

Documentation

-CED18I051

Introduction

There are 4 programs in the src folder. The folder tree is as follows:

```
.
├── Makefile
├── progpreassignment.pdf
├── Documentation.pdf
└── src
    ├── cp.c
    ├── multisort.cpp
    ├── mysort.c
    └── smartsort.cpp
```

1 directory, 6 files

cp.c: This program refers to Q1. This program is used to copy and remove files. Soft Links were used to have both the capabilities in one source code.

mysort.c: This program refers to Q2.

multisort.cpp: This program refers to Q3.

smartsort.cpp: This program refers to Q4.

Compile

Run the following command on the terminal:

```
make
```

After running the above command, the folder tree should look as follows:

```
.
├── cp
├── Makefile
├── multisort
├── mysort
├── progpreassignment.pdf
├── Documentation.pdf
├── rm
├── smartsort
└── src
    ├── cp.c
    ├── multisort.cpp
    ├── mysort.c
    └── smartsort.cpp
```

1 directory, 11 files

Running

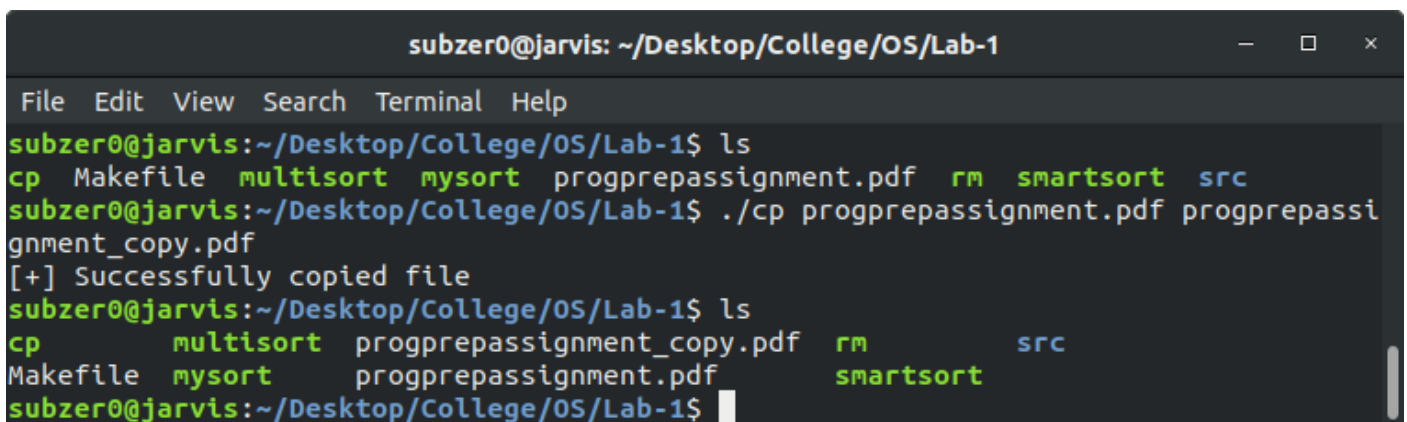
Question 1:

./cp (Copy)

Syntax:

```
./cp <path-to-source-file> <path-to-source-file>
```

Since the program opens the files in binary mode, any type of file can be copied.

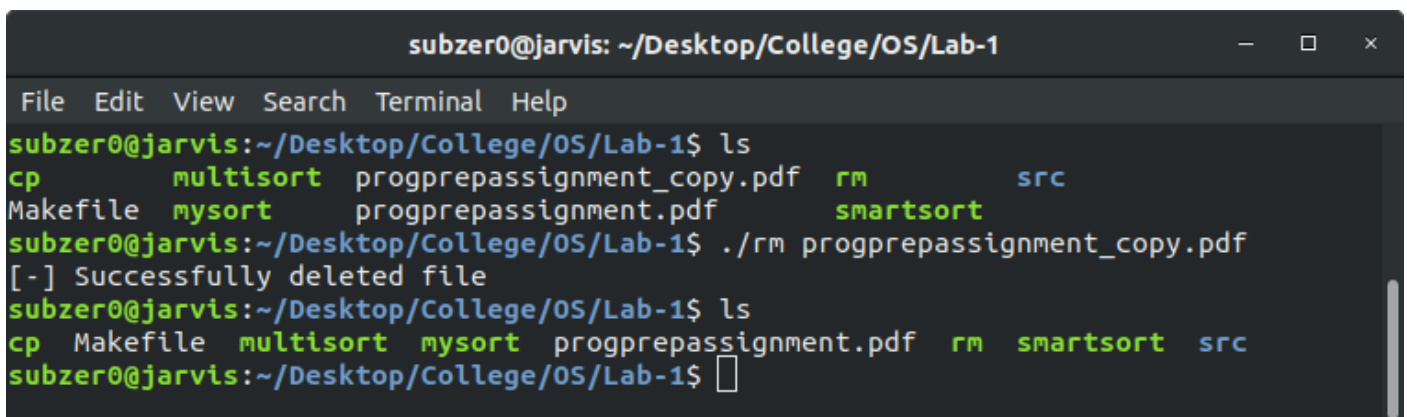


```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ls
cp Makefile multisort mysort progpreassignment.pdf rm smartsort src
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./cp progpreassignment.pdf progpreassi
gnment_copy.pdf
[+] Successfully copied file
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ls
cp          multisort  progpreassignment_copy.pdf  rm          src
Makefile    mysort      progpreassignment.pdf      smartsort
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

./rm (Delete)

Syntax:

```
./rm <path-to-source-file>
```



```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ls
cp          multisort  progpreassignment_copy.pdf  rm          src
Makefile    mysort      progpreassignment.pdf      smartsort
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./rm progpreassignment_copy.pdf
[-] Successfully deleted file
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ls
cp Makefile multisort mysort progpreassignment.pdf rm smartsort src
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

Since copy and delete are file operations, both the source codes were combined in the same source code named cp.c.

It can be observed from the above screenshots that ./cp and ./rm perform what they are meant to. *Soft links* were used to link ./rm to ./cp.

Assert statements were used to avoid the program from facing a segmentation fault runtime error.

The following are some of the validation errors outputted by the code:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./cp
[!] Invalid arguments.
[!] Correct usage: ./cp <path-to-source-file> <path-to-source-file>
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./cp test.txt
[!] Invalid arguments.
[!] Correct usage: ./cp <path-to-source-file> <path-to-source-file>
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./cp test.txt tes1a.t
cp: src/cp.c:21: cp: Assertion `src = fopen(sourcefile, "rb")' failed.
Aborted (core dumped)
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./rm
[!] Invalid arguments.
[!] Correct usage: ./rm <path-to-source-file>
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./rm test.txt
rm: src/cp.c:46: rm: Assertion `src = fopen(sourcefile, "rb")' failed.
Aborted (core dumped)
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

Question 2:

./mysort

Syntax:

```
./mysort <n> <q> <arr>
where
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
```

Function pointers were used to implement the sorting functions.

The output for a valid input is as follows:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./mysort 5 1 -10 2 0 2 9
-----Sorted in Ascending Order-----
-10 0 2 2 9

subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./mysort 5 0 -10 2 0 2 9
-----Sorted in Descending Order-----
9 2 2 0 -10

subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

The following are some of the validation errors outputted by the code:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./mysort
[!] Invalid Input:
Correct usage: ./mysort <n> <q> <arr>
where
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./mysort 5 1 2 4 1
[!] Invalid Input:
Correct usage: ./mysort <n> <q> <arr>
where
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./mysort 3 1 1 2 3 4 5
[!] Invalid Input:
Correct usage: ./mysort <n> <q> <arr>
where
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

Question 3:

./multisort

Syntax:

```
./mysort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
```

This program uses function overloading to sort arrays of different data types.

The output for some valid inputs is as follows:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort c 5 1 q w e r t
-----Sorted in Ascending Order-----
e q r t w

subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort c 5 0 q w e r t
-----Sorted in Descending Order-----
w t r q e

subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort i 5 1 3 1 4 2 -5
-----Sorted in Ascending Order-----
-5 1 2 3 4

subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort i 5 0 3 1 4 2 -5
-----Sorted in Descending Order-----
4 3 2 1 -5

subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort f 5 1 -10 -10.1 -10.01 0 11
-----Sorted in Ascending Order-----
-10.000000 -10.100000 -10.010000 0.000000 11.000000

subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort f 5 0 -10 -10.1 -10.01 0 11
-----Sorted in Descending Order-----
11.000000 0.000000 -10.000000 -10.100000 -10.010000

subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

The following are some of the validation errors outputted by the code:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort
[!] Invalid Input:
Correct usage: ./mysort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort r 3 1 1 2 3
[!] Invalid Input:
Correct usage: ./mysort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./multisort c q e r
[!] Invalid Input:
Correct usage: ./mysort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

Question 4:

./smartsort

This program uses function templates to sort arrays of different data types.

Syntax:

```
./smartsort <t> <n> <q> <arr>  
where  
t : type of array : c - character, i - integer, f - float  
n : number of elements in array  
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order  
arr : n space separated integers
```

The output for some valid inputs is as follows:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1  
File Edit View Search Terminal Help  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort c 5 1 p u s a z  
-----Sorted in Ascending Order-----  
a p s u z  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort c 5 0 p u s a z  
-----Sorted in Descending Order-----  
z u s p a  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1  
File Edit View Search Terminal Help  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort f 5 1 -1.1 2 -3.5 4 -5  
-----Sorted in Ascending Order-----  
-5 -3.5 -1.1 2 4  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort f 5 0 -1.1 2 -3.5 4 -5  
-----Sorted in Descending Order-----  
4 2 -1.1 -3.5 -5  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1  
File Edit View Search Terminal Help  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort i 5 1 -1 2 -3 4 -5  
-----Sorted in Ascending Order-----  
-5 -3 -1 2 4  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort i 5 0 -1 2 -3 4 -5  
-----Sorted in Descending Order-----  
4 2 -1 -3 -5  
  
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```

The following are some of the validation errors outputted by the code:

```
subzer0@jarvis: ~/Desktop/College/OS/Lab-1
File Edit View Search Terminal Help
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort
[!] Invalid Input:
Correct usage: ./smartsort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space seperated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort q 2 1 a s
[!] Invalid Input:
Correct usage: ./smartsort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space seperated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$ ./smartsort c 3 1 a d f s
[!] Invalid Input:
Correct usage: ./smartsort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space seperated integers
subzer0@jarvis:~/Desktop/College/OS/Lab-1$
```