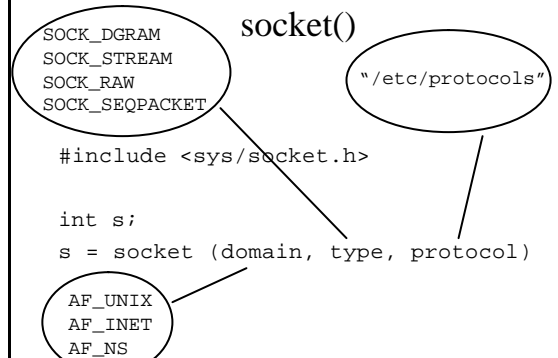
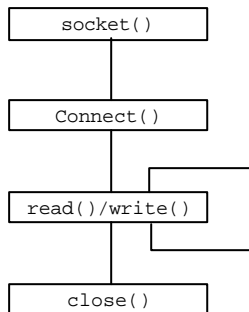
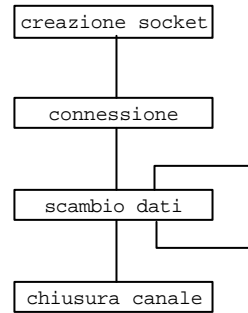


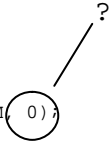
Implementazione di un client

Dobbiamo andare a costruire un programma in C che richiede un servizio



socket()

```
int s;  
s = socket(AF_INET, SOCK_STREAM, 0);  
If ( s < 0 ) {  
    perror("socket() ");  
    exit(1);  
}
```

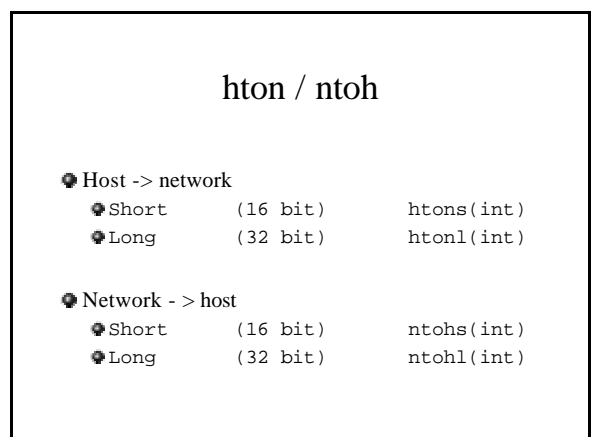
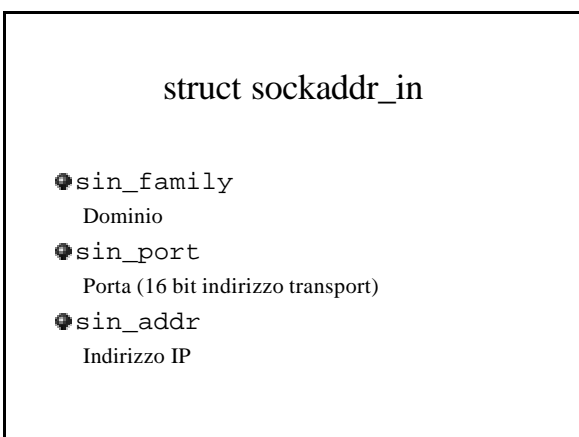
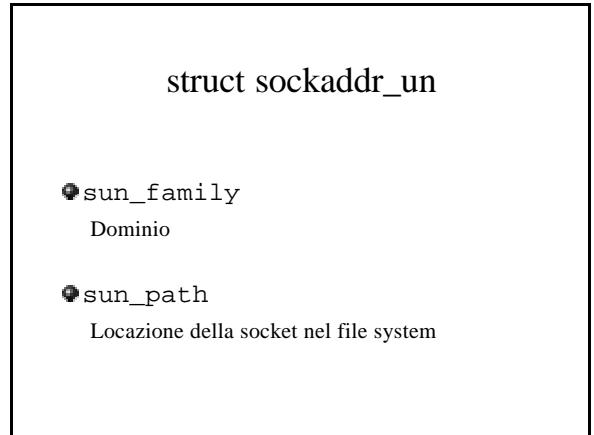
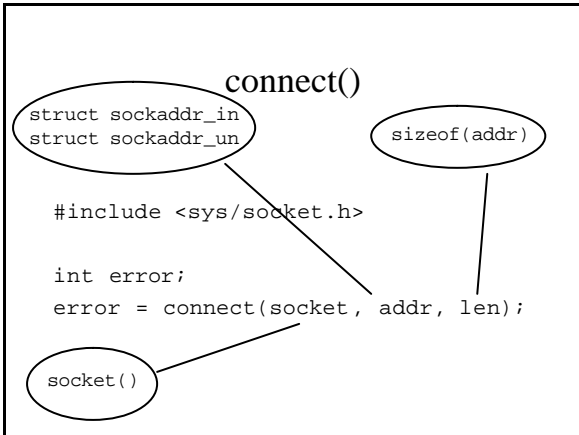


```
# Internet protocols  
#  
# FreeBSD: src/etc/protocols.v 1.13.2.1 20000924 11:26:39 asm odai Exp $  
# from: @(#)protocols 5.1 (Berkeley) 4/17/89  
#  
# See also http://www.isi.edu/in-notes/iana/assignments/protocol-numbers  
#  
ip 0 IP # internet protocol, pseudo protocol number  
icmp 1 ICMP # internet control message protocol  
igmp 2 IGMP # internet group management protocol  
ggp 3 GGP # gateway-gateway protocol  
ipencap 4 IP-ENCAP # IP encapsulated in IP (officially "IP")  
st2 5 ST2 # ST2 datagram mode (RFC 1819)  
tcp 6 TCP # transmission control protocol  
cbt 7 CBT # CBT, Tony Ballardie <-A.Ballardie@cs.ucl.ac.uk>  
egp 8 EGP # exterior gateway protocol  
nvp 11 NVP-II # Network Voice Protocol  
pup 12 PUP # PARC universal packet protocol  
argus 13 ARGUS # ARGUS  
emcon 14 EMCON # EMCON  
xnet 15 XNET # Cross Net Debugger  
chaos 16 CHAOS # Chaos  
udp 17 UDP # user datagram protocol  
...
```

getprotobyname()

```
#include <sys/socket.h>  
  
struct protoent *p;  
p = getprotobyname ("name")
```

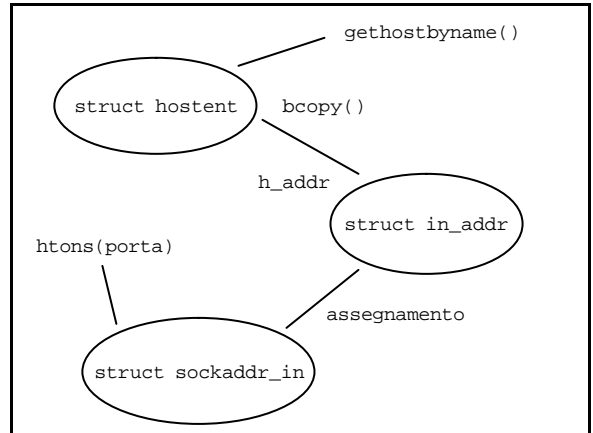
```
int s, protocol_number;  
struct protoent *protocol_entry;  
  
protocol_entry = getprotobyname("tcp");  
If ( protocol_entry == NULL ) {  
    perror("getprotobyname() ");  
    exit(1);  
}  
  
protocol_number = protocol_entry -> p_proto;  
  
s = socket(AF_INET, SOCK_STREAM, protocol_number);  
if ( s < 0 ) {  
    perror("socket() ");  
    exit(2);  
}
```



gethostbyname()

```
#include <sys/socket.h>

struct hostent *server;
server = gethostbyname("name");
```



```
struct hostent *server_host;
struct in_addr server_host_addr;
struct sockaddr_in server_addr;

server_host = gethostbyname("www.dsi.unimi.it");
if (!server_host) {
    perror("gethostbyname() ");
    exit(1);
}

bcopy(server_host->h_addr, &server_host_addr,
      server->h_length);

server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(12345);
bcopy(&server_host_addr, &server_addr.sin_addr,
      sizeof(server_host_addr));

if(connect(s, &server_addr, sizeof(server_addr))!=-1) {
    perror("connect() ");
    exit(1);
}
```

inet_addr()

```
#include <sys/socket.h>

unsigned long addr;
addr = inet_addr("xxx.xxx.xxx.xxx");

server_addr.sin_addr.s_addr = addr
```

getservbyname()

Get **SERVICE** by name

/etc/services

```
struct servent *service;
service = getservbyname("service", proto)
```

```
struct servent *service_entry;
int service_port;

service_entry = getservbyname("chargen", "tcp");
if(!service_entry) {
    perror("getservbyname() ");
    exit(2);
}
service_port = service_entry.s_port;
```

```
#
# Network services, Internet style
#
# $FreeBSD: src/etc/services,v 1.62.2.3 2000/10/05 07:37:37
# sheldonh Exp $
#       From: @(#)services      5.8 (Berkeley) 5/9/91
#
# WELL KNOWN PORT NUMBERS
#
rtmp      1/d4p  #Routing Table Maintenance Protocol
tcpmux    1/tcp  #TCP Port Service Multiplexer
tcpmux    1/udp  #TCP Port Service Multiplexer
nbp       2/d4p  #Name Binding Protocol
echo      4/d4p  #AppleTalk Echo Protocol
echo      7/tcp
echo      7/udp
discard   9/tcp  sink null
discard   9/udp  sink null
```

```
systat    11/tcp  users      #Active Users
systat    11/udp  users      #Active Users
daytime   13/tcp
daytime   13/udp
qotd      17/tcp      quote      #Quote of the Day
qotd      17/udp      quote      #Quote of the Day
chargen   19/tcp      ttytst     #Character Generator
chargen   19/udp      ttytst     #Character Generator
ftp-data  20/tcp      #File Transfer [Default Data]
ftp-data  20/udp      #File Transfer [Default Data]
ftp       21/tcp      #File Transfer [Control]
ftp       21/udp      #File Transfer [Control]
ssh       22/tcp      #Secure Shell Login
ssh       22/udp      #Secure Shell Login
telnet    23/tcp
telnet    23/udp
...
```

read / write

```
int num_bytes;  
num_bytes = read(socket, buffer, size);
```

```
int num_bytes;  
num_bytes = write(socket, buffer, size);
```

close()

```
int error;  
error = close(socket);
```