

`int socket(int domain, int type, int protocol);` 3 identificazioni

Non tutte le funzioni in C su Unix sono chiamate a sistema, ma tutte le chiamate a sistema hanno interfaccia C. Il manuale (es. `SOCKET(2)`) le distingue.

`socket()` crea un endpoint per comunicazione e restituisce un file descriptor che si riferisce a tale endpoint.

⇒ Quando legge e scrivo uso effettivamente metodi di come quelli in file.

Pero `socket` non apre un file, ma una comunicazione.

Devo fare

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

L'argomento `domain` specifica un dominio di comunicazione, seleziona la famiglia di protocolli usata per la comunicazione.

~~Personalizzazione:~~

```
unsigned char c;
```

```
uint8_t d;
```

Nella CPU sono distinti i numeri con o senza segno (complemento a 2)

Unsigned		signed	
0	↔	0	
⋮	↔	⋮	
127	↔	127	= 0 111 1111
128		-128	= 1 000 0000
⋮		⋮	
255		-1	= 111 11111

Senza unsigned a volte

un numero può essere

trattato come negativo

anche quando non lo

si vuole

**AF\_INET** : IPv4 Internet protocols  
↓  
Address family

ip(7)  
↳ capitolo 7 del manuale

**AF\_PACKET** : low level packet interface

packet(7)

Type di servizio:

**SOCK\_STREAM** → paradigma ereditato dai file (ho un canale aperto e "inietto" uno stream di byte) → non ho limiti se non fisici

**SOCK\_DGRAM** → messaggio "autosufficiente", ha dentro di sé tutto il necessario per arrivare a destinazione, ma ha lunghezza finita.

È il modello stesso che impone ai protocolli di essere affidabili ⇒ correzione errori

Nel servizio a messaggi non è facile gestire la perdita.

↳ o arriva tutto intero e subito, o non arriva

È sempre bene avere un riscontro dal ricevente (ricevuto e processato il messaggio)

```
1  #include <stdio.h>
2
3  int main(){
4
5      int i;
6      char c[10];
7      printf("Sono arrivato qui\n");
8      for (i=0; i<100000; i++)
9          c[i]=0xFF;
10     printf("Sono arrivato qua\n");
11
12 }
```

→ segmentation fault

flush : forza l'output di tutti i dati bufferizzati

Per applicazioni in cui servono reattività (es. tempo reali) meglio usare datagram.

Se invece devo usare file, meglio stream.

**SOCK\_SEQPACKET** : pacchetti sequenzializzati

Il protocollo specifica un particolare protocollo da usare con il socket.

Dato un singolo protocollo esiste per supportare un particolare tipo di socket per una data famiglia di protocolli, in questo caso protocollo può essere 0.

socket parte da 3  $\rightarrow$  0, 1, 2 sono standard. input  
standard. output  
standard. error  $\leftarrow$  è un output

apt-get install strace

Il file descriptor è l'indice della tabella con tutto ciò che mi serve per gestire la comunicazione

man 2 socket per gestione degli errori

↳ **errno** variabile globale viene settata con un codice che specifica l'ultimo errore (o successo)

```
1 #include <stdio.h>
2 #include <sys/socket.h>
3 #include <netinet/in.h>
4 #include <netinet/ip.h>
5
6 int main() {
7
8     int s;
9     s = socket(AF_INET, SOCK_STREAM, 0);
10    if (s == -1){
11
12        perror("SOCKET FALLITA"); //mette i : in automatico e una stringa di errore
13
14    }
15
16 }
```