**Named Pipe (FIFO):**

1. Pipes were used for related process, But we cant execute the related process (client) from one terminal and (server) from the other.

2. Named pipe are used for communicating between unrelated process.

3. Single named pipe can be used for two way communicaion, as named pipe support bi directional communication.

4. Named pipe are also called as FIFO.

*#include<sys/ypes.h>*

*#include<sys/stat.h>*

*int mkfifo(const char \*pathname, mode\_t mode);*

This library function creates a FIFO special file, which is used as named pipe. The arguments to this function are the file name and the mode.

This system call would return zero on success and -1 in case of failure. Use perror function.

Let us consider a program of running the server on one terminal and running the client on other terminal.

FifoServer.c

#include<stdio.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<fcntl.h>

#include<unistd.h>

#include<string.h>

int main()

{

int fd, fd1;

char readbuf[80];

char end[10];

int to\_end;

int read\_bytes;

//Create fifo

fd1 = mkfifo("server", 0777);

if(fd1 == -1){

perror("mkfifo");

return 1;

}

while(1){

//Open fifo in read only mode

fd = open("server", O\_RDONLY);

//read will return the count of the char read

read\_bytes = read(fd, readbuf, sizeof(readbuf));

//The last index will contain ‘\0’ to denote end of string

readbuf[read\_bytes] = '\0';

printf("Received string: \"%s\" and length is %d\n", readbuf, (int)strlen(readbuf));

//To compare two string

to\_end = strcmp(readbuf, “end”);

if(to\_end == 0){

close(fd);

break;

}

}

return 0;

}

FifoClient.c

#include<stdio.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<fcntl.h>

#include<unistd.h>

#include<string.h>

int main()

{

int fd, fd1;

int end\_process;

int stringlen;

char readbuf[80];

char end\_str[5];

printf("FIFO\_CLIENT: send messages, infinitely, to end enter \"end\"\n");

fd = open("server", O\_CREAT|O\_WRONLY);

strcpy(end\_str, "end");

while(1){

printf("Enter sring: ");

fgets(readbuf, sizeof(readbuf), stdin);

stringlen = strlen(readbuf);

readbuf[stringlen -1 ] = '\0';

end\_process = strcmp(readbuf, end\_str);

//printf

if(end\_process != 0){

write(fd, readbuf, strlen(readbuf));

printf("Sent sring: \"%s\" and string length is %d\n", readbuf, (int)strlen(readbuf));

} else {

write(fd, readbuf, strlen(readbuf));

printf("Sent string: \"%s\" and the string length is %d\n", readbuf, (int)strlen(readbuf));

close(fd);

break;

}

}

return 0;

}

Two way communication:

The pipe communication is ment to be unidirectional. Pipe are restricted to one way communication. Pipe are used for the related process communication.

But for communicating between unrelated process and to execute two process in different terminal we have named pipe or FIFO.