```
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#define PORT “58001”
...
int fd,errcode;
ssize_t n;
socklen_t addrlen;
struct addrinfo hints,*res;
struct sockaddr_in addr;
char buffer[128];
...
```

UDP Server

```
  ...
fd=socket(AF_INET,SOCK_DGRAM,0);     //UDP socket
if(fd==-1) /*error*/exit(1);

memset(&hints,0,sizeof hints);
hints.ai_family=AF_INET;      // IPv4
hints.ai_socktype=SOCK_DGRAM; // UDP socket
hints.ai_flags=AI_PASSIVE;

errcode=getaddrinfo(NULL,PORT,&hints,&res);
if(errcode!=0) /*error*/ exit(1);

n=bind(fd,res->ai_addr, res->ai_addrlen);
if(n==-1) /*error*/ exit(1);

while (1){
  addrlen=sizeof(addr);
  n=recvfrom(fd,buffer,128,0,
             (struct sockaddr*)&addr,&addrlen);
  if(n==-1)/*error*/exit(1);
  write(1,"received: ",10);write(1,buffer,n);
   ...
  n=sendto(fd,buffer,n,0,
           (struct sockaddr*)&addr,addrlen);
  if(n==-1)/*error*/exit(1);
}
  ...
freeaddrinfo(res);
close(fd);
```

blocks until datagram received from a client

UDP Client

```
  ...
fd=socket(AF_INET,SOCK_DGRAM,0);     //UDP socket
if(fd==-1) /*error*/exit(1);

memset(&hints,0,sizeof hints);
hints.ai_family=AF_INET;             //IPv4
hints.ai_socktype=SOCK_DGRAM;        //UDP socket

errcode=getaddrinfo("tejo.tecnico.ulisboa.pt",PORT,&hints,&res);
if(errcode!=0) /*error*/ exit(1);

n=sendto(fd,"Hello!\n",7,0,res->ai_addr,res->ai_addrlen);
if(n==-1) /*error*/ exit(1);

   ...

addrlen=sizeof(addr);
n=recvfrom(fd,buffer,128,0,
           (struct sockaddr*)&addr,&addrlen);
if(n==-1) /*error*/ exit(1);

write(1,"echo: ",6); write(1,buffer,n);

   ...

freeaddrinfo(res);
close(fd);
```

```
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#define PORT “58001”
...
int fd,errcode;
ssize_t n;
socklen_t addrlen;
struct addrinfo hints,*res;
struct sockaddr_in addr;
char buffer[128];
...
```

TCP Client

```
  ...
fd=socket(AF_INET,SOCK_STREAM,0);   //TCP socket
if (fd==-1) exit(1); //error

memset(&hints,0,sizeof hints);
hints.ai_family=AF_INET;            //IPv4
hints.ai_socktype=SOCK_STREAM;      //TCP socket

errcode=getaddrinfo("tejo.tecnico.ulisboa.pt",PORT,&hints,&res);
if(errcode!=0)/*error*/exit(1);

n=connect(fd,res->ai_addr,res->ai_addrlen);
if(n==-1)/*error*/exit(1);

n=write(fd,"Hello!\n",7);
if(n==-1)/*error*/exit(1);

n=read(fd,buffer,128);
if(n==-1)/*error*/exit(1);

write(1,"echo: ",6); write(1,buffer,n);
 ...
freeaddrinfo(res);
close(fd);
```

TCP Server

```
   ...
fd=socket(AF_INET,SOCK_STREAM,0);    //TCP socket
if (fd==-1) exit(1); //error

memset(&hints,0,sizeof hints);
hints.ai_family=AF_INET;             //IPv4
hints.ai_socktype=SOCK_STREAM;       //TCP socket
hints.ai_flags=AI_PASSIVE;

errcode=getaddrinfo(NULL,PORT,&hints,&res);
if((errcode)!=0)/*error*/exit(1);

n=bind(fd,res->ai_addr,res->ai_addrlen);
if(n==-1) /*error*/ exit(1);

if(listen(fd,5)==-1)/*error*/exit(1);
 ...
while(1){
  addrlen=sizeof(addr);
  if((newfd=accept(fd,(struct sockaddr*)&addr,
                 &addrlen))==-1
              /*error*/ exit(1);

  n=read(newfd,buffer,128);
  if(n==-1)/*error*/exit(1);
  write(1,"received: ",10);write(1,buffer,n);

  n=write(newfd,buffer,n);
  if(n==-1)/*error*/exit(1);

  close(newfd);
  }
 ...
freeaddrinfo(res);
close(fd);
```

blocks until connection from client

connection establishment TCP three-way handshake