**Q)2 way multiple time communication using pipe**

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

void main(){

int fd1[2],fd2[2];

char message[1024],buffer[1024];

int pid=0;

if(pipe(fd1)==-1){

perror("Pipe Creation Failed \n");

exit(0);

}

if(pipe(fd2)==-1){

perror("Pipe Creation Failed \n");

exit(0);

}

pid=fork();

if(pid>0){

close(fd1[0]);

close(fd2[1]);

while(1){

memset(message,0,sizeof(message));

printf("Write data for child: ");

gets(message);

write(fd1[1],message,1024);

if(strcmp(message,"exit")==0) break;

memset(buffer,0,sizeof(buffer));

read(fd2[0],buffer,1024);

printf("Recieved data from child: %s \n",buffer);

if(strcmp(buffer,"exit")==0) break;

}

}

else{

close(fd1[1]);

close(fd2[0]);

while(1){

memset(buffer,0,sizeof(buffer));

read(fd1[0],buffer,1024);

printf("Received data from parent: %s\n ",buffer);

if(strcmp(buffer,"exit")==0) break;

memset(message,0,sizeof(message));

printf("Write data for parent:");

gets(message);

write(fd2[1],message,1024);

if(strcmp(message,"exit")==0) break;

}

}

}

**Q2) FIFO 2 way**

**Read-write.c**

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

void main(){

int fd;

char \*myfifo="abc";

char message[1024],buffer[1024];

mkfifo(myfifo,0666);

while(1)

{

memset(buffer,0,sizeof(buffer));

fd=open(myfifo, O\_RDONLY);

read(fd, buffer,1024);

printf("Received data: %s\n",buffer);

if(strcmp(buffer,"exit")==0) break;

close(fd);

memset(message,0,sizeof(message));

fd=open(myfifo, O\_WRONLY);

printf("Enter Input string:");

gets(message);

write(fd, message, 1024);

if(strcmp(message,"exit")==0) break;

close(fd);

}

}

**Write-read.c**

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

void main(){

int fd;

char \*myfifo="abc";

char message[1024],buffer[1024];

mkfifo(myfifo,0666);

while(1)

{

memset(message,0,sizeof(message));

fd=open(myfifo, O\_WRONLY);

printf("Enter Input string:");

gets(message);

write(fd, message, 1024);

if(strcmp(message,"exit")==0) break;

close(fd);

memset(buffer,0,sizeof(buffer));

fd=open(myfifo, O\_RDONLY);

read(fd, buffer,1024);

printf("Received data: %s\n",buffer);

if(strcmp(buffer,"exit")==0) break;

close(fd);

}

}

**Q3)2 way multiple time communication using tcp/ip**

**TCP SERVER**

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 8090

void main()

{ int opt=1;

int svrsock\_fd,new\_conn;

char buffer[1024],message[1024];

struct sockaddr\_in address;

socklen\_t addrlen=sizeof(struct sockaddr\_in);

svrsock\_fd=socket(AF\_INET,SOCK\_STREAM,0);

address.sin\_family=AF\_INET;

address.sin\_addr.s\_addr=INADDR\_ANY;

address.sin\_port=htons(PORT);

setsockopt(svrsock\_fd,SOL\_SOCKET,SO\_REUSEADDR|SO\_REUSEPORT,opt,&opt);

bind(svrsock\_fd,(struct sockadddr\*)&address,addrlen);

printf("waiting for client\n");

listen(svrsock\_fd,3);

new\_conn=accept(svrsock\_fd,(struct sockaddr\*)&address,&addrlen);

while(1)

{

memset(buffer,0,sizeof(buffer));

read(new\_conn,buffer,1024);

printf("received data from TCP/IP client:%s\n",buffer);

if(strcmp(buffer,"exit")==0)break;

memset(message,0,sizeof(message));

printf("enter data for TCP/IP client:");

gets(message);

send(new\_conn,message,strlen(message),0);

if(strcmp(message,"exit")==0)break;

}

}

**TCP CLIENT**

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 8090

void main()

{

int clnsock\_fd;

char buffer[1024],message[1024];

struct sockaddr\_in svraddr;

socklen\_t svraddrlen=sizeof(struct sockaddr\_in);

clnsock\_fd=socket(AF\_INET,SOCK\_STREAM,0);

svraddr.sin\_family=AF\_INET;

svraddr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

svraddr.sin\_port=htons(PORT);

connect(clnsock\_fd,(struct sockadddr\*)&svraddr,svraddrlen);

while(1)

{

memset(message,0,sizeof(message));

printf("enter data for TCP/IP server:");

gets(message);

send(clnsock\_fd,message,strlen(message),0);

if(strcmp(message,"exit")==0)break;

memset(buffer,0,sizeof(buffer));

read(clnsock\_fd,buffer,1024);

printf("received data from TCP/IP server:%s\n",buffer);

if(strcmp(buffer,"exit")==0)break;

}

}

**Q4)UDP CONNECTION 2 way**

**UDP SERVER**

#include<stdio.h>

#include<unistd.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 8080

void main()

{char buffer[1024],message[1024];

int svrsock\_fd;

struct sockaddr\_in svraddr,clnaddr;

socklen\_t svraddrlen=sizeof(struct sockaddr\_in);

socklen\_t clnaddrlen=sizeof(struct sockaddr\_in);

svrsock\_fd=socket(AF\_INET,SOCK\_DGRAM,0);

svraddr.sin\_family=AF\_INET;

svraddr.sin\_addr.s\_addr=INADDR\_ANY;

svraddr.sin\_port=htons(PORT);

bind(svrsock\_fd,(struct sockaddr \*)&svraddr,svraddrlen);

printf("WAITING FOR UDP/IP CLIENT\n");

while(1)

{

memset(buffer,0,sizeof(buffer));

recvfrom(svrsock\_fd,buffer,sizeof(buffer),0,&clnaddr,&clnaddrlen);

printf("received data from UDP/IP CLIENT:%s\n",buffer);

if(strcmp(buffer,"exit")==0)break;

memset(message,0,sizeof(message));

printf("enter data for UDP/IP client:");

gets(message);

sendto(svrsock\_fd,message,sizeof(message),0,&clnaddr,clnaddrlen);

if(strcmp(message,"exit")==0)break;

}

}

**UDP CLIENT**

#include<stdio.h>

#include<unistd.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 8080

void main()

{char message[1024],buffer[1024];

int clnsock\_fd;

struct sockaddr\_in svraddr,clnaddr;

socklen\_t svraddrlen=sizeof(struct sockaddr\_in);

socklen\_t clnaddrlen=sizeof(struct sockaddr\_in);

clnsock\_fd=socket(AF\_INET,SOCK\_DGRAM,0);

svraddr.sin\_family=AF\_INET;

svraddr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

svraddr.sin\_port=htons(PORT);

while(1)

{

memset(message,0,sizeof(message));

printf("enter data for UDP/IP server:");

gets(message);

sendto(clnsock\_fd,message,sizeof(message),0,&svraddr,svraddrlen);

if(strcmp(message,"exit")==0)break;

memset(buffer,0,sizeof(buffer));

recvfrom(clnsock\_fd,buffer,sizeof(buffer),0,&svraddr,&svraddrlen);

printf("received data from UDP/IP server:%s\n",buffer);

if(strcmp(buffer,"exit")==0)break;

}

}