**Week-4**

Implement the Client server communication using Internet Domain Sockets

**Code for Server:**

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#define MAX 80

#define PORT 3262

#define SA struct sockaddr

void func(int sockfd){

char buff[MAX];

int n;

for(;;){

bzero(buff,MAX);

read(sockfd,buff,sizeof(buff));

printf("From client: %s\t To client:",buff);

bzero(buff,MAX);

n=0;

while((buff[n++]=getchar())!='\n');

write(sockfd,buff,sizeof(buff));

if(strncmp("exit",buff,4)==0){

printf("Server exit..\n");

break;

}

}

}

int main(){

int sockfd,connfd,len;

struct sockaddr\_in servaddr,cli;

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

if(sockfd==-1){

printf("Socket creation failed..\n");

exit(0);

}else{

printf("Socket successfully created..\n");

}

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

servaddr.sin\_port = htons(PORT);

if((bind(sockfd,(SA\*)&servaddr,sizeof(servaddr))) != 0){

printf("Socket bind failed..\n");

exit(0);

}else{

printf("Socket successfully binded..\n");

}

if((listen(sockfd,5))!=0){

printf("Listen failed..\n");

exit(0);

}else{

printf("Server listening..\n");

}

len = sizeof(cli);

connfd = accept(sockfd,(SA\*)&cli,&len);

if(connfd < 0){

printf("Server accept failed..\n");

exit(0);

}else{

printf("Server accepted the client..\n");

}

func(connfd);

close(sockfd);

return 0;

}

**Code for Client:**

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#define MAX 80

#define PORT 3262

#define SA struct sockaddr

void func(int sockfd){

char buff[MAX];

int n;

for(;;){

bzero(buff,sizeof(buff));

printf("Enter the string:");

n = 0;

while((buff[n++]=getchar())!='\n');

write(sockfd,buff,sizeof(buff));

bzero(buff,sizeof(buff));

read(sockfd,buff,sizeof(buff));

printf("From server:%s",buff);

if((strncmp(buff,"exit",4))==0){

printf("Client exit..\n");

break;

}

}

}

int main(){

int sockfd,connfd;

struct sockaddr\_in servaddr;

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

if(sockfd == -1){

printf("Socket creation failed..\n");

exit(0);

}else{

printf("Socket successfully created..\n");

}

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(PORT);

if(connect(sockfd,(SA\*)&servaddr,sizeof(servaddr)) != 0){

printf("connection with server failed..\n");

exit(0);

}else{

printf("Connected to server..\n");

}

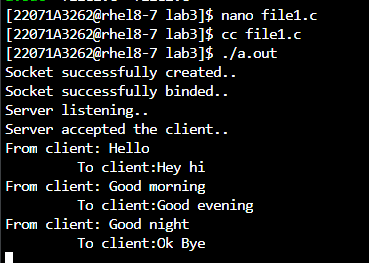
func(sockfd);

close(sockfd);

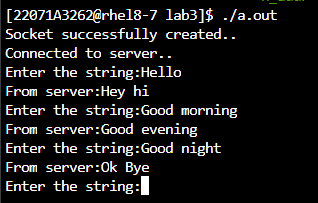
return 0;

}

**Output for Server:**

****

**Output for Client:**

****