

Assignment No. 7: Inter process communication in Linux using following.

a. FIFOs: Full duplex communication between two independent processes. First process accepts sentences and writes on one pipe to be read by second process and second process counts number of characters, number of words and number of lines in accepted sentences, writes this output in a text file and writes the contents of the file on second pipe to be read by first process and displays on standard output.

b. Inter-process Communication using Shared Memory using System V. Application to demonstrate: Client and Server Programs in which server process creates a shared memory segment and writes the message to the shared memory segment. Client process reads the message from the shared memory segment and displays it to the screen.

SMwriter.cpp file content:

```
#include <iostream>
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
using namespace std;

int main()
{
    // ftok to generate unique key
    key_t key = ftok("shmfile", 65);

    // shmget returns an identifier in shmid
    int shmid = shmget(key, 1024, 0666 | IPC_CREAT);

    // shmat to attach to shared memory
    char* str = (char*)shmat(shmid, (void*)0, 0);

    cout << "Write Data : ";
    cin.getline(str, 1024);

    cout << "Data written in memory:\n " << str << endl;

    // detach from shared memory
    shmdt(str);

    return 0;
}
```

SMreader.cpp file content:

```
#include <iostream>
#include <stdio.h>
```

```

#include
<sys/ipc.h>
#include
<sys/shm.h
> using
namespace
std;

int main()
{
    // ftok to generate unique
    key key_t key =
    ftok("shmfile", 65);

    // shmget returns an identifier in shmid
    int shmid = shmget(key, 1024, 0666 | IPC_CREAT);

    // shmat to attach to shared memory
    char* str = (char*)shmat(shmid, (void*)0, 0);

    cout << "Data read from memory:\n"
    << str; cout << "\n";
    // detach from shared memory
    shmdt(str);

    // destroy the shared memory
    shmctl(shmid, IPC_RMID,
    NULL);

    return 0;
}

```

OUTPUT:

