4.develop a c program which demonstrates interprocess communication between a reader process and a writer process . use mkfifo ,open,read,write and close AIP's in your program

```c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>

#define FIFO_PATH "myfifo"

void writer_process() {
    int fd;
    char message[] = "Hello, reader!";

    // Create the FIFO (named pipe)
    mkfifo(FIFO_PATH, 0666);

    // Open the FIFO for writing
    fd = open(FIFO_PATH, O_WRONLY);

    // Write message to the FIFO
    write(fd, message, sizeof(message));

    // Close the FIFO
    close(fd);
}

void reader_process() {
    int fd;
    char buffer[100];

    // Open the FIFO for reading
    fd = open(FIFO_PATH, O_RDONLY);

    // Read from the FIFO
    read(fd, buffer, sizeof(buffer));

    // Display the message received from the writer
    printf("Reader received: %s\n", buffer);

    // Close the FIFO
    close(fd);
}

int main() {
```

```
    pid_t pid;

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

    if (pid < 0) {
        perror("Fork failed");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        // Child process (Writer)
        writer_process();
    } else {
        // Parent process (Reader)
        reader_process();
    }

    return 0;
}
```

In this program:

- The `writer_process` function creates a named pipe (FIFO) using `mkfifo`, opens it for writing using `open`, writes a message to it using `write`, and then closes the pipe using `close`.
- The `reader_process` function opens the same named pipe for reading using `open`, reads from it using `read`, displays the received message, and then closes the pipe using `close`.
- The main function forks a child process, and the parent becomes the reader while the child becomes the writer.

Note: The named pipe (`myfifo` in this example) must be created before running the program. You can create it using the following command in the terminal:

mkfifo myfifo

This command creates a named pipe with the specified path, and the reader and writer processes will use this path to access the same named pipe for communication.