A.HARSHA MUKUNDHA - 192324070

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

#define BUFFER_SIZE 1024

int main() {
    int src_fd, dest_fd, n;
    char buffer[BUFFER_SIZE];
```

```
src_fd = open("source.txt", O_RDONLY);
if (src_fd < 0) {
    perror("Error opening source file");
    return 1;
}

dest_fd = open("destination.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);
if (dest_fd < 0) {
    perror("Error opening destination file");
    close(src_fd);
    return 1;
}

while ((n = read(src_fd, buffer, BUFFER_SIZE)) > 0) {
    if (write(dest_fd, buffer, n) != n) {
        perror("Error writing to destination file");
        close(src_fd);
        close(dest_fd);
        return 1;
    }
}

if (n < 0) {
    perror("Error reading from source file");
}

close(src_fd);
close(dest_fd);
```

```
    return 0;
}
```

OUTPUT: