

# Exp-19 Inter-Process Comm

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <limits.h>
#include <sys/types.h>
#include <sys/stat.h>
```

\* (WRITE)

using namespace std;

```
#define BUFFER_SIZE PIPE_BUF
```

```
int main (int argc, char *argv[])
```

```
{
    int pipe-fd;
```

```
    char buffer [BUFFER_SIZE + 1];
```

```
    if (argc != 2)
```

```
    {
        cout << "usage : ./a.out pipe-name \n";
        return 1;
    }
```

```
    if (access (argv[1], F_OK) == -1)
```

```
    {
        if (mkfifo (argv[1], 0777))
```

```
        {
```

```
            perror ("mkfifo error \n");
```

```
            exit 0;
        }
```

```
    }
```

```
    cout << "Process " << getpid() << " opening fifo in write mode " << endl;
```

```
    pipe-fd = open (argv[1], O_WRONLY);
```



```

    cout << "FD of fifo in write mode : " << pipe-fd << endl;
    if (pipe-fd != -1)
    {
        cout << "enter data\n";
        cin >> buffer;
        int res = write (pipe-fd, buffer, BUFFER-SIZE);
        if (res == -1)
        {
            perror ("write error\n");
            exit (0);
        }
        close (pipe-fd);
    }
    else
        perror ("fifo write");
    cout << "\n process" << getpid() << " finish writing\n" << endl;
    unlink (argv[1]);
    return 0;
}

```

\* (READ)

```

using namespace std;
#define BUFFER-SIZE PIPE-BUF
int main (int argc, int argv[1])
{
    int pipe-fd, res = 0;
    char buffer [BUFFER-SIZE + 1];
    if (argc != 2)
    {
        cout << "usage : ./a.out pipe-name\n";
        return -1;
    }
    cout << "\n FD of fifo in read mode : " << pipe-fd << endl;
}

```



```

if (( pipe-fd = open (argv[i], O_RDONLY)) != -1 )
{
    nos = read (pipe-fd, buffer, BUFFER-SIZE);
    cout << "In data read .. \n";
    cout << buffer;
    close (pipe-fd);
}
else
{
    perror ("In fifo read \n");
}
cout << "In process " << getpid() << " finish reading \n" << endl;
return 0;
}

```