**Name: Shivam Tiwari**

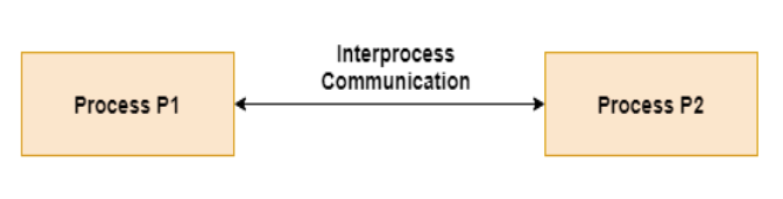
**Roll No: 5117060**

**EXPERIMENT 04**

**Aim:** Implementation of Inter Process Communication.

**Theory:**

Interprocess communication is the mechanism provided by the operating system that allows processes to communicate with each other. This communication could involve a process letting another process know that some event has occurred or the transferring of data from one process to another.

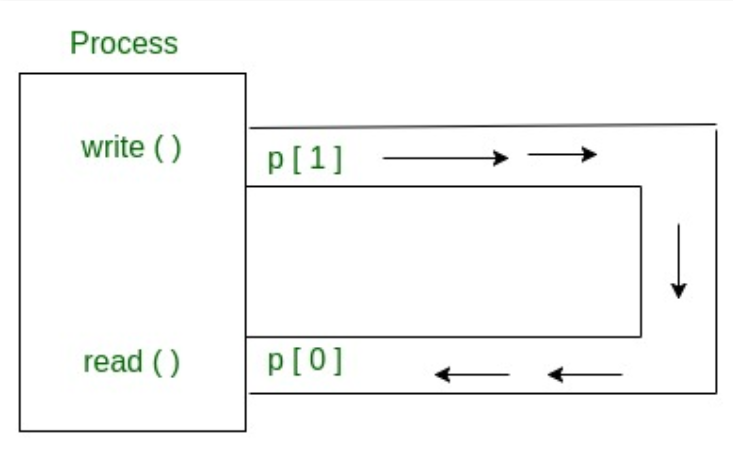


**Fig. Interprocess communication**

For implementing IPC, we have used the concept of pipe in C programming

Conceptually, a pipe is a connection between two processes, such that the standard output from one process becomes the standard input of the other process. In UNIX Operating System, Pipes are useful for communication between related processes (inter-process communication).

* Pipe is one-way communication only i.e. we can use a pipe such that One process write to the pipe, and the other process reads from the pipe. It opens a pipe, which is an area of main memory that is treated as a “virtual file”.
* The pipe can be used by the creating process, as well as all its child processes, for reading and writing. One process can write to this “virtual file” or pipe and another related process can read from it.
* If a process tries to read before something is written to the pipe, the process is suspended until something is written.
* The pipe system call finds the first two available positions in the process’s open file table and allocates them for the read and write ends of the pipe.



**Fig. Pipe in C**

**Program :**

#include<stdio.h>

#include<unistd.h>

int main() {

int pipefds[2];

int returnstatus;

char writemessages[2][20]={"Hi", "Hello"};

char readmessage[20];

returnstatus = pipe(pipefds);

if (returnstatus == -1) {

printf("Unable to create pipe\n");

return 1;

}

printf("Writing to pipe - Message 1 is %s\n", writemessages[0]);

write(pipefds[1], writemessages[0], sizeof(writemessages[0]));

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Reading from pipe – Message 1 is %s\n", readmessage);

printf("Writing to pipe - Message 2 is %s\n", writemessages[0]);

write(pipefds[1], writemessages[1], sizeof(writemessages[0]));

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Reading from pipe – Message 2 is %s\n", readmessage);

return 0;

}

**Output:**

Writing to pipe - Message 1 is Hi

Reading from pipe – Message 1 is Hi

Writing to pipe - Message 2 is Hi

Reading from pipe – Message 2 is Hello

**Conclusion:**

Thus we have successfully implemented a program in C to realize interprocess communication.