Nachiket G Kallapur

USN: 1RV18CS096

Lab Internal

Prog 7

Server.c

```
#include<stdio.h>
#include <stdlib.h>
#include<string.h>
#include<unistd.h>
#include<netinet/in.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <errno.h>
int main(){
    int sd, acpt, len, bytes, port;
    char send[50], receive[50];
    struct sockaddr_in serv, cli;
    if ((sd = socket(AF_INET, SOCK_STREAM, 0)) < 0){
        printf("Error in socket\n");
        exit(0);
    }

    bzero(&serv, sizeof(serv));
    serv.sin_family = AF_INET;
    serv.sin_port = htons(8000);
    serv.sin_addr.s_addr = htonl(INADDR_ANY);
    if(bind(sd (struct sockaddr *)&serv, sizeof(serv)) < 0) {
        printf("Error in Bind\n");
        exit(0)
    }
if(listen(sd,3)<0){
        printf("Error in listen");
        exit(0);
  }
 if((acpt =accept(sd,(struct sockaddr*)NULL,NULL))<0){
        printf("Error in accept");
        exit(0);
  }
 while(1){

        bytes=recv(acpt,receive,50,0);
        receive[bytes]='\0';
        if(strcmp(receive,"end")==0){
            close(acpt);
            close(sd);
            exit(0);
        }

      else {
            printf("Command Received: %s",receive);
            system(receive);
            printf("\n");
           }
      }
 }
```

client.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<errno.h>

int main(){

        int sd, acpt, len, bytes, port;
        char send[50], receive[50];
        struct sockaddr_in serv, cli;
        if((sd = socket(AF_INET, SOCK_STREAM,0)) >0){
            printf("Error in socket\n");
            exit(0);
        }
        bzero(&serv, sizeof(serv));
        serv.sin_family = AF_INET;
        serv.sin_port = htons(8000);
        serv.sin_addr.s_addr = htonl(INADDR_ANY);
        if(connect(sd, (struct sockaddr*)&serv, sizeof(serv))<0){
            printf("Error in connection\n");
            exit(0);
        }
while(1){
  printf("Enter the command: ");
   gets(send1);
   if(strcmp(send1,"end")!=0){
     send(sd, send1, 50, 0);
      close(sd);
       break;
     }
   }
}
```

Marks - PartA - Programs

Prog1:	6 + 4 = 10
Prog2:	6+4 = 10
Prog3:	6 + 4 = 10
Prog4:	6 + 4 = 10
Prog5:	6 + 4 = 10
Prog6:	6+4=10
Prog7:	6+4=10
Prog8:	6+4=10

Total: 80/80 Reduced to: 30/30

Part B - 9/10

Internal: 10/10

Total: 49/50