# Assignment 3

**Description:** This assignment covers the usage of basic IPC(Inter Process Communication) using **pipe()** and named pipe or FIFO i.e. **mknod()** system calls.

## Problem for the Lab

**Assignment no 3(a) :**

**Write a C program to handle the reception of INTERRUPT FROM THE KEYBOARD signal by executing a particular (user) function, which function is responsible for creating a child process by using fork() system call. Also, the child process will take an input N (N is the no. of terms to generate the Fibonacci series) from the user and generate the Fibonacci series up to N terms. Then the child process will send the Fibonacci series terms one by one to the parent process by using pipe() system call and the parent process will read the Fibonacci series terms one by one and will display it.**

**(N.B. – Before sending the Fibonacci terms one by one from the child process, first you send the Process ID of the child process to the parent process and read and display it in the parent process. Again after sending the terms one by one from the child process to the parent process, from the child process you send the Signal ID for which handler function has been generated and read and display it in the parent process.)**

**Hints:-**

- Learn about signal () system call, pipe () system call, fork () system call, write () system call, read () system call.
- To keep synchronize the child process execution with parent process execution and vice versa, you may need to use the sleep () system call.
- For the above mentioned system call, follow the man pages.
- Try to understand the following diagram (Diagram1).

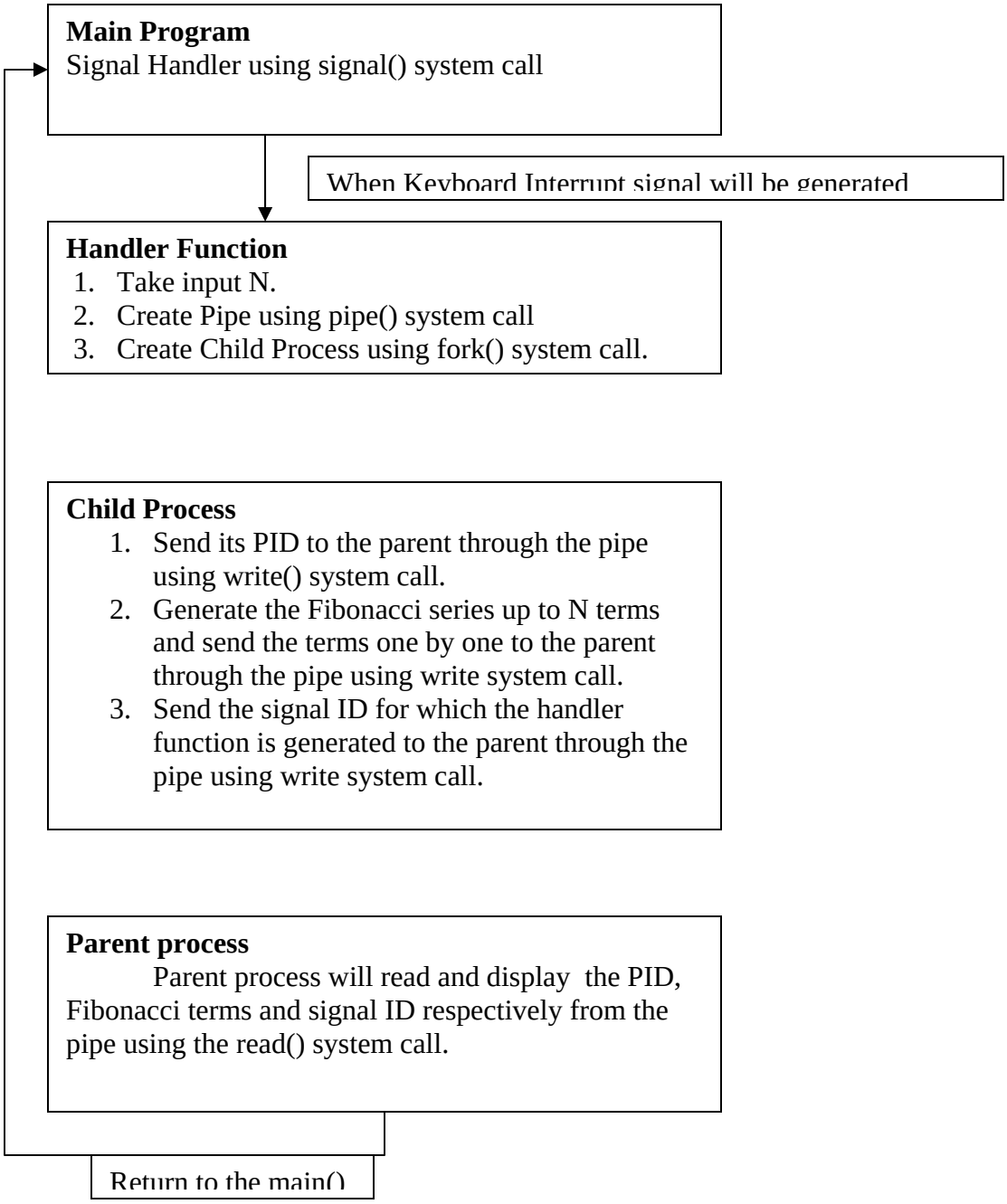


**Figure1:- Diagram1**

**Assignment no 3(b) :**

**Write a C program called Program1 which is responsible for creating a named PIPE or FIFO with necessary permission for all users by using the mknod( ) system call. Now, you keep the facility to read the information from the FIFO if any other process write some information into the FIFO and as well as keep the facility to write some information into the FIFO, so that Program2 can read it.**

**Again write another C program called Program2 which is responsible for taking the user input and write it back to the FIFO so that Program1 can read it. Moreover, you have to provide the facility to read the FIFO in this program.**

---

**(N.B. - First run the Program1 and keep it in running state in the background. Then run the Program2 which will take user input string and write the input string into the FIFO. Then Program1 will read it from the FIFO and display it and after that Program1 will write something into the FIFO as ACKNOWLEDGEMENT and program2 will read the ACKNOWLEDGEMENT string from the FIFO. In Program2, you may take the input string as a command line argument)**

**Hints:**

- Learn about mknod() system call, write () system call, read () system call.
- To keep synchronize the Program1 execution with Program2 execution and vice versa, you may need to use the sleep () system call.
- For the above mentioned system call, follow the man pages.