

Lezione 7 - 15 03 18

Thursday, March 15, 2018 10:30

#include <errno.h> per utilizzare la variabile globale sugli errori

es. if (errno = 97) fai ...

```
CONNECT(2)
NAME
    connect - initiate a connection on a socket
SYNOPSIS
    #include <sys/types.h>          /* See NOTES */
    #include <sys/socket.h>
    int connect(int sockfd, const struct sockaddr *addr,
                socklen_t addrlen);
```

Chiamata al sistema per connettersi al socket
riferito dal file descriptor sockfd
all'indirizzo specificato da addr.
addrlen size of 5

La connect deve funzionare per tutte le strutture dati dei protocolli

Quindi la connect prende l'indirizzo di un blocco di memoria e la sua lunghezza e
per capire il tipo di struttura si usa un identificatore come primo campo della
struttura.

La connect si adatta al tipo di dati.

```
struct sockaddr_in {
    sa_family_t    sin_family; /* address family: AF_INET */
    in_port_t      sin_port;   /* port in network byte order */
    struct in_addr sin_addr;    /* internet address */
};
```

Esempio di struttura per
AF_INET

SERVER → Ambiguo, macchina creata per fare da server, ma in realtà il
server è un programma.

Per questo ci servono due indirizzi, uno per la macchina (addr) e uno per
il programma server (port)

Questa scelta di port su macchine non è arbitraria

È stata definita una tabella con servizi di sistema associati a numeri standard
vi /etc/services per vederli tutti

es. http	80 / tcp	← transmission control protocol	STREAM
http	80 / udp	← user datagram protocol	MESSAGGI

Il livello di TRANSPORT deve sapere a che applicazione mandare i dati ⇒ lo si fa
attraverso port. ci siamo trovati tramite addr della macchina

Su port bisogna che sia scritto in network order per evitare inconsistenze tra sistemi little e big endian.

=> funzione di conversione htonl (htonl long)

Per caso girabyte (void *p, int d)

```
/* Internet address. */
struct in_addr {
    uint32_t s_addr; /* address in network byte order */
};
```

indirizzo IP => a.b.c.d 4 byte 0-255 ciascuna notazione decimale puntata

DOMAIN NAME SYSTEM struttura nomi -> indirizzo

nslookup [www.google.com] -> 216.58.213.228

((unsigned char *) &(indirizzo.s_addr))[0] = 216; e così via

In C tutti i passaggi a parametro è fatto by value (anche per strutture)

↳ by reference fatto tramite puntatori

netstat -> connessioni aperte

sprintf -> scrivere su stringa anziché su schermo