

Writeup for Q2

Note:- make file will only generate p1 and p2 executable files. One must type the ./p1 and ./p2 commands after make command.

Moreover, commands should follow this order

1:- make

2:- ./p1

3:- ./p2

If commands are executed in a different order then the program might not execute.

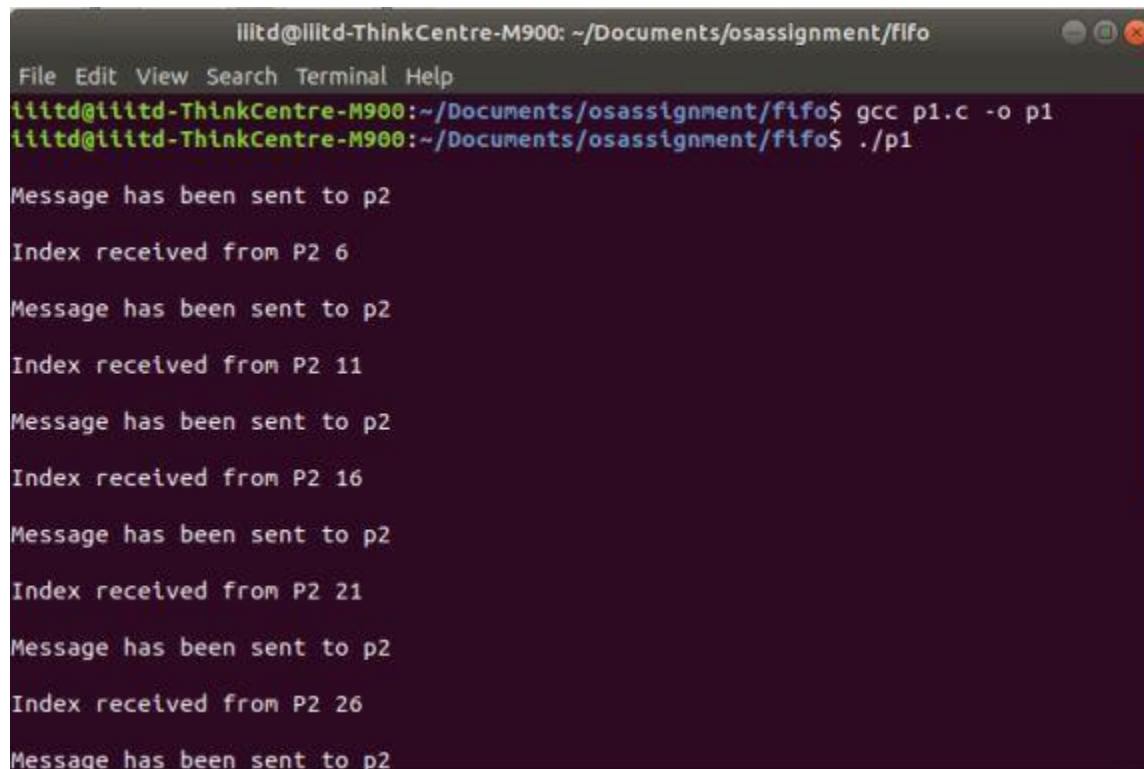
FIFO IPC:-

To establish a two-way communication using FIFO we must use named FIFO with a common file descriptor in the /tmp directory, so that both the process can communicate with each other.

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

mkfifo() system call is used to create a special FIFO file with a specific path and mode. More can be read, write or both. After creating the fifo file we shall open the file using open() system call and use read() and write() to read and write from the file.

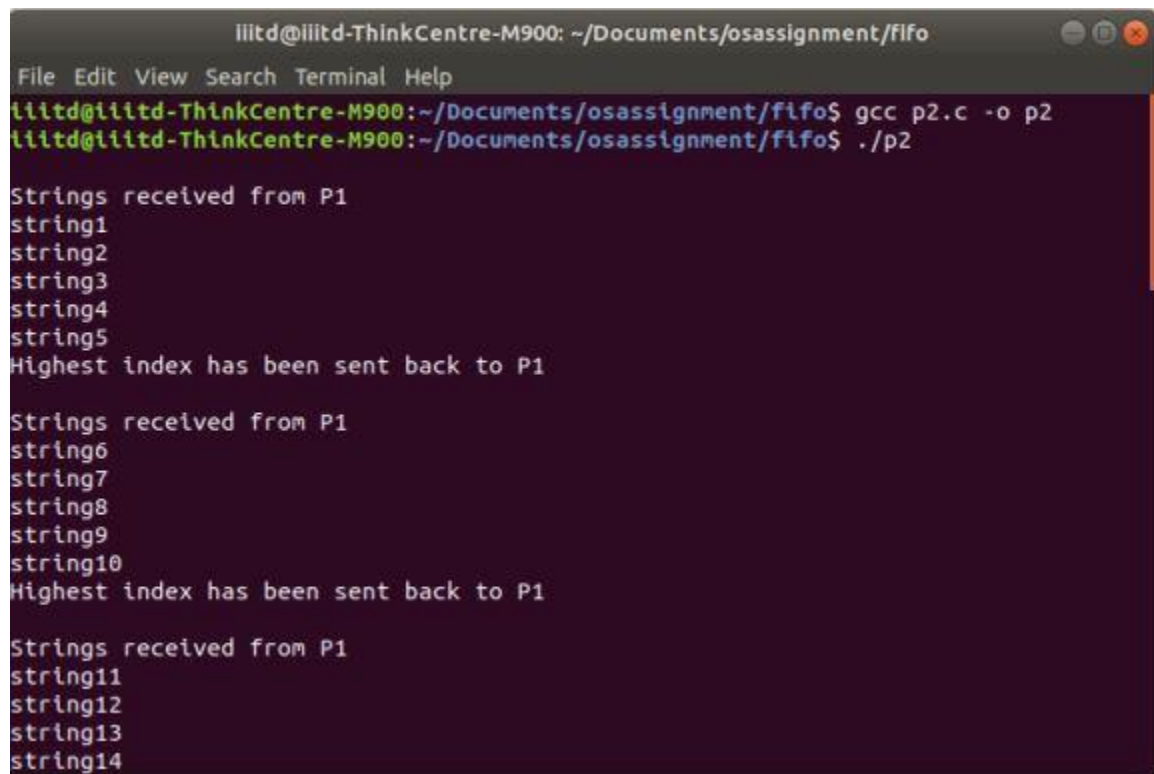
Screenshot of the output of the p1:-



```
lilitd@lilitd-ThinkCentre-M900: ~/Documents/osassignment/fifo
File Edit View Search Terminal Help
lilitd@lilitd-ThinkCentre-M900:~/Documents/osassignment/fifo$ gcc p1.c -o p1
lilitd@lilitd-ThinkCentre-M900:~/Documents/osassignment/fifo$ ./p1

Message has been sent to p2
Index received from P2 6
Message has been sent to p2
Index received from P2 11
Message has been sent to p2
Index received from P2 16
Message has been sent to p2
Index received from P2 21
Message has been sent to p2
Index received from P2 26
Message has been sent to p2
```

Screenshot of the output of the p2:-



```
lilitd@lilitd-ThinkCentre-M900: ~/Documents/osassignment/fifo
File Edit View Search Terminal Help
lilitd@lilitd-ThinkCentre-M900:~/Documents/osassignment/fifo$ gcc p2.c -o p2
lilitd@lilitd-ThinkCentre-M900:~/Documents/osassignment/fifo$ ./p2

Strings received from P1
string1
string2
string3
string4
string5
Highest index has been sent back to P1

Strings received from P1
string6
string7
string8
string9
string10
Highest index has been sent back to P1

Strings received from P1
string11
string12
string13
string14
```

In the program p1 is sending a group of strings with the index to p2 and p2 is printing them on the screen and sending the highest index back to p1. This process will be repeated until all the 50 strings are sent to p2.

Messages passing using QUEUE:-

To use the queue as a medium of inter-process communication we shall have to define a structure with two fields.

```
struct message
{
    long type;
    char data[100];
};
```

type is the message type and it is long int. Data is the message that we wish to send to the other process.

To start the communication we must define a common key using `ftok()` system call. A key is a common file that both the processes will use to communicate.

```
key = ftok("common.c",'b');
```

After defining a common key we shall use `msgsnd()` system call to send the message to another program/process.

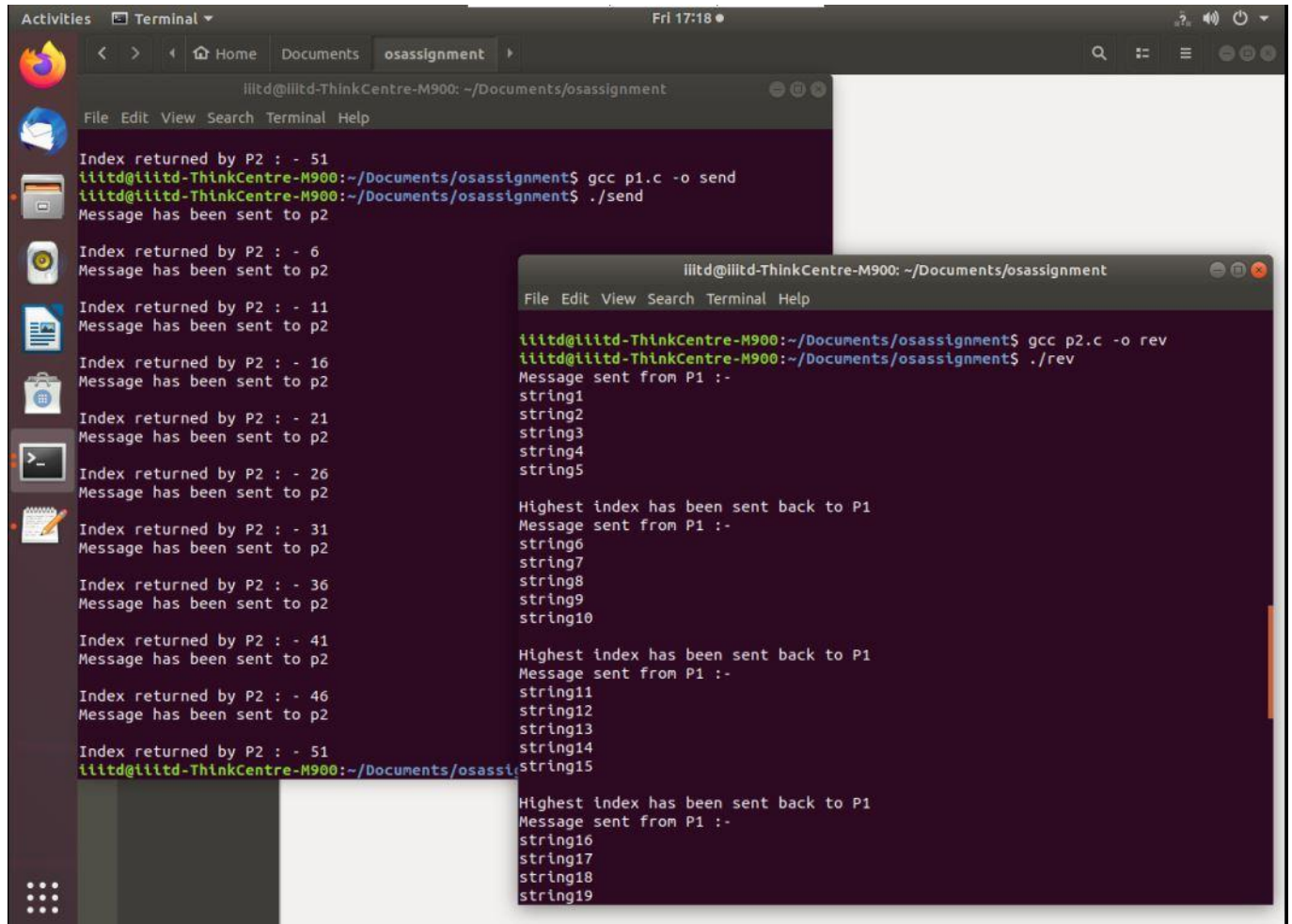
```
msgsnd(msgid, &msg, sizeof(msg), 0)
```

where msg is the instance of the structure.

After sending the message from the p1 we shall receive that message in p2 using `msgrcv()` system call.

```
msgrcv(msgid,&msg,sizeof(msg),1,0)
```

Screenshot of the output:-



```
llitd@llitd-ThinkCentre-M900: ~/Documents/osassignment
File Edit View Search Terminal Help

Index returned by P2 : - 51
llitd@llitd-ThinkCentre-M900:~/Documents/osassignment$ gcc p1.c -o send
llitd@llitd-ThinkCentre-M900:~/Documents/osassignment$ ./send
Message has been sent to p2

Index returned by P2 : - 6
Message has been sent to p2

Index returned by P2 : - 11
Message has been sent to p2

Index returned by P2 : - 16
Message has been sent to p2

Index returned by P2 : - 21
Message has been sent to p2

Index returned by P2 : - 26
Message has been sent to p2

Index returned by P2 : - 31
Message has been sent to p2

Index returned by P2 : - 36
Message has been sent to p2

Index returned by P2 : - 41
Message has been sent to p2

Index returned by P2 : - 46
Message has been sent to p2

Index returned by P2 : - 51
llitd@llitd-ThinkCentre-M900:~/Documents/osassignment$ gcc p2.c -o rev
llitd@llitd-ThinkCentre-M900:~/Documents/osassignment$ ./rev
Message sent from P1 :-
string1
string2
string3
string4
string5

Highest index has been sent back to P1
Message sent from P1 :-
string6
string7
string8
string9
string10

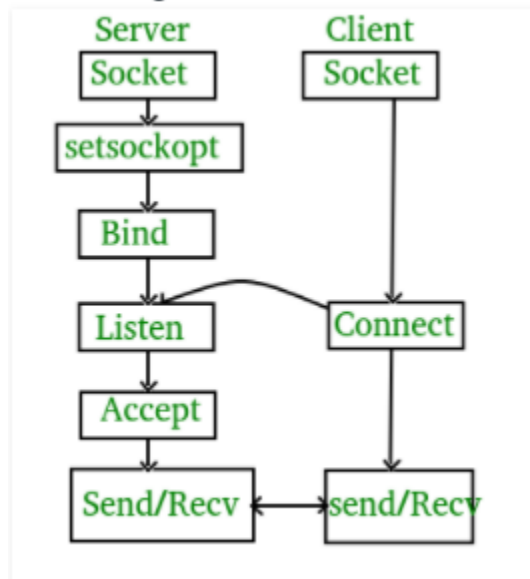
Highest index has been sent back to P1
Message sent from P1 :-
string11
string12
string13
string14
string15

Highest index has been sent back to P1
Message sent from P1 :-
string16
string17
string18
string19
```

IPC using UNIX DOMAIN SOCKETS

Socket programming is a way to connect two nodes on a network to communicate with each other.

There are many stages to establish the network between the two processes.



First we create a socket using `socketfd()` system call

```
int socketfd = socket(domain, type, protocol)
```

Then we bind the socket and port number using `bind()` system call

```
int bind(int socketfd, const struct sockaddr *addr, socklen_t addrlen);
```

To receive a message we wait for the other process to send something using `listen()` system call

```
int listen(int socketfd, int backlog);
```

To accept the connection request we use `accept()` system call

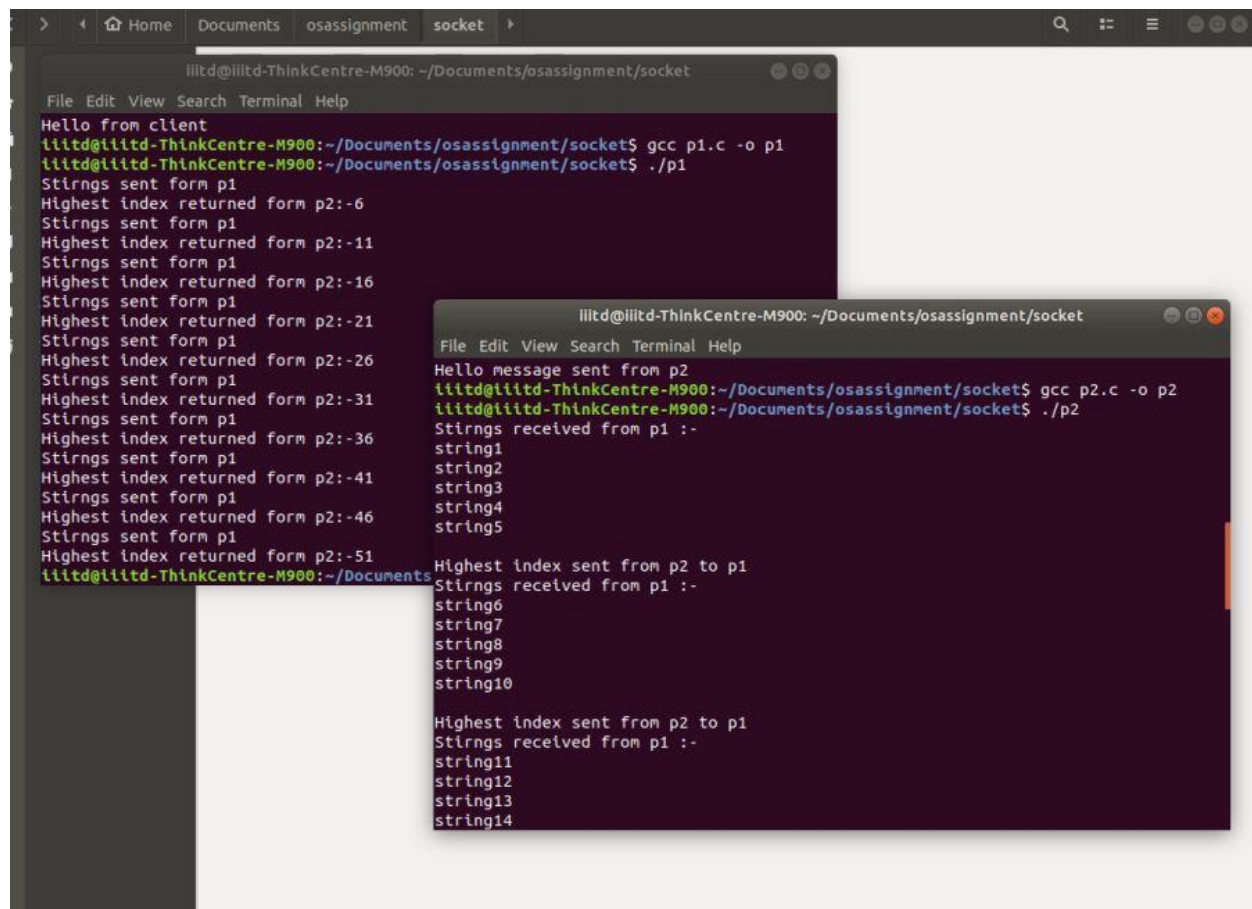
```
int new_socket= accept(int socketfd, struct sockaddr *addr, socklen_t *addrlen);
```

To connect the two process to exchange messages we use connect() system call

```
int connect(int sockfd, const struct sockaddr *addr,  
            socklen_t addrlen);
```

Once the connection has been established now we can exchange messages using read() and send() system calls.

Output of the Unix domain socket program:-



```
llltd@llltd-ThinkCentre-M900: ~/Documents/osassignment/socket
File Edit View Search Terminal Help
Hello from client
llltd@llltd-ThinkCentre-M900:~/Documents/osassignment/socket$ gcc p1.c -o p1
llltd@llltd-ThinkCentre-M900:~/Documents/osassignment/socket$ ./p1
Strings sent form p1
Highest index returned form p2:-6
Strings sent form p1
Highest index returned form p2:-11
Strings sent form p1
Highest index returned form p2:-16
Strings sent form p1
Highest index returned form p2:-21
Strings sent form p1
Highest index returned form p2:-26
Strings sent form p1
Highest index returned form p2:-31
Strings sent form p1
Highest index returned form p2:-36
Strings sent form p1
Highest index returned form p2:-41
Strings sent form p1
Highest index returned form p2:-46
Strings sent form p1
Highest index returned form p2:-51
llltd@llltd-ThinkCentre-M900:~/Documents

llltd@llltd-ThinkCentre-M900: ~/Documents/osassignment/socket
File Edit View Search Terminal Help
Hello message sent from p2
llltd@llltd-ThinkCentre-M900:~/Documents/osassignment/socket$ gcc p2.c -o p2
llltd@llltd-ThinkCentre-M900:~/Documents/osassignment/socket$ ./p2
Strings received from p1 :-
string1
string2
string3
string4
string5

Highest index sent from p2 to p1
Strings received from p1 :-
string6
string7
string8
string9
string10

Highest index sent from p2 to p1
Strings received from p1 :-
string11
string12
string13
string14
```