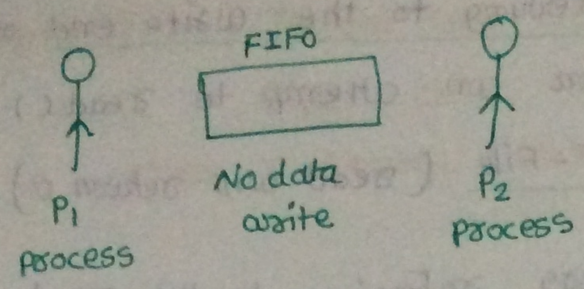


⇒ Pipe is create than no file is Creat so its Called a unnamed pipe, because no involment of Hard-disk.

⇒ Thus the FIFO Special File has no Content on the file system.
(means every time size of FIFO is 0.)

⇒ FIFO create using 2 ways :-

1) MKfifo (command line)
2) using Function.

```
mkfifo fi
ls -l  →  p   (Pipe file)
          0   size of FIFO
```

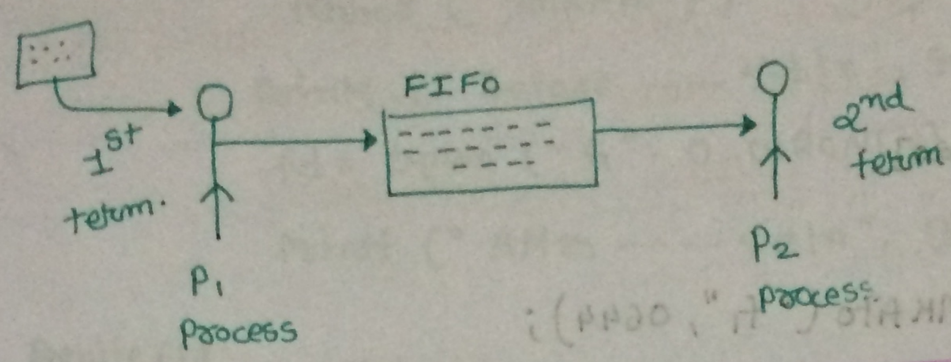
⇒ The File system entry merely server as a reference point so that processes can access the pipe using a name in the file system.

(*) $ man 2 mkfifo

mkfifo → make a FIFO Special file (a named pipe)

```
#include <sys/types.h>
#include <sys/stat.h>

int mkfifo (const char * pathname , mod_t mode) ;
```

Program :- Parent is write the data on the FIFO then child read the data from FIFO.



- ⇒ File is present the mkfifo is Failed.
- ⇒ File is not present than it is Create.
- ⇒ mkfifo is failed then not return it.

Both reading and writing should be alive on. (success).

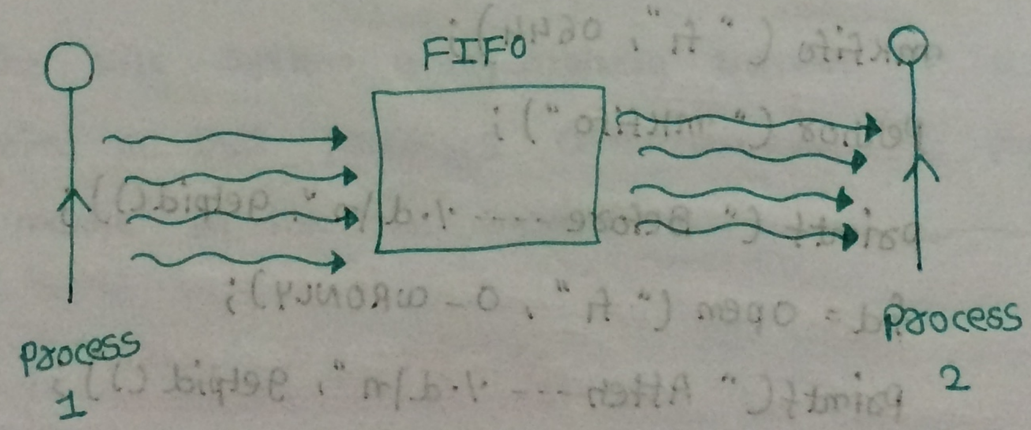
1) vi P1.c

```
#include " header.h "
main()
{
    char a[20];
    int fd;
        mkfifo (" f1", 0644);
        perror (" mkfifo ");
        printf (" Before ---- %d \n", getpid());
        fd = open (" f1", O_WRONLY);
        printf(" After --- %d \n", getpid ());

    Printf (" Enter the data ---- \n");
    scanf (" %s", a);
    write (fd, a, strlen(a)+1);
    Printf (" Done --- \n");
}
```

2) Vi P2.c

```
# include " header.h "
main()
{
        char  a[20] ;
        int fd ;
                mkfifo (" f1 ", 0644);
                perror (" mkfifo ");
                Printf (" Before ---- %d\n", getpid()) ;
                fd = open ("f1", O_ONLY ) ;
                Printf (" After --- %d\n ", getpid()) ;

    read ( fd, a, sizeof (a)) ;
    Printf (" Data : %s\n", a) ;
}
```

Program-2 One process continue write the data and second process continue Read the data from FIFO.



1) Vi P1.c (write)

```
# include " header.h "
main()
{
        char  a [20] ;
        int fd ;
```

```
        mkfifo ("f1", 0644);
        ~~mkfifo ("f~~
        Perror ("mkfifo");
        Printf ("Before ---- %d\n", getpid());
        fd = open ("f1", O_WRONLY);
        Printf ("After ---- %d\n", getpid());

    while(1)
    {
        printf ("Enter the data ---\n");
        Scanf ("%s\n", a);
        write (fd, a, strlen(a)+1);
    }
    Printf ("Done ----\n");
}
```

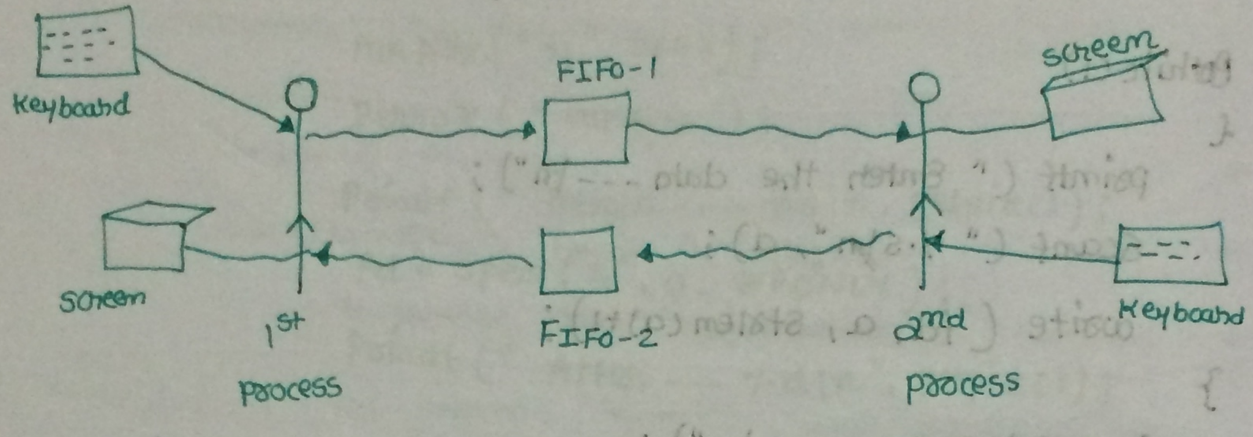
2) Vi P2.C (Read)

```
# include "header.h"
main()
{
    char a[20];
    int fd;
        mkfifo ("f1", 0644);
        Perror ("mkfifo");
        Printf ("Before ---- %d\n", getpid());
        fd = open ("f1", O_RDONLY);
        Printf ("After ---- %d\n", getpid());

    while(1)
    {
        read (fd, a, sizeof(a));
        Printf ("Data : %s\n", a);
    }
}
```

Program-3 One process write the data in FIFO_1 after other process can read data from the FIFO_1.

- 2nd process write the data on FIFO-2 and 1st process Read data from FIFO-2.

(half duplex communication).



1) **Vi P1.c :- (write)**

```
#include "header.h"
main()
{
    char a[20], b[20];
    int fd, fd1;
        mkfifo("f1", 0644);
        mkfifo("f2", 0644);
        fd = open("f1", O_WRONLY);
        fd1 = open("f2", O_RDONLY);
while(1)
{
        Printf("Enter the data");
        Scanf("%s", a);
        write(fd, a, strlen(a)+1);
read(fd1, b, sizeof(b));
Printf("Data : %s\n", b);
}
}
```

2) Vi P2.c :- (Read)

```
# include "header.h"
main()
{
    Char a[20], b[20]
    int fd, fd1;
            mkfifo ("f1", 0644);
            mkfifo ("f2", 0644);
    fd= open ("f1", O_RDONLY);
    fd1= open ("f2", O_WRONLY);
    while(1)
    {
        read (fd, a, sizeof(a));
        Printf (" Data : %s \n", a);
        Printf (" Enter the data ---\n");
        Scanf (" %s ", b);
        write (fd1, b, strlen(b)+1);
    }
}
```

Program-4 Full-duplex Communication System :-

Parent → Scan the data from the Keyboard.

Child → print data on monitor Screen.

```
if (fork() )        Process-1
{
    write(  );
}
else
{
    read(  );
}
```

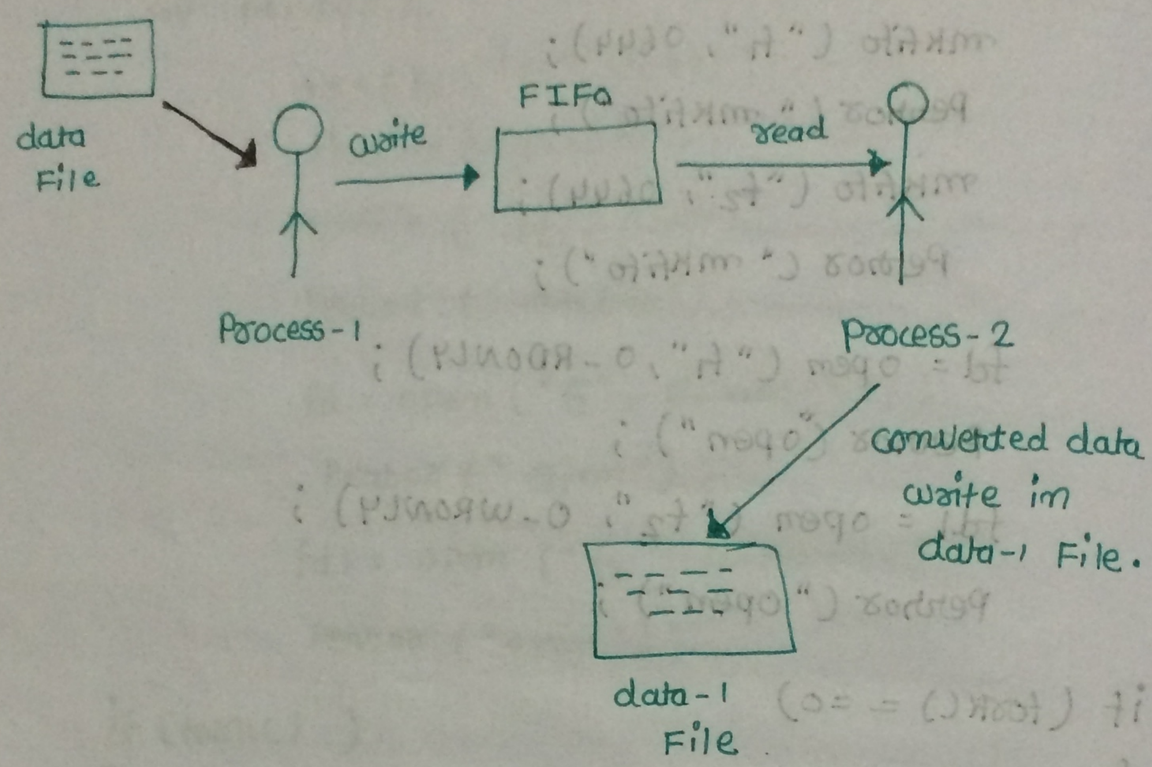
```
if (fork() == 0)
{                   Process-2
    read(  );
}
else
{
    Scanf( );
    write( );
}
```

1) vi P1.c

```
# include " header.h "

main()
{
    char a[20], b[20];
    int fd, fd1;
        mkfifo ("f1", 0644);
        Perror (" mkfifo");
        mkfifo ("f2", 0644);
        Perror (" mkfifo");
        fd = open ("f1", O_WRONLY);
        Perror (" open");
        fd1 = open ("f2", O_RDONLY);
        Perror ("open");
    if (fork())
    {
        while(1)
        {
            Printf (" Enter the data----\n");
            Scanf (" %s", a);
            write (fd, a, strlen(a)+1);
        }
    }
    else
    {
        while(1)
        {
            read (fd1, b, Sizeof(b));
            Printf (" DATA : %s\n", b);
        }
    }
}
```

Program-5 :- 1st process scan the data from (data file) Keyboard and put in FIFO, other process is read the data from FIFO.

- 2nd process convert the data and write in data-1 file.



1) V: P1.c

```
# include "header.h"
main()
{
        int fi, fd ;
        char s[100] ;
            mkfifo ("fi", 0644) ;
            Perror ("mkfifo") ;
            fd = Open ("fi", O_WRONLY) ;
            Perror ("open") ;
    fi = open ("data", O_RDONLY) ;
    read (fi, s, sizeof(s)) ;
    write (fd, s, sizeof(s)+1) ;
}
```