B.Siva Shirish -192324016

2. Identify the system calls to copy the content of one file to another and illustrate the same using a C program.

Aim:

To copy the content of one file to another using system calls in a C program.

Algorithm:

1. Start the program.
2. Open the source file in read-only mode using the open() system call.
3. Open or create the destination file in write mode using the open() system call.
4. Read the content from the source file in chunks using the read() system call.
5. Write the content to the destination file using the write() system call.
6. Close both files using the close() system call.
7. End the program.

Procedure:

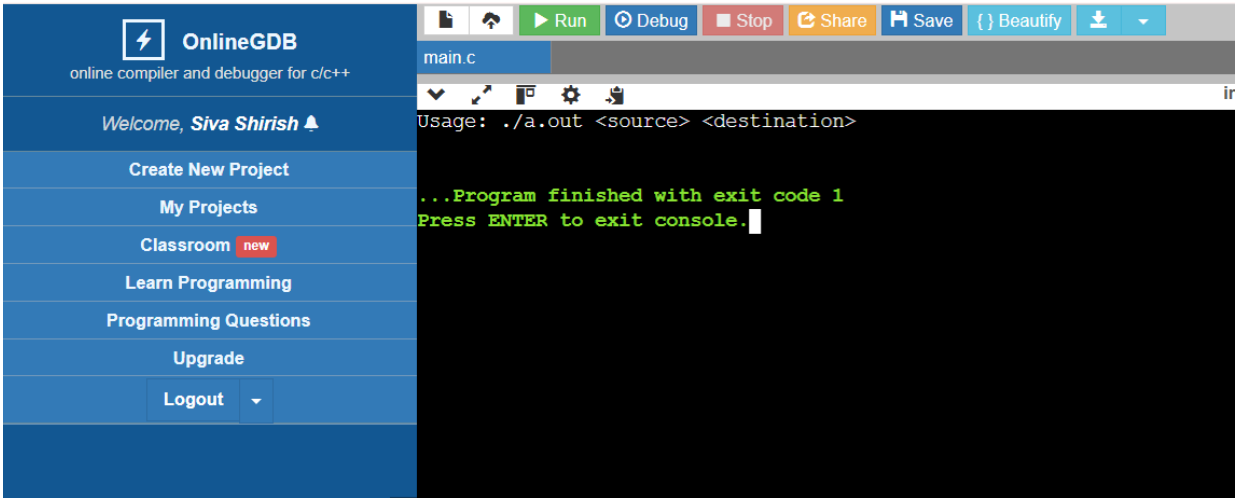
1. Include necessary headers: <fcntl.h>, <unistd.h>, <stdio.h>.
2. Use open() to access the source and destination files.
3. Use read() and write() in a loop to transfer data.
4. Handle errors appropriately (e.g., file not found).
5. Use close() to release file descriptors after the operation.

CODE:

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

#define BUF_SIZE 1024

int main(int argc, char *argv[]) {
   int src_fd, dest_fd, bytes_read, bytes_written;
   char buffer[BUF_SIZE];

   if (argc != 3) {
      perror("Usage: ./copyfile <source_file> <destination_file>");
      exit(EXIT_FAILURE);
   }
```

```
    src_fd = open(argv[1], O_RDONLY);
    if (src_fd == -1) {
        perror("Error opening source file");
        exit(EXIT_FAILURE);
    }


    dest_fd = open(argv[2], O_WRONLY | O_CREAT | O_TRUNC, 0644); //
0644: rw-r--r--
    if (dest_fd == -1) {
        perror("Error opening destination file");
        close(src_fd); // Close source file before exiting
        exit(EXIT_FAILURE);
    }

    while ((bytes_read = read(src_fd, buffer, BUF_SIZE)) > 0) {
        bytes_written = write(dest_fd, buffer, bytes_read);
        if (bytes_written != bytes_read) {
            perror("Error writing to destination file");
            close(src_fd);
            close(dest_fd);
            exit(EXIT_FAILURE);
        }
    }
    if (bytes_read == -1) {
        perror("Error reading from source file");
    }

    close(src_fd);
    close(dest_fd);

    printf("File copied successfully from '%s' to '%s'\n", argv[1], argv[2]);

    return 0;
}
```

OUTPUT:


OnlineGDB
online compiler and debugger for c/c++
Welcome, Siva Shirish
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Upgrade
Logout
Run
Debug
Stop
Share
Save
{ } Beautify
main.c
Usage: ./a.out <source> <destination>
...Program finished with exit code 1
Press ENTER to exit console.