UNIX Utilities Overview

This shell project implements several Unix-like commands (ls, cat, grep, wc, cp, mv, rm, and time) in the ./bin/ directory. The shell interacts with these commands by providing functionality similar to a lightweight terminal. Each command is implemented in C and supports common options or features associated with the original Unix commands.

Table of Contents:

1.Setup Instructions

Clone or Copy Files: Copy the provided code into a working directory.

Compile the Commands: Use the provided Makefile to compile all the commands.

```
Run:
        bash
        make all
```

This will generate executable files in the ./bin/ directory.

Run the Shell: Launch the shell using:

bash

```
./bin/shell
```

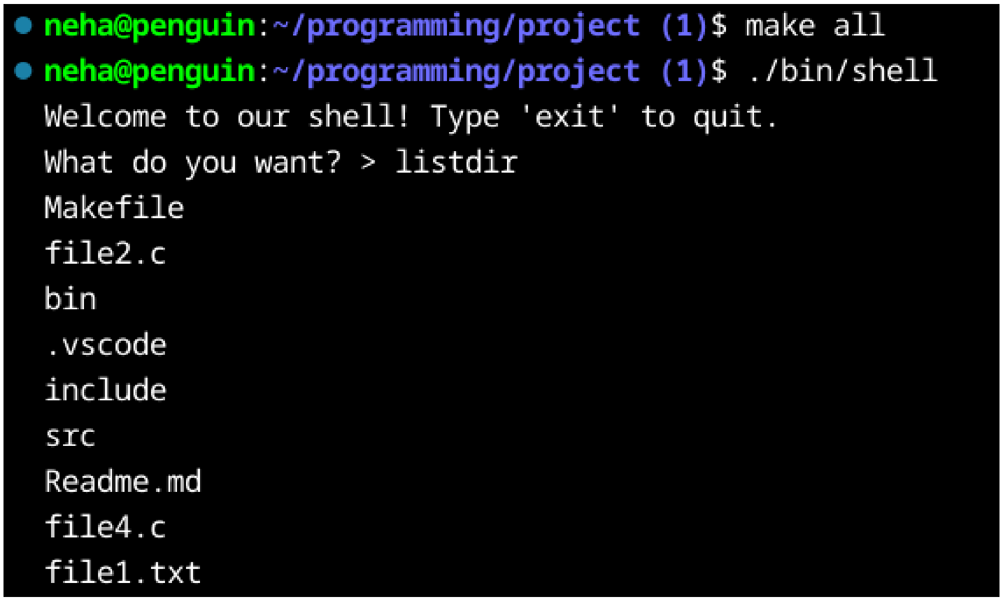
2. Commands Overview

->listdir

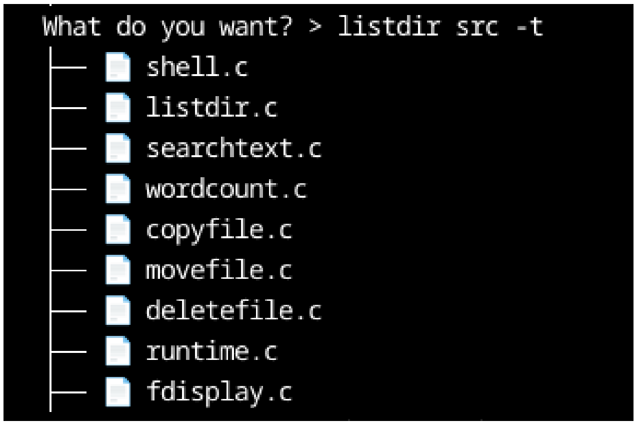
Description: Lists files and directories in the current or specified directory.

```
Usage:
        bash
      listdir [OPTION] [DIRECTORY]
```

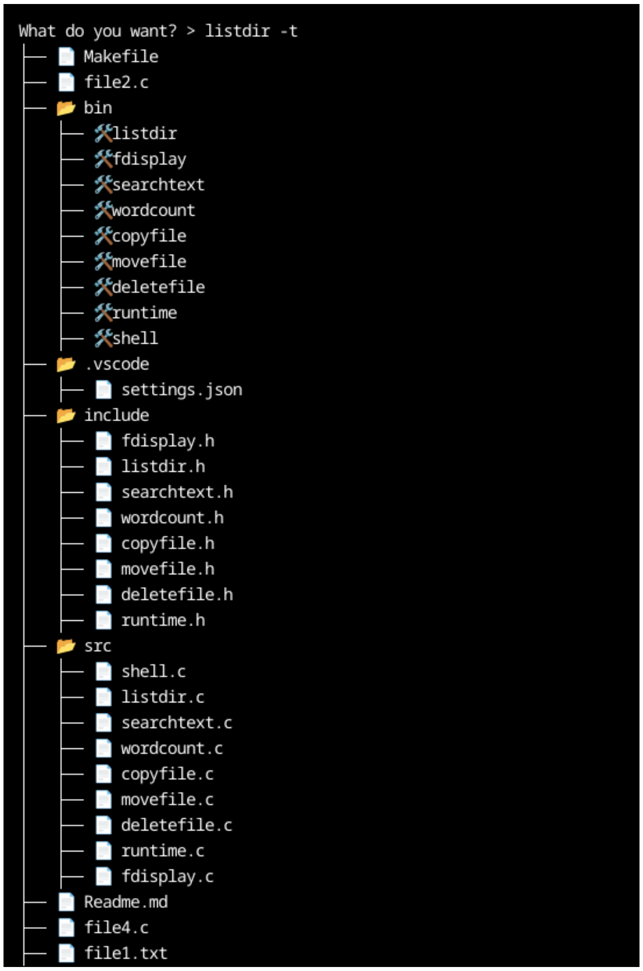
Options:
-a: Show all files (including hidden files).
-l: Display detailed information about files.
Example:
  bash
  listdir -l



```
● neha@penguin:~/programming/project (1)$ make all
● neha@penguin:~/programming/project (1)$ ./bin/shell
  Welcome to our shell! Type 'exit' to quit.
  What do you want? > listdir
  Makefile
  file2.c
  bin
  .vscode
  include
  src
  Readme.md
  file4.c
  file1.txt
```



```
What do you want? > listdir src -t
├── 📄 shell.c
├── 📄 listdir.c
├── 📄 searchtext.c
├── 📄 wordcount.c
├── 📄 copyfile.c
├── 📄 movefile.c
├── 📄 deletefile.c
├── 📄 runtime.c
├── 📄 fdisplay.c
```



```
What do you want? > listdir -t
├── 📄 Makefile
├── 📄 file2.c
├── 📂 bin
│   ├── 🛠listdir
│   ├── 🛠fdisplay
│   ├── 🛠searchtext
│   ├── 🛠wordcount
│   ├── 🛠copyfile
│   ├── 🛠movefile
│   ├── 🛠deletefile
│   ├── 🛠runtime
│   ├── 🛠shell
├── 📂 .vscode
│   ├── 📄 settings.json
├── 📂 include
│   ├── 📄 fdisplay.h
│   ├── 📄 listdir.h
│   ├── 📄 searchtext.h
│   ├── 📄 wordcount.h
│   ├── 📄 copyfile.h
│   ├── 📄 movefile.h
│   ├── 📄 deletefile.h
│   ├── 📄 runtime.h
├── 📂 src
│   ├── 📄 shell.c
│   ├── 📄 listdir.c
│   ├── 📄 searchtext.c
│   ├── 📄 wordcount.c
│   ├── 📄 copyfile.c
│   ├── 📄 movefile.c
│   ├── 📄 deletefile.c
│   ├── 📄 runtime.c
│   ├── 📄 fdisplay.c
├── 📄 Readme.md
├── 📄 file4.c
├── 📄 file1.txt
```

->fdisplay

Description: Displays the content of a file.

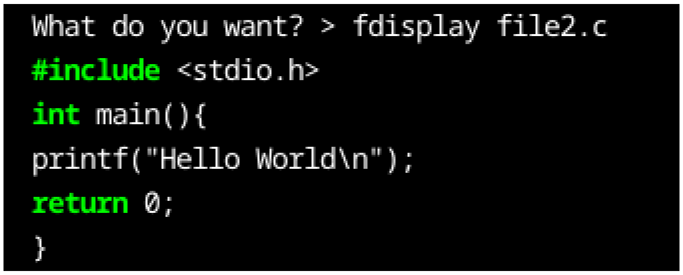
Usage:

bash

```
fdisplay [FILE]
```

Example:

bash
fdisplay file.txt



```
What do you want? > fdisplay file2.c
#include <stdio.h>
int main(){
printf("Hello World\n");
return 0;
}
```

->searchtext

Description: Searches for a pattern in a file and prints matching lines.

Usage:

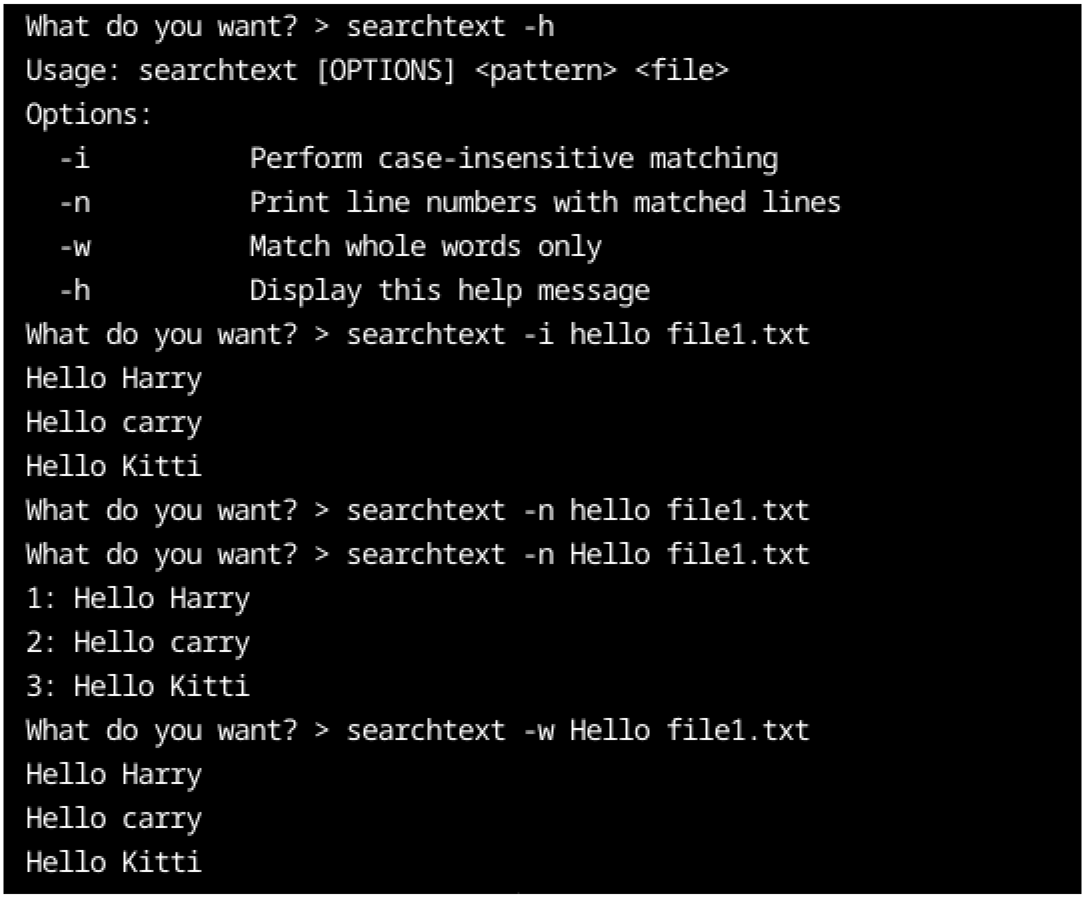
bash

searchtext [PATTERN] [FILE]

Example:

bash

searchtext "hello" file.txt



```
What do you want? > searchtext -h
Usage: searchtext [OPTIONS] <pattern> <file>
Options:
  -i         Perform case-insensitive matching
  -n         Print line numbers with matched lines
  -w         Match whole words only
  -h         Display this help message
What do you want? > searchtext -i hello file1.txt
Hello Harry
Hello carry
Hello Kitti
What do you want? > searchtext -n hello file1.txt
What do you want? > searchtext -n Hello file1.txt
1: Hello Harry
2: Hello carry
3: Hello Kitti
What do you want? > searchtext -w Hello file1.txt
Hello Harry
Hello carry
Hello Kitti
```

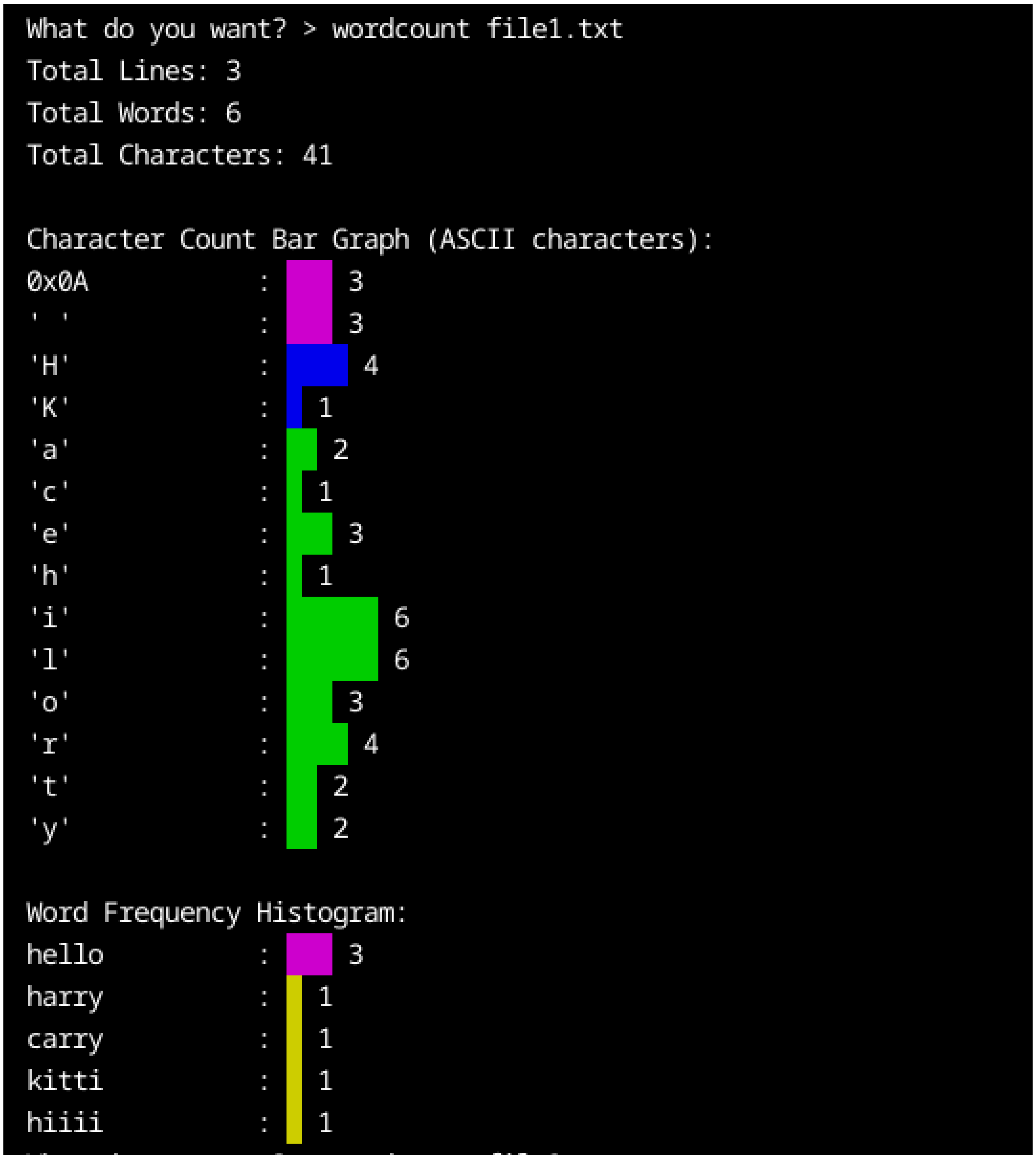
->wordcount

Description: Counts lines, words, and characters in a file. Displays histograms of word and character frequency.

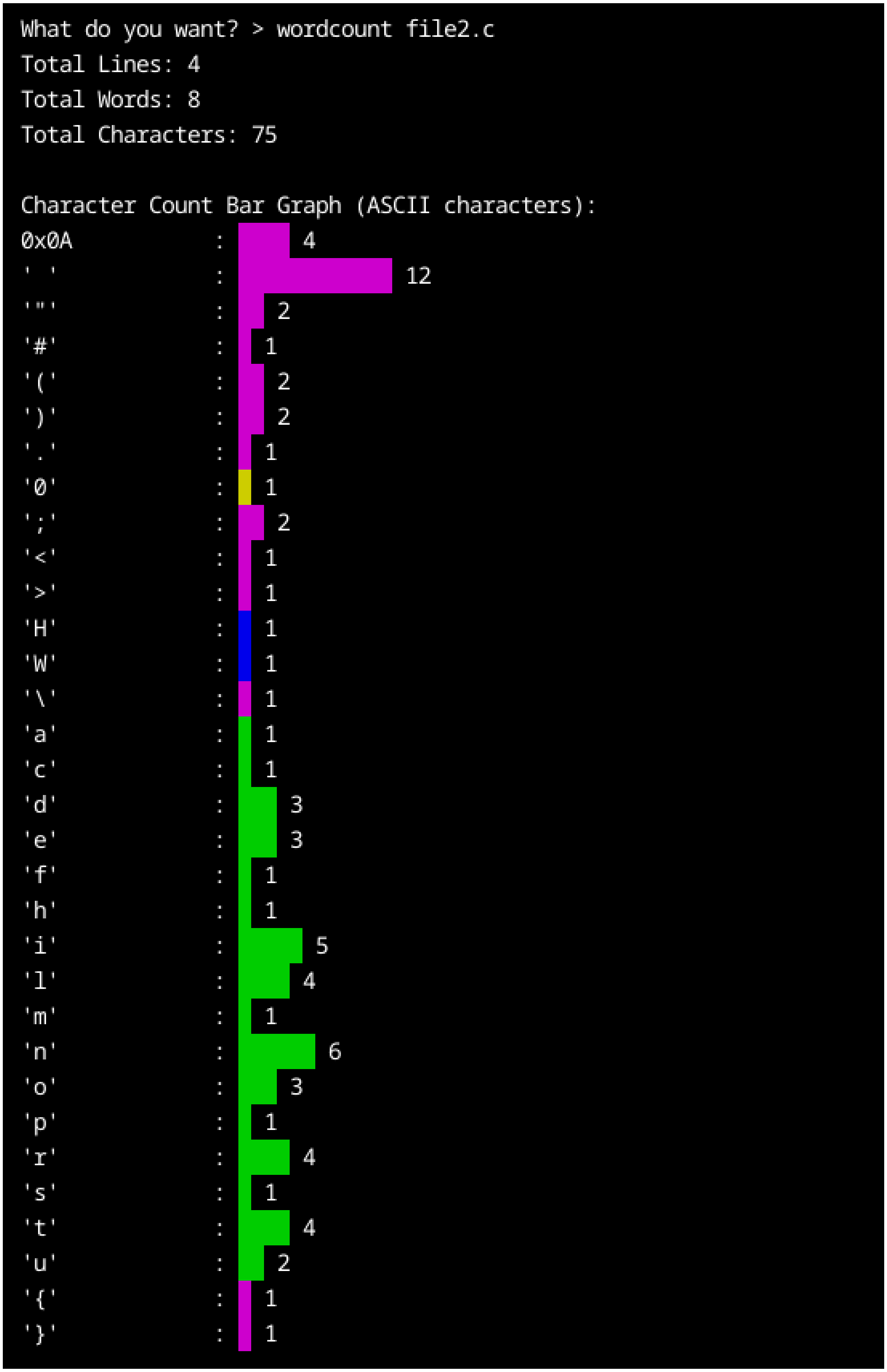
```
Usage:
    bash
    wordcount [FILE]
```

Example:

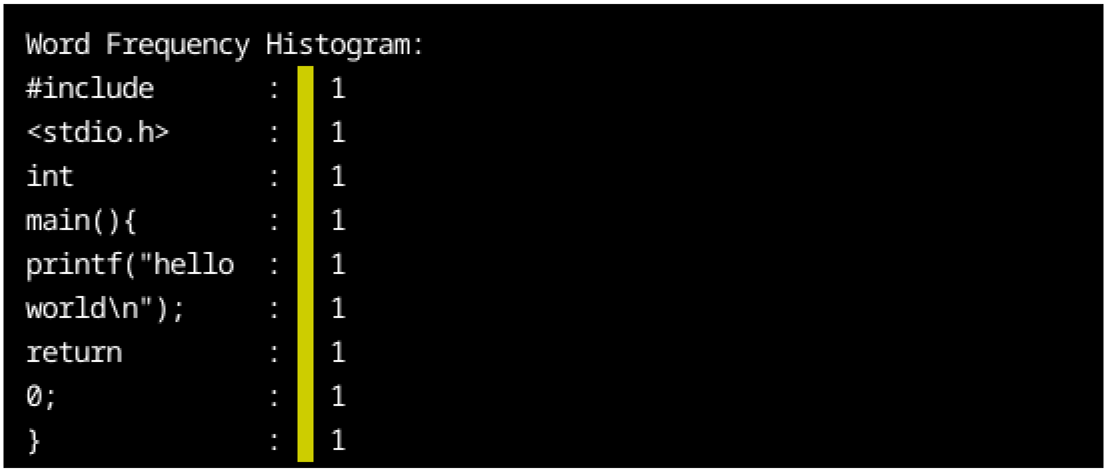
bash
wordcount file.txt



```
What do you want? > wordcount file1.txt
Total Lines: 3
Total Words: 6
Total Characters: 41

Character Count Bar Graph (ASCII characters):
0x0A           : ███ 3
' '            : ███ 3
'H'            : ████ 4
'K'            : █ 1
'a'            : ██ 2
'c'            : █ 1
'e'            : ███ 3
'h'            : █ 1
'i'            : ██████ 6
'l'            : ██████ 6
'o'            : ███ 3
'r'            : ████ 4
't'            : ██ 2
'y'            : ██ 2

Word Frequency Histogram:
hello          : ███ 3
harry          : █ 1
carry          : █ 1
kitti          : █ 1
hiiii          : █ 1
```



```
What do you want? > wordcount file2.c
Total Lines: 4
Total Words: 8
Total Characters: 75

Character Count Bar Graph (ASCII characters):
0x0A           : ████ 4
' '            : ████████████ 12
'"'            : ██ 2
'#'            : █ 1
'('            : ██ 2
')'            : ██ 2
'.'            : █ 1
'0'            : █ 1
';'            : ██ 2
'<'            : █ 1
'>'            : █ 1
'H'            : █ 1
'W'            : █ 1
'\'            : █ 1
'a'            : █ 1
'c'            : █ 1
'd'            : ███ 3
'e'            : ███ 3
'f'            : █ 1
'h'            : █ 1
'i'            : █████ 5
'l'            : ████ 4
'm'            : █ 1
'n'            : ██████ 6
'o'            : ███ 3
'p'            : █ 1
'r'            : ████ 4
's'            : █ 1
't'            : ████ 4
'u'            : ██ 2
'{'            : █ 1
'}'            : █ 1
```



```
Word Frequency Histogram:
#include       : █ 1
<stdio.h>      : █ 1
int            : █ 1
main(){        : █ 1
printf("hello  : █ 1
world\n");     : █ 1
return         : █ 1
0;             : █ 1
}              : █ 1
```

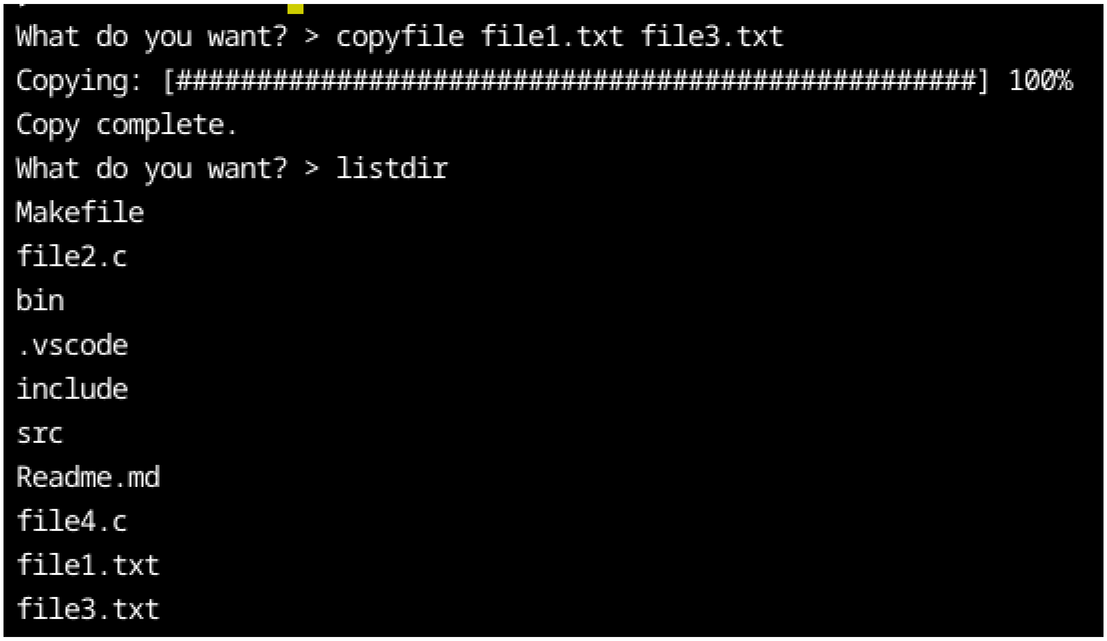
->copyfile

Description: Copies a file with progress indication.

```
Usage:
    bash
    copyfile [SOURCE_FILE] [DEST_FILE]
```

Example:

```
  bash
  copyfile source.txt dest.txt
```



```
What do you want? > copyfile file1.txt file3.txt
Copying: [##################################################] 100%
Copy complete.
What do you want? > listdir
Makefile
file2.c
bin
.vscode
include
src
Readme.md
file4.c
file1.txt
file3.txt
```

->movefile

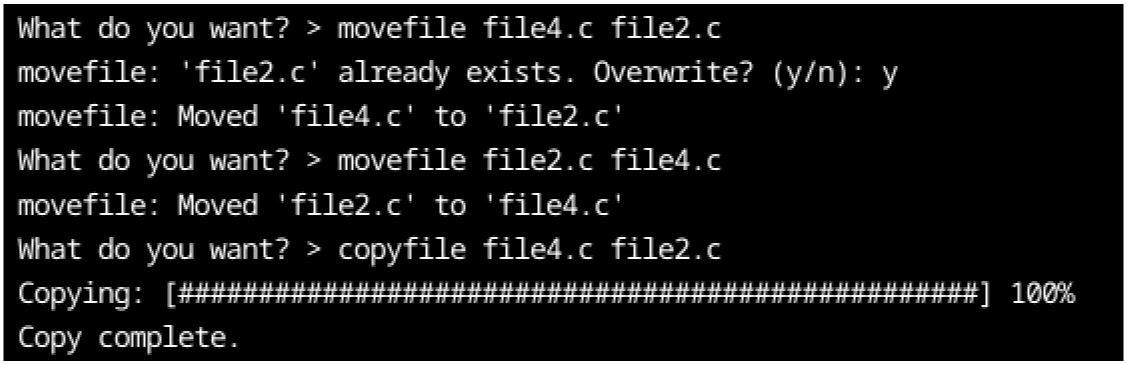
Description: Moves or renames a file.

```
Usage:
    bash
    movefile [SOURCE_FILE] [DEST_FILE]
```

Example:

bash

movefile old_name.txt new_name.txt



```
What do you want? > movefile file4.c file2.c
movefile: 'file2.c' already exists. Overwrite? (y/n): y
movefile: Moved 'file4.c' to 'file2.c'
What do you want? > movefile file2.c file4.c
movefile: Moved 'file2.c' to 'file4.c'
What do you want? > copyfile file4.c file2.c
Copying: [##################################################] 100%
Copy complete.
```

->deletefile

Description: Removes files or directories (recursively if required).

```
Usage:
    bash
    deletefile [-r] [FILE_OR_DIRECTORY]
```

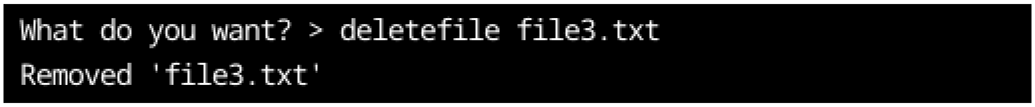
Options:

-r: Recursively remove directories.

Example:

bash

deletefile -r folder_name



```
What do you want? > deletefile file3.txt
Removed 'file3.txt'
```

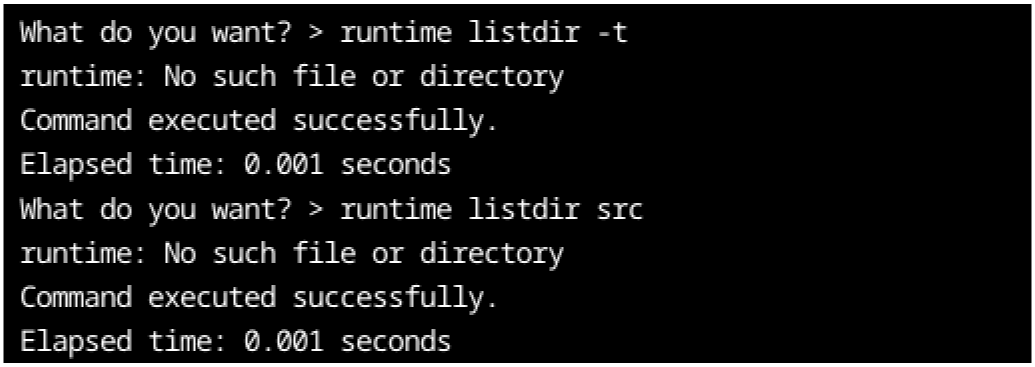
->runtime

Description: Measures the time taken to execute another command.

```
Usage:
    bash
    runtime [COMMAND] [ARGS...]
```

Example:

bash
runtime listdir -l



```
What do you want? > runtime listdir -t
runtime: No such file or directory
Command executed successfully.
Elapsed time: 0.001 seconds
What do you want? > runtime listdir src
runtime: No such file or directory
Command executed successfully.
Elapsed time: 0.001 seconds
```

->Shell

Description: An interactive shell (shell) to execute the above commands.

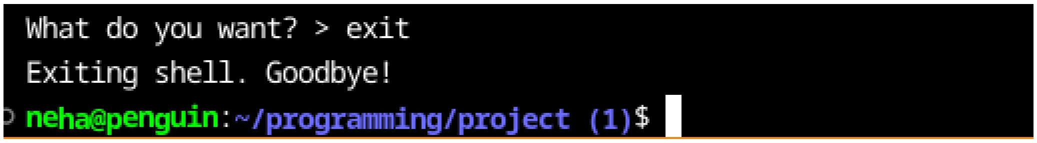
How to Use:

```
Run the shell:
    bash
    ./shell
```

Type a command:

bash
What do you want? > listdir -l

```
Exit the shell:
    bash
    What do you want? > exit
```



```
What do you want? > exit
Exiting shell. Goodbye!
neha@penguin:~/programming/project (1)$
```

Notes:

1.All commands must be placed in the ./bin/ directory.
2.Ensure the executables have the correct permissions:

```
bash
chmod +x ./bin/*
```

3.The runtime command can be used to measure the execution time of other commands.

Issues or Debugging

If a command does not execute:

1. Verify it exists in the ./bin/ directory.
2. Ensure it has execution permissions.
3. check if the path is correctly set in the shell.

This README provides an overview of how to use and run the Linux commands implemented in C.