Ex. No:	**4**	**Illustrate the inter process communication strategy**
Date:		

AIM:

To implement **Inter-Process Communication (IPC)** in C programming by using different methods such as **Pipes** in order to enable data exchange and synchronization between multiple processes. The objective is to understand how processes coordinate and communicate effectively in an operating system environment.

ALGORITHM:

Step 1: Start the program.

Step 2: Create a pipe using the pipe(fd) system call.

Step 3: Create a new process using the fork() system call.

Step 4: If fork() fails, display an error and terminate.

Step 5: If it is the parent process:

Close the read end of the pipe.

Write a message into the pipe using write().

Close the write end.

Step 6: If it is the child process:

Close the write end of the pipe.

Read the message from the pipe using read().

Display the received message.

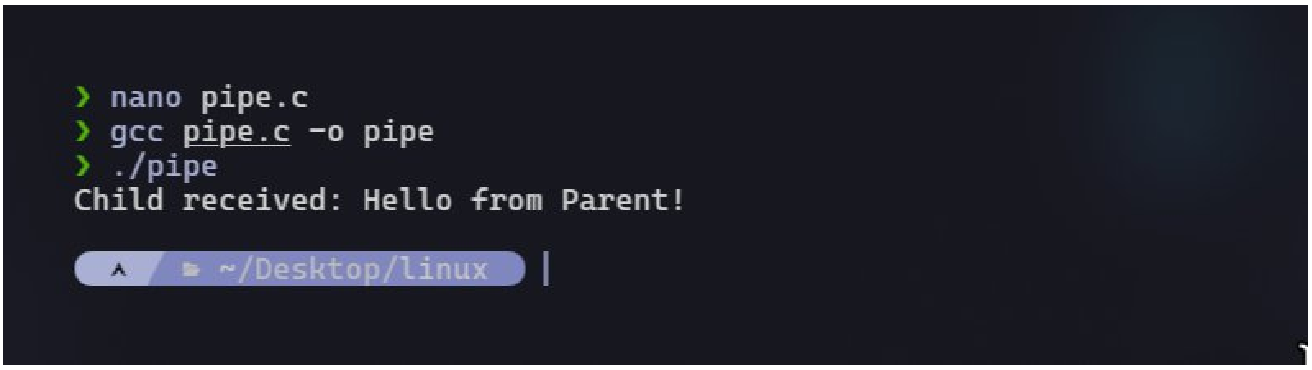
Close the read end.

Step 7: Stop the program.

PROGRAM:

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>

int main() {
    int fd[2]; // fd[0] = read end, fd[1] = write end
    pid_t pid;
    char write_msg[] = "Hello from Parent!";
    char read_msg[50];

    // Create a pipe
    if (pipe(fd) == -1)
        { printf("Pipe failed!\n");
        return 1;
    }

    // Fork a child process
    pid = fork();

    if (pid < 0) {
        printf("Fork failed!\n");
        return 1;
    }

    if (pid > 0) {
        // Parent process
        close(fd[0]); // Close unused read end
        write(fd[1], write_msg, strlen(write_msg) + 1);
        close(fd[1]); // Close write end after writing
    } else {
        // Child process
        close(fd[1]); // Close unused write end
        read(fd[0], read_msg, sizeof(read_msg));
        printf("Child received: %s\n", read_msg);
        close(fd[0]);
    }

    return 0;
}
```

OUTPUT:



```
❯ nano pipe.c
❯ gcc pipe.c -o pipe
❯ ./pipe
Child received: Hello from Parent!

 ~/Desktop/linux  |
```

RUBRICS FOR EVALUATION	MAXIMUM	AWARDED
Fundamental Knowledge	**2**	
Design of Experiment	**2**	
Implementation	**4**	
Viva	**2**	
Total	**10**	
Signature		

RESULT:

Thus, the Inter-Process Communication strategies using **Pipes** were implemented successfully in C. The execution verified that data could be transferred correctly between parent and child processes, demonstrating the working principles of IPC.The experiment also highlighted how different IPC mechanisms provide process synchronization, resource sharing, and reliable communication in operating systems.