TÉCNICO LISBOA

UDP Client

```
#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 "58000"
...
int fd;
ssize_t n;
socklen_t addrlen;
struct addrinfo hints,*res;
struct sockaddr_in addr;
char buffer[128];
...
```

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

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

fd=socket(res->ai_family,res->ai_socktype,res->ai_protocol);
if(fd==-1)/*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);
```

UDP Server

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

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


fd=socket(res->ai_family,res->ai_socktype,
          res->ai_protocol);
if(fd==-1)/*error*/exit(1);


n=bind(fd,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,"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

TÉCNICO LISBOA

TCP Client

```
#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 "58000"
...
int fd;
ssize_t n;
socklen_t addrlen;
struct addrinfo hints,*res;
struct sockaddr_in addr;
char buffer[128];
...
```

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

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

fd=socket(res->ai_family,res->ai_socktype,res->ai_protocol);
if(fd==-1)/*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

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

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

fd=socket(res->ai_family,res->ai_socktype,
          res->ai_protocol);
if(fd==-1)/*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);
    ...
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);

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

blocks until connection from client

connection establishment TCP three-way handshake