

## 9.INTER PROCESS COMMUNICATION

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
int main() {
    int fd[2];
    pid_t pid;
    char write_msg[100], read_msg[100];
    if (pipe(fd) == -1) {
        printf("Pipe failed!\n");
        return 1;
    }
    pid = fork();
    if (pid < 0) {
        printf("Fork failed!\n");
        return 1;
    }
    else if (pid == 0) { // Child
        close(fd[1]);
        read(fd[0], read_msg, sizeof(read_msg));
        printf("Child received: %s\n", read_msg);
        close(fd[0]);
    }
    else {
        close(fd[0]);
        printf("Enter message to send: ");
        fgets(write_msg, sizeof(write_msg), stdin);
        write(fd[1], write_msg, strlen(write_msg) + 1);
    }
}
```

```
        close(fd[1]);  
    }  
    return 0;  
}
```