Process 1:Counter

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

#define FIFO1 "/tmp/fifo1"
#define FIFO2 "/tmp/fifo2"
#define OUTPUT_FILE "output.txt"

int main() {
    char buffer[256];
    int fd;
    FILE *file;

    while (1) {
        // Read from FIFO1
        fd = open(FIFO1, O_RDONLY);
        read(fd, buffer, sizeof(buffer));
        close(fd);

        if (strncmp(buffer, "exit", 4) == 0) {
            break;
        }

        // Count characters, words, and lines
        int chars = strlen(buffer);
        int words = 0, lines = 0;

        for (char *p = buffer; *p; p++) {
            if (*p == ' ' || *p == '\n') {
                words++;
            }
            if (*p == '\n') {
                lines++;
            }
        }
        words++; // To account for the last word if there is any

        // Write to output file
        file = fopen(OUTPUT_FILE, "a");
        fprintf(file, "Characters: %d, Words: %d, Lines: %d\n", chars, words, lines);
        fclose(file);

        // Prepare message to send back
        snprintf(buffer, sizeof(buffer), "Characters: %d, Words: %d, Lines: %d\n", chars, words, lines);

        // Write to FIFO2
        fd = open(FIFO2, O_WRONLY);
        write(fd, buffer, sizeof(buffer));
```

```
        close(fd);
    }

    return 0;
}
```

Process 2:Writer

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

#define FIFO1 "/tmp/fifo1"
#define FIFO2 "/tmp/fifo2"

int main() {
    char buffer[256];
    int fd;

    // Create FIFOs
    mkfifo(FIFO1, 0666);
    mkfifo(FIFO2, 0666);

    while (1) {
        printf("Enter a sentence (type 'exit' to quit): ");
        fgets(buffer, sizeof(buffer), stdin);

        if (strncmp(buffer, "exit", 4) == 0) {
            break;
        }

        // Write to FIFO1
        fd = open(FIFO1, O_WRONLY);
        write(fd, buffer, sizeof(buffer));
        close(fd);

        // Read from FIFO2
        fd = open(FIFO2, O_RDONLY);
        read(fd, buffer, sizeof(buffer));
        printf("Received from Process 2: %s", buffer);
        close(fd);
    }

    return 0;
}
```

Terminal 1:

```
meit@meit-OptiPlex-3046:~/33265/7a$ mkfifo /tmp/fifo1
meit@meit-OptiPlex-3046:~/33265/7a$ mkfifo /tmp/fifo2
meit@meit-OptiPlex-3046:~/33265/7a$ gedit writer.c
^C
meit@meit-OptiPlex-3046:~/33265/7a$ gedit counter.c
^C
meit@meit-OptiPlex-3046:~/33265/7a$ gcc writer.c -o writer
meit@meit-OptiPlex-3046:~/33265/7a$ gcc counter.c -o counter
meit@meit-OptiPlex-3046:~/33265/7a$ ./counter
```

Terminal 2:

```
meit@meit-OptiPlex-3046:~/33265/7a$ ./writer
Enter a sentence (type ’exit’ to quit): Hello
Received from Process 2: Characters: 6, Words: 2, Lines: 1
Enter a sentence (type ’exit’ to quit): ^C
meit@meit-OptiPlex-3046:~/33265/7a$ ./writer
Enter a sentence (type ’exit’ to quit): Hello World
Received from Process 2: Characters: 6, Words: 2, Lines: 1
Characters: 13, Words: 4, Lines: 1
Enter a sentence (type ’exit’ to quit): Samali
Received from Process 2: Characters: 6, Words: 2, Lines: 1
Characters: 13, Words: 4, Lines: 1
Characters: 7, Words: 2, Lines: 1
Enter a sentence (type ’exit’ to quit): exit
meit@meit-OptiPlex-3046:~/33265/7a$ ./writer
Enter a sentence (type ’exit’ to quit): samali
Received from Process 2: Characters: 7, Words: 2, Lines: 1
Enter a sentence (type ’exit’ to quit): ^C
```

Output.txt (Output File)

```
Characters: 6, Words: 2, Lines: 1
Characters: 13, Words: 4, Lines: 1
Characters: 7, Words: 2, Lines: 1
Characters: 7, Words: 2, Lines: 1
```

Terminal 1: (last step)

```
meit@meit-OptiPlex-3046:~/33265/7a$ rm /tmp/fifo1 /tmp/fifo2
```