```c
//Process.c
#include <stdio.h>
 #include <stdlib.h>
 #include <fcntl.h>
 #include <unistd.h>
 #include <string.h>
 #include <sys/types.h>
 #include <sys/stat.h>
 #include <ctype.h>  // For isspace()
 #define FIFO1 "fifo1"
 #define FIFO2 "fifo2"
 #define BUFFER_SIZE 1024
 #define OUTPUT_FILE "output.txt"

void count_stats(const char *str, int *chars, int *words, int *lines) {
    *chars = strlen(str);
    *words = 0;
    *lines = 0;

    int in_word = 0;
    for (int i = 0; str[i]; i++) {
        if (str[i] == '\n')
            (*lines)++;

        if (isspace((unsigned char)str[i])) {
            if (in_word) {
                (*words)++;
                in_word = 0;
            }
        } else {
            in_word = 1;
        }
    }
    if (in_word)
        (*words)++;

    // If no newline found, at least one line
    if (*lines == 0)
        *lines = 1;
```

```
}

int main() {
    int fd1, fd2;
    char buffer[BUFFER_SIZE];

    // Create FIFOs if not already existing
    mkfifo(FIFO1, 0666);
    mkfifo(FIFO2, 0666);
    printf("Process started. Waiting for input via %s...\n", FIFO1);

    // Open FIFO1 for reading
    fd1 = open(FIFO1, O_RDONLY);
    if (fd1 < 0) {
        perror("open fifo1");
        exit(EXIT_FAILURE);
    }
    // Open FIFO2 for writing
    fd2 = open(FIFO2, O_WRONLY);
    if (fd2 < 0) {
         perror("open fifo2");
        close(fd1);
        exit(EXIT_FAILURE);
    }

    while (1) {
        int n = read(fd1, buffer, BUFFER_SIZE - 1);
        if (n <= 0)
            break;

        buffer[n] = '\0';

        if (strcmp(buffer, "exit") == 0)
            break;

        int chars, words, lines;
        count_stats(buffer, &chars, &words, &lines);

        // Write stats to file
```

```
        FILE *fp = fopen(OUTPUT_FILE, "w");
        if (!fp) {
            perror("fopen output.txt");
            break;
        }
        fprintf(fp, "Characters: %d\nWords: %d\nLines: %d\n", chars, words, lines);
        fclose(fp);

        // Prepare response to send via fifo2
        snprintf(buffer, sizeof(buffer),
                 "Characters: %d\nWords: %d\nLines: %d\n", chars, words, lines);
        if (write(fd2, buffer, strlen(buffer)) < 0) {
            perror("write fifo2");
            break;
        }
    }
    close(fd1);
    close(fd2);
    printf("Process terminated.\n");
    return 0;
}

//sender_receiver.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#define FIFO1 "fifo1"
#define FIFO2 "fifo2"
#define BUFFER_SIZE 1024

int main() {
    int fd1, fd2;
    char buffer[BUFFER_SIZE];

    // Create FIFO files if not already existing
```

```
mkfifo(FIFO1, 0666);
mkfifo(FIFO2, 0666);

// Open FIFO1 for writing (sending)
fd1 = open(FIFO1, O_WRONLY);
if (fd1 < 0) {
    perror("open fifo1");
    exit(EXIT_FAILURE);
}

// Open FIFO2 for reading (receiving)
fd2 = open(FIFO2, O_RDONLY);
if (fd2 < 0) {
    perror("open fifo2");
    close(fd1);
    exit(EXIT_FAILURE);
}
printf("Enter sentences (type 'exit' to quit):\n");

while (1) {
    printf("> ");
    fflush(stdout);

    if (fgets(buffer, BUFFER_SIZE, stdin) == NULL) {
        perror("fgets");
        break;
    }
    buffer[strcspn(buffer, "\n")] = '\0'; // remove newline
    if (write(fd1, buffer, strlen(buffer) + 1) < 0) {
        perror("write fifo1");
        break;
    }

    if (strcmp(buffer, "exit") == 0)
        break;
    int n = read(fd2, buffer, BUFFER_SIZE - 1);
    if (n < 0) {
        perror("read fifo2");
        break;
```

```
        }
        buffer[n] = '\0';
    printf("\nProcessed Output:\n%s\n\n", buffer);
    }
    close(fd1);
    close(fd2);
    printf("Sender/Receiver terminated.\n");
    return 0;
    }
```

OUTPUT:

Terminal 1 (Processor)
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ gcc processor.c -o processor
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ ./processor
Process started. Waiting for input via fifo1...

Terminal 2 (Sender/Receiver)
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ gcc sender.c -o sender
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ ./sender
Enter sentences (type 'exit' to quit):
> Hello World from FIFO

Processed Output:
Characters: 20
Words: 4
Lines: 1

> Adaptive Traffic Signal Control using DRL

Processed Output:
Characters: 41
Words: 6
Lines: 1

> exit
Sender/Receiver terminated.

Content of output.txt after last input:
Characters: 41
Words: 6
Lines: 1