
Invalid input
/home/garv/Desktop/Cprograms/GarvShell:~ $ date -A hi
Invalid input
/home/garv/Desktop/Cprograms/GarvShell:~ $ date jkkgf
Invalid input
/home/garv/Desktop/Cprograms/GarvShell:~ $ dt
Invalid Command
/home/garv/Desktop/Cprograms/GarvShell:~ $
/home/garv/Desktop/Cprograms/GarvShell:~ $
```

Working of command ls

The image shows a Visual Studio Code editor window titled "ls.c - Untitled (Workspace) - Visual Studio Code". The interface includes a sidebar on the left with icons for Explorer, Search, Source Control, and Run and Debug. The Explorer sidebar shows a project structure with folders "Cprograms" and "GarvShell", and various files including ".vscode", "f1/fnew", "fol3", "fol4", "fol5", "cat", "cat.c", "date", "date.c", "ls", "ls.c", "ls.c", "Makefile", "mkdir", "mkdir.c", "myshell", "myshell.c", "rm", and "rm.c". The main editor area displays a terminal window with the prompt "garv@garv: ~/Desktop/Cprograms/GarvShell". The terminal output shows the execution of the "make" command, which compiles several C files (cat.c, date.c, ls.c, mkdtr.c, rm.c, myshell.c) and creates a Makefile. The output also shows the execution of the program with the command "cat /home/garv/Desktop/Cprograms/GarvShell:~ \$ ls -l", which results in an "Invalid Command" error. The terminal prompt is "garv@garv: ~/Desktop/Cprograms/GarvShell".

Working of command cat



cat.c - Untitled (Workspace) - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

UNTITLED (WORKSPACE)

Cprograms

GarvShell

fol3

fol4

fol5

abc.txt

cat

cat.c

date

date.c

ls

ls.c

Makefile

mkdir

mkdir.c

myshell

myshell.c

rm

rm.c

garv@garv: ~/Desktop/Cprograms/GarvShell

```
garv@garv:~/Desktop/Cprograms/GarvShell$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o myshell myshell.c
garv@garv:~/Desktop/Cprograms/GarvShell$ ./myshell
/hone/garv/Desktop/Cprograms/GarvShell:- $ cat abc.txt
hi all
how are you
bye all/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $ cat -n abc.txt
1 hi all
2 how are you
3 bye all/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $ cat -E abc.txt
hi all$
how are you$
bye all/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $
/hone/garv/Desktop/Cprograms/GarvShell:- $ cat lol
cat: lol : No such file or directory
/hone/garv/Desktop/Cprograms/GarvShell:- $ multiple filenames can be added after cat, and all the contents will be printed
```

101 {