

ASSIGNMENT 2: DOCUMENTATION:

SUMMARY:

This assignment shows a command line interface of Unix/Linux. The program is written in C language. Initially there is a while loop for user input until there is an exit call. Inside every external command there is a `pid` process to check status and how the program's child process works. Usage of `fork()` and `execv()` is to initiate a child process and call the external command from a different C file. The `execv` takes two arguments; one is the executable file of the external command present in a different C file and second is the input arguments to the C file. In my case the input argument is null because every input is taken in the main of C file and not passed to main. Once the child process is executed the parent process waits for null time and nothing operates in the parent process.

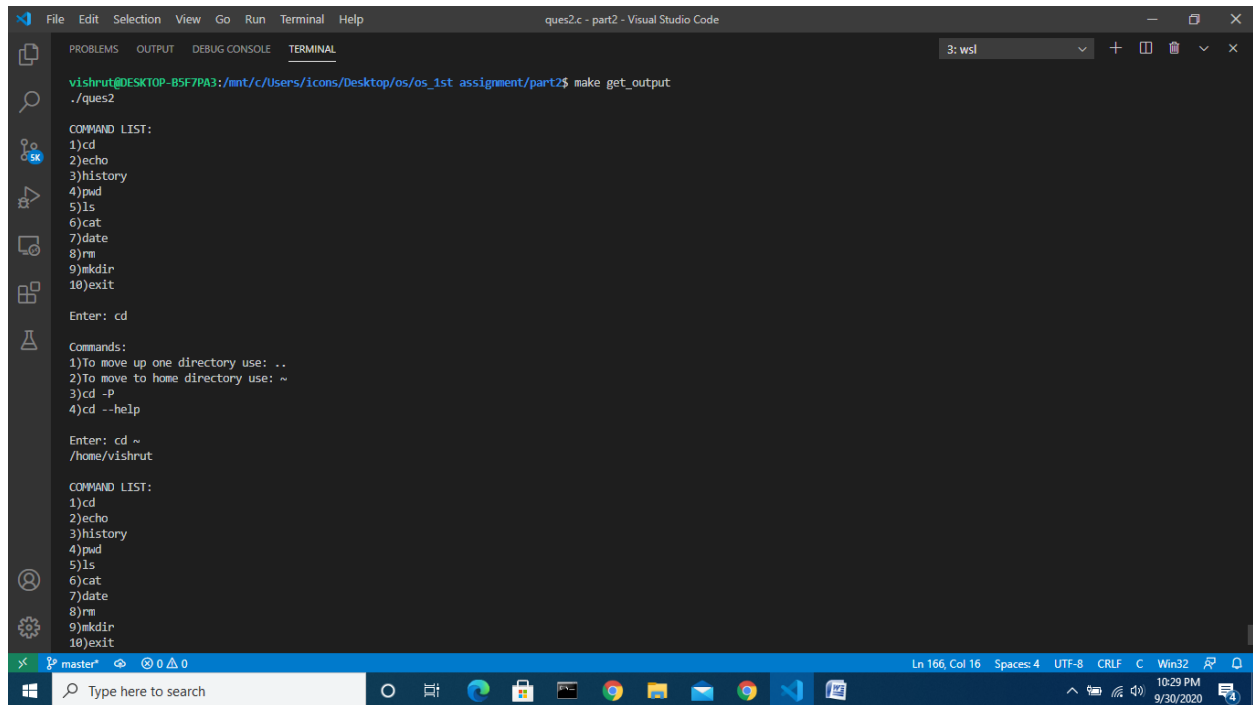
ERROR REPORTING:

Next we move to the commands. Each command has at least 2 options to work on and handled error reporting strategy. "Invalid syntax" or "Invalid command" or "Error" is shown once there is reporting. In all these cases the program exits. In commands like "rm" and "mkdir" may show file error and directory cannot be created respectively. In such cases the command is exited not the program. Every command has error of syntax and many sub-errors inside as given examples previously.

Some other features: History.txt file has the history of every command. Delimiter has been used. Makefile for compilation and pausing at each phase. Although running the program makes it very clear. Also all internal commands and external commands can run in one go but there is an assumption that try not to run an external command after an internal due to over-reading between internal to external. Apart from that all internal commands can run after external easily. All warnings must be ignored.

Screenshots of all commands and their basic working is attached for your reference:-

cd command



The screenshot shows a Visual Studio Code window with a terminal open. The terminal title is "ques2.c - part2 - Visual Studio Code". The terminal content shows the user running the command `make get_output` in the directory `./ques2`. A command list is displayed, including `cd`, `echo`, `history`, `pwd`, `ls`, `cat`, `date`, `rm`, `mkdir`, and `exit`. The user enters `cd`, and the terminal shows the command's purpose: to move up one directory (`..`) or to the home directory (`~`). The user then enters `cd ~`, and the terminal shows the current directory as `/home/vishrut`. The status bar at the bottom indicates the file is at line 166, column 16, with 4 spaces, UTF-8 encoding, CRLF line endings, and the Windows 32-bit architecture.

```
File Edit Selection View Go Run Terminal Help
ques2.c - part2 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
vishrut@DESKTOP-B5F7PA3:/mnt/c/Users/Icons/Desktop/os/os_1st assignment/part2$ make get_output
./ques2

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: cd

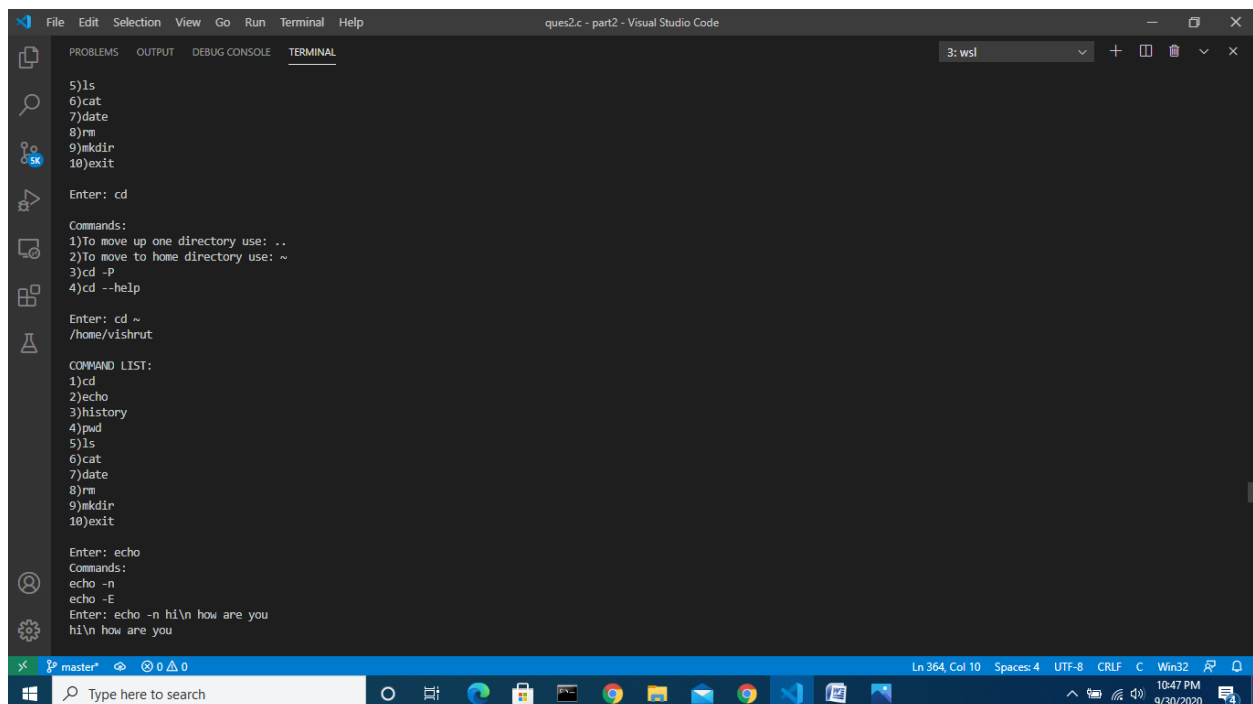
Commands:
1)To move up one directory use: ..
2)To move to home directory use: ~
3)cd -P
4)cd --help

Enter: cd ~
/home/vishrut

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Ln 166, Col 16 Spaces: 4 UTF-8 CRLF C Win32 10:29 PM 9/30/2020
```

echo command



The screenshot shows a Visual Studio Code window with a terminal open. The terminal title is "ques2.c - part2 - Visual Studio Code". The terminal content shows the user running the command `make get_output` in the directory `./ques2`. A command list is displayed, including `cd`, `echo`, `history`, `pwd`, `ls`, `cat`, `date`, `rm`, `mkdir`, and `exit`. The user enters `cd`, and the terminal shows the command's purpose: to move up one directory (`..`) or to the home directory (`~`). The user then enters `cd ~`, and the terminal shows the current directory as `/home/vishrut`. The user then enters `echo`, and the terminal shows the command's purpose: to echo the output. The user then enters `echo -n`, `echo -E`, and `echo -n hi\n how are you`, and the terminal shows the output: `hi\n how are you`. The status bar at the bottom indicates the file is at line 364, column 10, with 4 spaces, UTF-8 encoding, CRLF line endings, and the Windows 32-bit architecture.

```
File Edit Selection View Go Run Terminal Help
ques2.c - part2 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: cd

Commands:
1)To move up one directory use: ..
2)To move to home directory use: ~
3)cd -P
4)cd --help

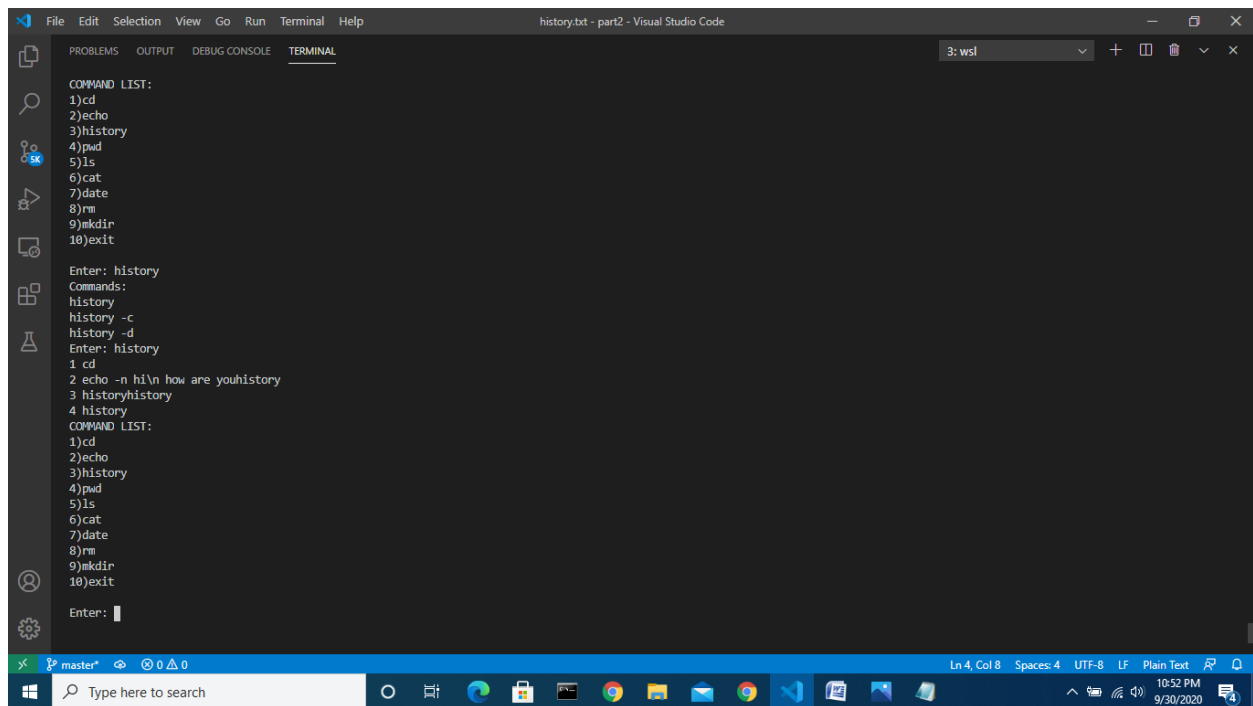
Enter: cd ~
/home/vishrut

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: echo
Commands:
echo -n
echo -E
Enter: echo -n hi\n how are you
hi\n how are you

Ln 364, Col 10 Spaces: 4 UTF-8 CRLF C Win32 10:47 PM 9/30/2020
```

history command



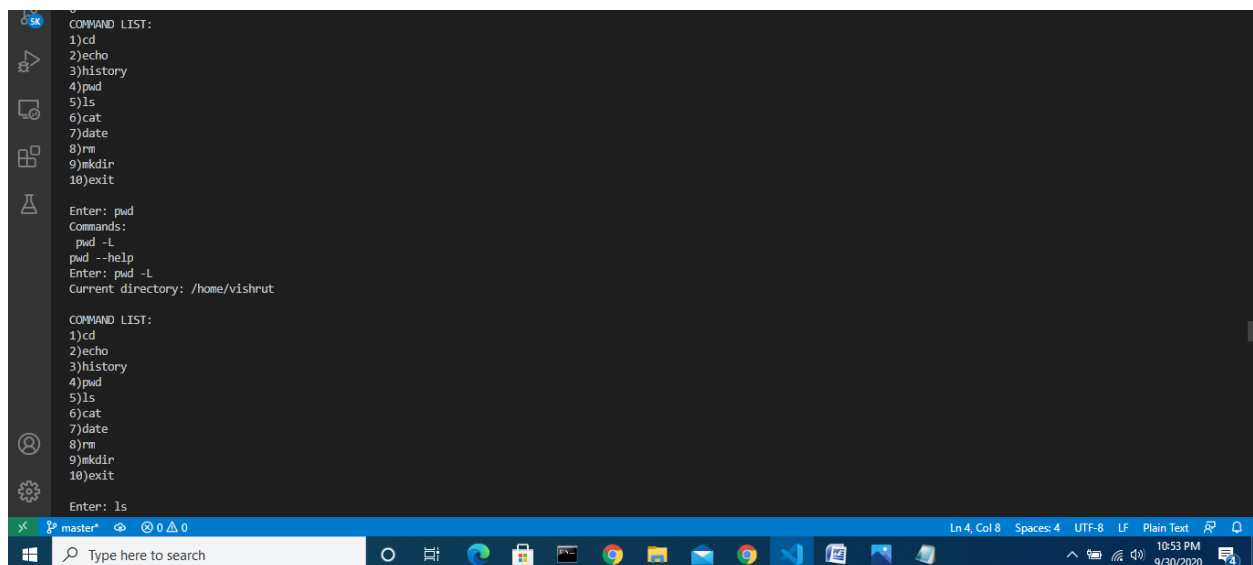
The screenshot shows a Visual Studio Code terminal window titled "history.txt - part2 - Visual Studio Code". The terminal is running a shell (wsl) and displays the output of the `history` command. The output shows a list of commands entered, including `cd`, `echo`, `history`, `pwd`, `ls`, `cat`, `date`, `rm`, `mkdir`, and `exit`. The terminal also shows the command list and the current directory.

```
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: history
Commands:
history
history -c
history -d
Enter: history
1 cd
2 echo -n hi\n how are youhistory
3 historyhistory
4 history
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: 
```

pwd command



The screenshot shows a Visual Studio Code terminal window titled "history.txt - part2 - Visual Studio Code". The terminal is running a shell (wsl) and displays the output of the `pwd` command. The output shows the current directory as `/home/vishrut`. The terminal also shows the command list and the current directory.

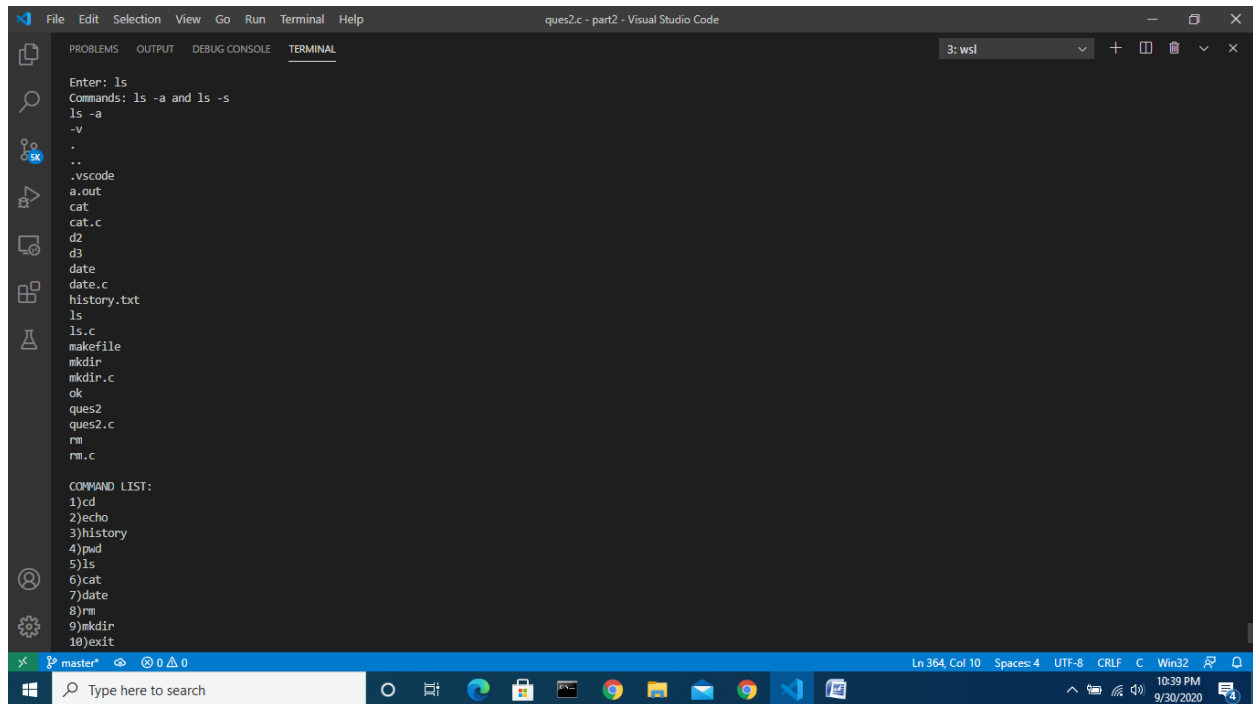
```
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: pwd
Commands:
pwd -L
pwd --help
Enter: pwd -L
Current directory: /home/vishrut

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: ls
```

ls command

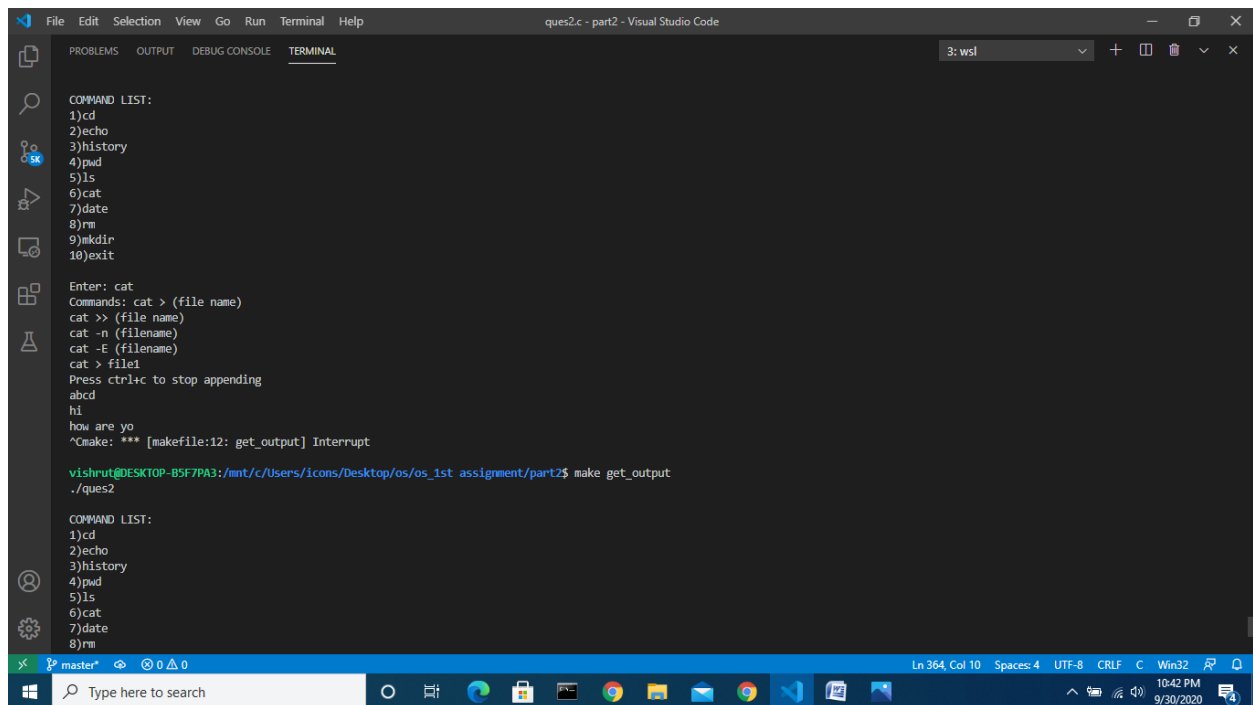


The screenshot shows a Visual Studio Code window with a terminal open. The terminal title is "3: wsl". The command prompt shows "Enter: ls" and "Commands: ls -a and ls -s". The output of the "ls -a" command is displayed, listing files and directories including hidden ones like ".vscode", ".history.txt", and ".ls.c". Below the output, a "COMMAND LIST:" is shown with 10 numbered commands: 1)cd, 2)echo, 3)history, 4)pwd, 5)ls, 6)cat, 7)date, 8)rm, 9)mkdir, and 10)exit. The status bar at the bottom indicates "Ln 364, Col 10" and "Spaces: 4".

```
Enter: ls
Commands: ls -a and ls -s
ls -a
-v
.
..
.vscode
a.out
cat
cat.c
d2
d3
date
date.c
history.txt
ls
ls.c
makefile
mkdir
mkdir.c
ok
ques2
ques2.c
rm
rm.c

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit
```

cat command

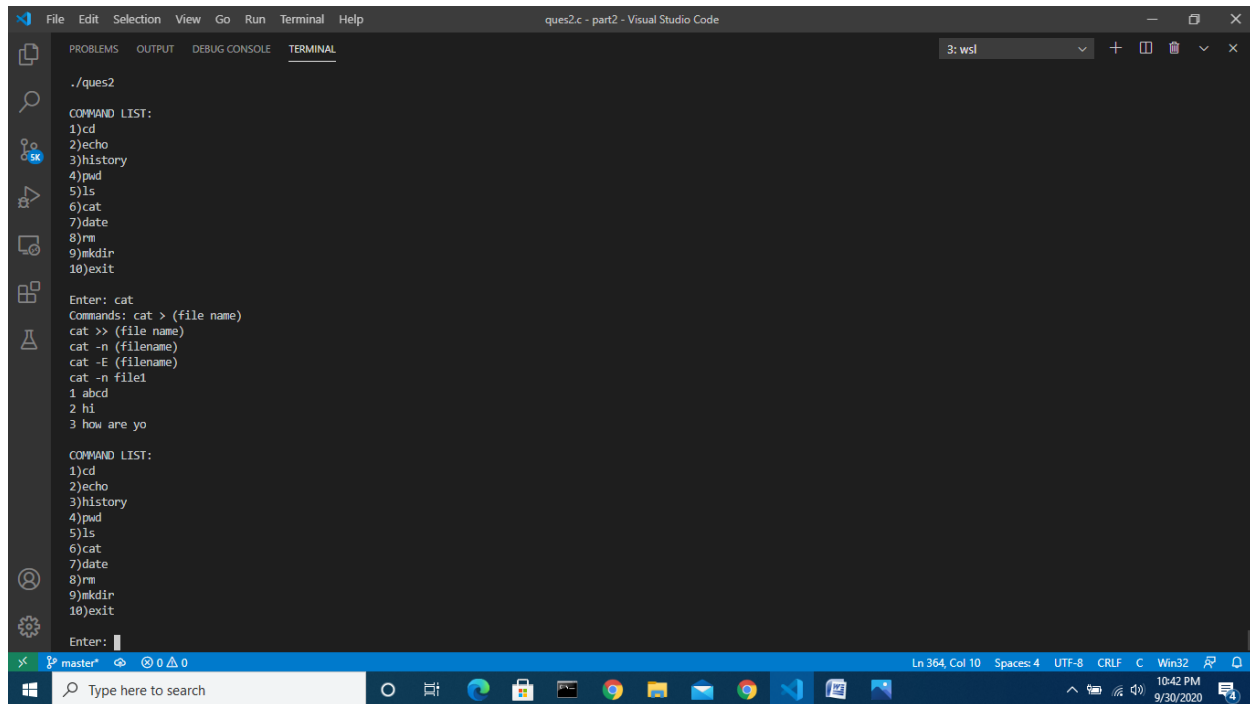


The screenshot shows a Visual Studio Code window with a terminal open. The terminal title is "3: wsl". The command prompt shows "Enter: cat" and "Commands: cat > (file name)". The output of the "cat" command is displayed, showing the contents of a file named "file1". The output includes the text "abcd", "hi", and "how are yo". Below the output, a "COMMAND LIST:" is shown with 8 numbered commands: 1)cd, 2)echo, 3)history, 4)pwd, 5)ls, 6)cat, 7)date, and 8)rm. The status bar at the bottom indicates "Ln 364, Col 10" and "Spaces: 4".

```
Enter: cat
Commands: cat > (file name)
cat >> (file name)
cat -n (filename)
cat -E (filename)
cat > file1
Press ctrl+c to stop appending
abcd
hi
how are yo
^Cmake: *** [makefile:12: get_output] Interrupt

vishrut@DESKTOP-B5F7PA3:/mnt/c/Users/vishrut/Desktop/os/os_1st assignment/part2$ make get_output
./ques2

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
```



The screenshot shows a Visual Studio Code window with a terminal pane. The terminal is running a Windows Subsystem for Linux (WSL) environment. The user has entered the command `cat` and is prompted for a file name. The terminal shows the command list and the output of the `cat` command, which displays the contents of a file named `file1`.

```
./ques2

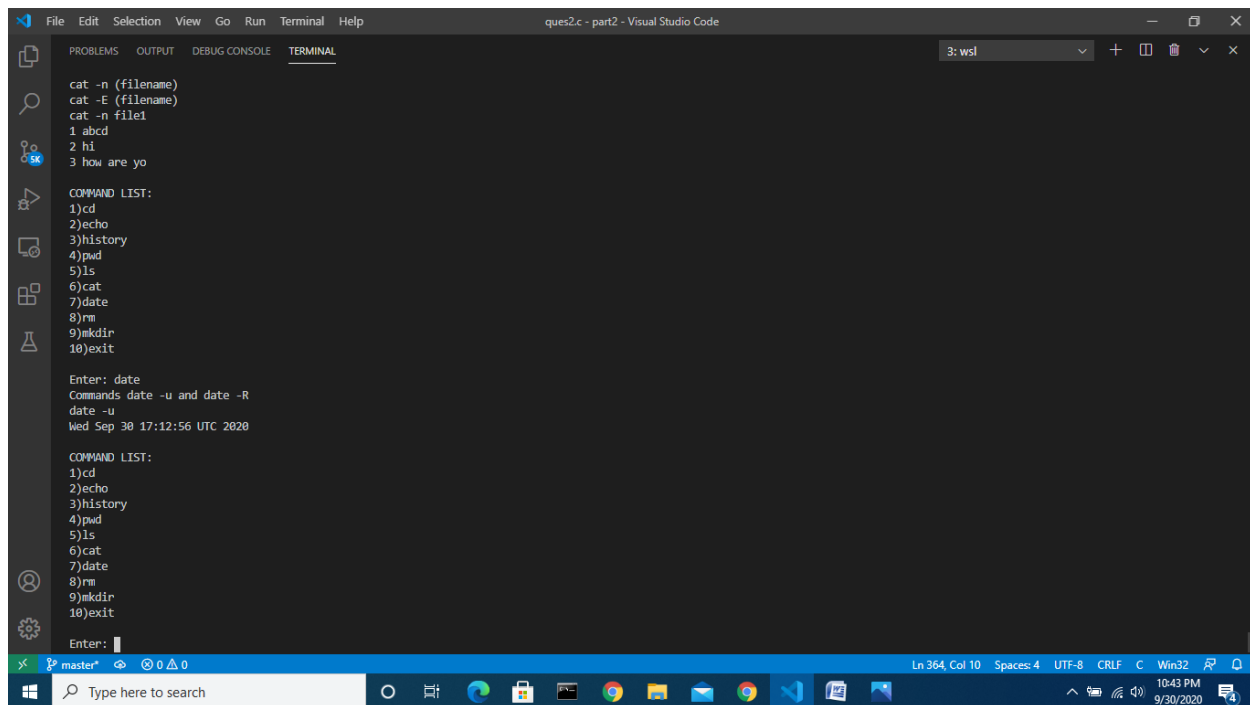
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: cat
Commands: cat > (file name)
cat >> (file name)
cat -n (filename)
cat -E (filename)
cat -n file1
1 abcd
2 hi
3 how are yo

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: 
```

date command



The screenshot shows a Visual Studio Code window with a terminal pane. The terminal is running a Windows Subsystem for Linux (WSL) environment. The user has entered the command `date` and is prompted for a file name. The terminal shows the command list and the output of the `date` command, which displays the current date and time.

```
cat -n (filename)
cat -E (filename)
cat -n file1
1 abcd
2 hi
3 how are yo

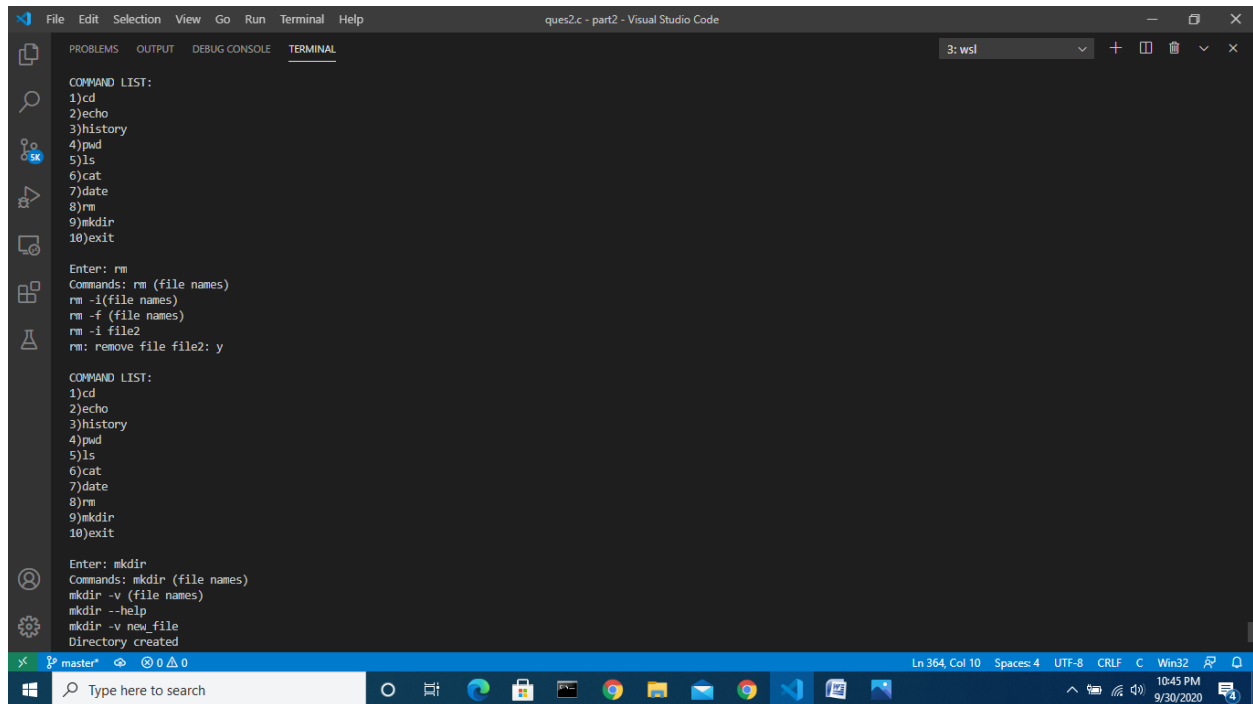
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: date
Commands: date -u and date -R
date -u
Wed Sep 30 17:12:56 UTC 2020

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: 
```

rm command



The screenshot shows a Visual Studio Code terminal window with the title "ques2.c - part2 - Visual Studio Code". The terminal is running a shell (3: wsl) and displays a command list and the usage of the 'rm' command. The command list includes: 1)cd, 2)echo, 3)history, 4)pwd, 5)ls, 6)cat, 7)date, 8)rm, 9)mkdir, and 10)exit. The user enters 'rm', and the terminal shows the commands: rm (file names), rm -i (file names), rm -f (file names), and rm -i file2. The user then enters 'rm: remove file file2: y'.

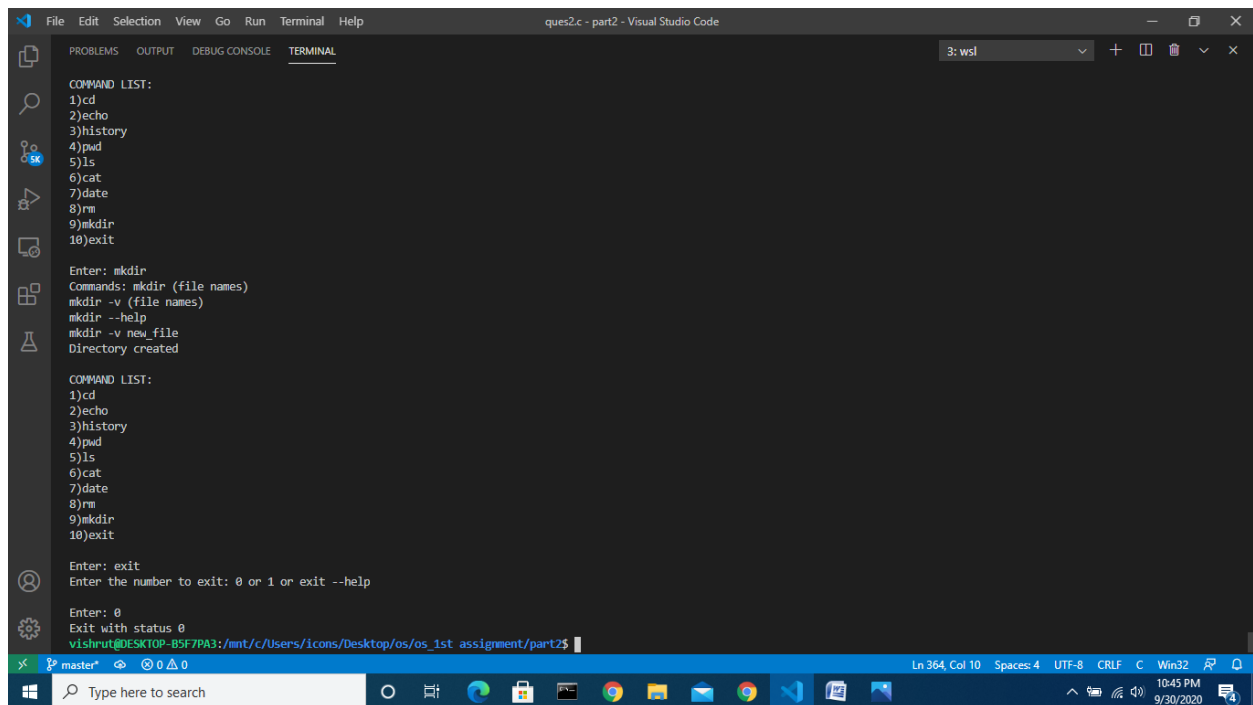
```
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: rm
Commands: rm (file names)
rm -i (file names)
rm -f (file names)
rm -i file2
rm: remove file file2: y

COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: mkdir
Commands: mkdir (file names)
mkdir -v (file names)
mkdir --help
mkdir -v new_file
Directory created
```

mkdir command and exit command



The screenshot shows a Visual Studio Code terminal window with the title "ques2.c - part2 - Visual Studio Code". The terminal is running a shell (3: wsl) and displays a command list and the usage of the 'mkdir' and 'exit' commands. The command list includes: 1)cd, 2)echo, 3)history, 4)pwd, 5)ls, 6)cat, 7)date, 8)rm, 9)mkdir, and 10)exit. The user enters 'mkdir', and the terminal shows the commands: mkdir (file names), mkdir -v (file names), mkdir --help, and mkdir -v new_file. The user then enters 'mkdir: Directory created'. The user enters 'exit', and the terminal shows the command: exit. The user then enters '0', and the terminal shows the command: Exit with status 0. The user then enters 'vishrut@DESKTOP-B5F7PA3:/mnt/c/Users/Icons/Desktop/os_1st_assignment/part2\$'.

```
COMMAND LIST:
1)cd
2)echo
3)history
4)pwd
5)ls
6)cat
7)date
8)rm
9)mkdir
10)exit

Enter: mkdir
Commands: mkdir (file names)
mkdir -v (file names)
mkdir --help
mkdir -v new_file
Directory created

Enter: exit
Enter the number to exit: 0 or 1 or exit --help

Enter: 0
Exit with status 0
vishrut@DESKTOP-B5F7PA3:/mnt/c/Users/Icons/Desktop/os_1st_assignment/part2$
```