```
//Server

#include<sys/socket.h>
#include<arpa/inet.h>
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>
#define maxlen 70000
#define mlen 100000
int main()
{
  char fileName[100];
  char filebuffer[2000],caufile[maxlen];
  char *vfilep;
  int aufile[700000],vfile[mlen];
  int sd,connfd,len;

for(int i=0;i<=100;i++){
 fileName[i]='\0';
}
  struct sockaddr_in servaddr,cliaddr;

  sd = socket(AF_INET, SOCK_DGRAM, 0);

  if(sd==-1)
    {
     printf(" socket not created in server\n");
     exit(0);
    }
  else
    {
     printf("socket created in  server\n");
    }

  bzero(&servaddr, sizeof(servaddr));

  servaddr.sin_family = AF_INET;
  servaddr.sin_addr.s_addr = INADDR_ANY;
  servaddr.sin_port = htons(8000);
  memset(&(servaddr.sin_zero),'\0',8);
  if ( bind(sd, (struct sockaddr *)&servaddr, sizeof(servaddr)) != 0 )
    printf("Not binded\n");
  else
    printf("Binded\n");

  len=sizeof(cliaddr);

  int choice =1;
  while(1)
  {
   char num;
```

```
  recvfrom(sd,&num,sizeof(num),0,(struct sockaddr *)&cliaddr, &len);


choice = num;


switch(choice)
{
 case 1:
  recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
    printf("NAME OF TEXT FILE RECEIVED : %s\n",fileName);
  FILE *fp;
     printf("Contents in the received text file : \n");
     recvfrom(sd,filebuffer,1024,0,(struct sockaddr *)&cliaddr, &len);
     printf("%s\n",filebuffer);
    int fsize=strlen(filebuffer);
  fp=fopen(fileName,"w");
    if(fp)
    {
    fwrite(filebuffer, fsize, 1, fp);
     printf("File received successfully.\n");
    }
    else
    {
     printf("Cannot create to output file.\n");
    }
    memset(fileName, '\0', sizeof(fileName));
    fclose(fp);
    break;
   case 2:
    recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
     printf("NAME OF AUDIO FILE RECEIVED : %s\n",fileName);
     FILE *afp;
     int numbytes;
       afp=fopen(fileName,"w");
       size_t  afsize;
       afsize=recvfrom(sd,aufile,700000,0,(struct sockaddr *)&cliaddr, &len);
       if(afp)
       {
        fwrite(aufile, afsize, 1, afp);
        printf("File received successfully.\n");
       }
       else
       {
        printf("Cannot open output file.\n");
       }
       memset(fileName, '\0', sizeof(fileName));
       fclose(afp);
       break;

      case 3:
       recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
         printf("VIDEO FILE NAME RECEIVED : %s\n",fileName);
```

```
        FILE *vfp;
        vfp=fopen(fileName,"w");
        size_t  vfsize;
        vfsize=recvfrom(sd,vfile,100000,0,(struct sockaddr *)&cliaddr, &len);

        if(vfp)
        {
           fwrite(vfile, vfsize, 1, vfp);
           printf("File received successfully.\n");
          }
          else
          {
           printf("Cannot open output file.\n");
          }
       fclose(vfp);
       break;

      case 4:
    close(sd);
    break;

 }
  }
  return(0);
}



//Client

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

int main() {
 int fd;
 char fileName[2000],afileName[2000],vfileName[2000],file_buffer[2000],c,caufile[70000],aufile[7000000],vfile[1
000000];
 struct sockaddr_in  servaddr;

 // Creating socket file descriptor
 if ( (fd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
  perror("socket creation failed");
  exit(EXIT_FAILURE);
 }

 memset(&servaddr, 0, sizeof(servaddr));

 bzero(&servaddr,sizeof(servaddr));
```

```
 // Filling server information
 servaddr.sin_family = AF_INET;
 servaddr.sin_port = htons(8000);
 servaddr.sin_addr.s_addr = INADDR_ANY;
// servaddr.sin_addr.s_addr=inet_addr("10.10.10.73");
 int choice = 1;

 while(choice!=4)
 {
  printf("ENTER \n 1.TEXT \n 2.AUDIO \n 3.VIDEO\n4.EXIT");
  scanf("%d",&choice);

  char num=choice;

  sendto(fd, &num, sizeof(num), 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr));

  switch(choice)
  {
   case 1:
    printf("Enter text file name to send : \n");
       scanf("%s",fileName);
       sendto(fd, fileName, strlen(fileName), 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr));

       FILE *fp;
        fp=fopen(fileName,"r");

        if(fp)
        {
         printf("Reading file contents.\n");
         fseek(fp,0,SEEK_END);
           size_t file_size=ftell(fp);
           fseek(fp,0,SEEK_SET);
         if(fread(file_buffer,file_size,1,fp)<=0)
           {
             printf("Unable to copy file into buffer or empty file.\n");
             exit(1);
           }
         }
         else
         {
        printf("Cannot open file.\n");
         exit(0);
        }
       printf("FILE CONTENTS TO SEND : %s\n",file_buffer);
       if(sendto(fd, file_buffer,strlen(file_buffer), 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr))<0)
       {
         printf("FILE WAS NOT SENT\n");
       }
       else
       {
         printf("FILE SENT\n");
       }
       fclose(fp);
       break;
```

```
   case 2:
    printf("Enter audio file name to send : \n");
    scanf("%s",afileName);
    sendto(fd, afileName, strlen(afileName), 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr));
 FILE *afp;
 afp=fopen(afileName,"r");
 fseek(afp,0,SEEK_END);
 size_t  afsize=ftell(afp);
 fseek(afp,0,SEEK_SET);

 if(afp)
 {
  printf("Reading file contents.\n");
   if(fread(aufile,afsize,1,afp)<=0)
         {
           printf("Unable to copy file into buffer or empty file.\n");
           exit(1);
         }
 }
 else
 {
  printf("Could not read audio file.\n");
  exit(0);
 }

 if(sendto(fd, aufile, afsize, 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr))<0)
 {
   printf("FILE WAS NOT SENT\n");
     }
     else
     {
       printf("FILE SENT\n");
     }
 fclose(afp);
 break;

case 3:
 printf("Enter video file name to send : \n");
    scanf("%s",vfileName);
 sendto(fd, vfileName, strlen(vfileName), 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr));
 FILE *vfp;
 vfp=fopen(vfileName,"r");

 fseek(vfp, 0, SEEK_END);
 size_t vfsize = ftell(vfp);
 fseek(vfp, 0, SEEK_SET);

 if(vfp)
 {
  if(fread(vfile, 1, vfsize, vfp)<=0)
  {
   printf("No contents or error reading file \n");
  }
```

```
    }
    else
    {
     printf("Could not read audio file.\n");
     exit(0);
    }
    if(sendto(fd, vfile, vfsize, 0,(struct sockaddr *)&servaddr, sizeof(struct sockaddr))<0)
    {
    printf("FILE WAS NOT SENT\n");
    }
    else
    {
      printf("FILE SENT\n");
    }
    fclose(vfp);
    break;

   case 4:
    close(fd);
    break;



  }


 }


}
```