main.c

Share Run

```c
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    pid_t pid, ppid;

    pid = fork();

    if (pid < 0) {
        perror("Fork failed");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {

        ppid = getppid();
        printf("Child Process ID: %d\n", getpid());
        printf("Parent Process ID: %d\n", ppid);
    } else {

        printf("Parent Process ID: %d\n", getpid());
        printf("Created Child Process ID: %d\n", pid);
    }

    return 0;
}
```

Output

```
Parent Process ID: 9210
Created Child Process ID: 9211
Child Process ID: 9211
Parent Process ID: 9210


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#define BUFFER_SIZE 1024
int main(int argc, char *argv[]) {
    int source, destination, bytesRead;
    char buffer[BUFFER_SIZE];
    if (argc != 3) {
        write(2, "Usage: ./filecopy <source> <destination>\n", 42);
        exit(1);
    }
    source = open(argv[1], O_RDONLY);
    if (source < 0) {
        perror("Error opening source file");
        exit(1);
    }
    destination = open(argv[2], O_WRONLY | O_CREAT | O_TRUNC, 0644);
    if (destination < 0) {
        perror("Error opening destination file");
        close(source);
        exit(1);
    }
    while ((bytesRead = read(source, buffer, BUFFER_SIZE)) > 0) {
        write(destination, buffer, bytesRead);
    }
    close(source);
```

Output

```
Usage: ./filecopy <source> <destination>

=== Code Exited With Errors ===
```