```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#define FIFO1 "/tmp/fifo1"
#define FIFO2 "/tmp/fifo2"
#define BUFFER_SIZE 1024
// Function to count characters, words, and lines
void count_char_word_line(char *input, int *char_count, int *word_count, int *line_count)
{
    *char_count = *word_count = *line_count = 0;
    int in_word = 0;
    for (int i = 0; input[i] != '\0'; i++)
    {
        (*char_count)++; // Count characters
        if (input[i] == '\n')
            (*line_count)++; // Count lines
        if (input[i] == ' ' || input[i] == '\n')
        {
            in_word = 0;
        }
        else if (in_word == 0)
        {
            in_word = 1;
            (*word_count)++; // Count words
        }
    }
}
// Process 1: Sends data, receives result
void process1()
{
    char buffer[BUFFER_SIZE];
    // Open FIFO1 for writing
    int fd_write = open(FIFO1, O_WRONLY);
    printf("Enter sentences (Ctrl+D to end):\n");
    // Read from stdin and send to Process 2
    while (fgets(buffer, BUFFER_SIZE, stdin) != NULL)
    {
        write(fd_write, buffer, strlen(buffer) + 1);
    }
    close(fd_write);
    // Open FIFO2 for reading result from Process 2
    int fd_read = open(FIFO2, O_RDONLY);
    while (read(fd_read, buffer, sizeof(buffer)) > 0)
    {
        printf("Received: %s\n", buffer);
    }
    close(fd_read);
}
// Process 2: Receives data, processes it, sends result
void process2()
{
    char buffer[BUFFER_SIZE];
    int char_count, word_count, line_count;
    // Open FIFO1 for reading
    int fd read = open(FIFO1, O RDONLY);
```

```
    // Read from FIFO1 and process it
    while (read(fd_read, buffer, sizeof(buffer)) > 0)
    {
        count_char_word_line(buffer, &char_count, &word_count, &line_count);
        sprintf(buffer, "Chars: %d, Words: %d, Lines: %d", char_count, word_count,
                line_count);
        // Open FIFO2 for writing result back
        int fd_write = open(FIFO2, O_WRONLY);
        write(fd_write, buffer, strlen(buffer) + 1);
        close(fd_write);
    }
    close(fd_read);
}
int main()
{
    // Create two FIFOs
    mkfifo(FIFO1, 0666);
    mkfifo(FIFO2, 0666);
    int choice;
    printf("Enter 1 for Process 1, 2 for Process 2: ");
    scanf("%d", &choice);
    getchar(); // Consume newline
    if (choice == 1)
    {
        process1();
    }
    else if (choice == 2)
    {
        process2();
    }
    // Clean up FIFOs
    unlink(FIFO1);
    unlink(FIFO2);
    return 0;
}
```