

# Socket

Component name	Description	Utility	Parameters	Note	Tag
sockfd	descrittore di socket	identifica la socket		numero intero usato per le operazioni con la socket	#sockfd
struct sockaddr_in	struttura dati	usato per creare un indirizzo di socket per IPv4.	usato per local_addr e remote_addr	crea strutture dati quindi gli indirizzi sono strutture che contengono altri dati	#sockaddr_in
socklen_t	tipo intero	memorizza la lunghezza degli indirizzi del socket.			
socket(socket family, socket type, protocol);	crea una socket	restituisce il descrittore di socket sockfd	#protocolFamily #typeSocket #protocol	sockfd = socket(AF_INET, SOCK_DGRAM, 0);	#socket
PF_INET AF_INET	indica la famiglia di protocolli	usati in #socket			#protocolFamily
SOCK_DGRAM	indica il tipo di socket per UDP	usati in #socket			#typeSocket
SOCK_STREAM	indica il tipo di socket per TCP	usati in #socket			#typeSocket
0	specifica il protocollo da utilizzare in questo caso è automatico	usato in #socket			#protocol

Component name	Description	Utility	Parameters	Note	Tag
<code>memset(&amp;local_addr, 0, len);</code>	setta i byte del primo parametro con il secondo parametro, il numero di bit e deciso dal terzo paramtetro	inizializza a 0 l'indirizzo locale			

Component name: `sockfd`

- **Description:** Descrittore di socket.
- **Utility:** Identifica la socket.
- **Parameters:** Nessuno.
- **Note:** È un numero intero usato per le operazioni con la socket.
- **Tag:** `#sockfd`

Component name: `struct sockaddr_in`

- **Description:** *Crea strutture dati.*
- **Utility:** Usato per creare un indirizzo di socket per IPv4.
- **Parameters:** Usato per `#local_addr` e `#remote_addr`.
- **Structure attributes:**
  - `sin_family`: Famiglia di indirizzi (IPv4 o IPv6).
  - `sin_port`: Numero di porta del socket.
  - `sin_addr`: Indirizzo IP del socket.
- **Note:** Viene utilizzata come tipo di dato per gli indirizzi dei socket, ad esempio nell'associazione di un indirizzo locale a un socket tramite la funzione `bind()`. Crea strutture dati quindi gli indirizzi sono strutture che contengono altri attributi.
- **Tag:** `#sockaddr_in`

Component name: `socklen_t`

- **Description:** Tipo intero.
- **Utility:** Memorizza la lunghezza degli indirizzi del socket.
- **Parameters:** Nessuno.
- **Note:** -
- **Tag:** -

Component name: `socket()`

`socket(socket family, socket type, protocol)`

- **Description:** Crea un nuovo socket per la comunicazione di rete.
- **Utility:** Utilizzata per creare un nuovo socket che può essere utilizzato per la comunicazione di rete, come ad esempio TCP o UDP. Restituisce il descrittore di socket `sockfd`.

- **Parameters:**
  - #protocolFamily, #typeSocket, #protocol.
  - **domain:** Dominio del protocollo, che indica il tipo di comunicazione che verrà effettuata (ad esempio, **AF\_INET** per IPv4).
  - **type:** Tipo di socket da creare (ad esempio, **SOCK\_STREAM** per TCP o **SOCK\_DGRAM** per UDP).
  - **protocol:** Protocollo da utilizzare con il socket (0 per il protocollo predefinito, automatico).
- **Note:** Restituisce il descrittore del socket appena creato. Se il valore restituito è inferiore a 0, indica un errore nella creazione del socket.
- **Examples:**
  - `sockfd = socket(AF_INET, SOCK_DGRAM, 0);`.
  - `sockfd = socket(PF_INET, SOCK_STREAM, 0);`.
- **Tag:** #socket

### Component name: **PF\_INET / AF\_INET**

- **Description:** Indicano la famiglia di protocolli.
- **Utility:** Usati in #socket.
- **Parameters:** -
- **Note:** -
- **Tag:** #protocolFamily

### Component name: **SOCK\_DGRAM**

- **Description:** Indica il tipo di socket per UDP.
- **Utility:** Usati in #socket.
- **Parameters:** -
- **Note:** -
- **Tag:** #typeSocket

### Component name: **SOCK\_STREAM**

- **Description:** Indica il tipo di socket per TCP.
- **Utility:** Usati in #socket.
- **Parameters:** -
- **Note:** -
- **Tag:** #typeSocket

### Component name: **0**

- **Description:** Specifica il protocollo da utilizzare in questo caso è automatico.
- **Utility:** Usato in #socket.
- **Parameters:** -
- **Note:** -
- **Tag:** #protocol

### Component name: **memset(&local\_addr, 0, len);**

- **Description:** Setta i byte del primo parametro con il secondo parametro, il numero di bit è deciso dal terzo parametro.
- **Utility:** Inizializza a 0 l'indirizzo locale.
- **Parameters:**

- **Note:** -
- **Tag:** -
- 

### Component name: `local_addr.sin_family`

- **Description:** Indica il tipo di indirizzo utilizzato (IPv4 o IPv6).
- **Utility:** se imposta il campo `sin_family` della struttura `local_addr`
- **Parameters:** con `AF_INET`, che indica l'utilizzo di IPv4 e con `AF_INET6`, che indica l'utilizzo di IPv6.
- **Note:**
- **Tag:**

### Component name: `local_addr.sin_port`

- **Description:** Numero di porta del socket.
- **Utility:** Imposta il campo `sin_port` della struttura `local_addr`.
- **Parameters:** numero di porta con `htons()` che converte il numero della porta da host byte order a network byte order. Host To Network Short
- **Note:** la porta viene convertita da stringa a numero intero tramite `atoi(argv[1])`.
- **Tag:**

### Component name: `local_addr.sin_addr`

- **Description:** Indirizzo IP del socket.
- **Utility:** Imposta il campo `sin_addr` della struttura `local_addr`.
- **Parameters:** `htonl(INADDR_ANY)`, che imposta l'indirizzo IP a `INADDR_ANY`, il quale indica al socket di accettare connessioni da qualsiasi indirizzo IP.
- **Note:** `htonl()` converte l'indirizzo IP da host byte order a network byte order. Host To Network Long;
- **Tag:**

### Component name: `bind()`

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

- **Description:** Associa un indirizzo locale a un socket.
- **Utility:** Utilizzato per assegnare un indirizzo locale a un socket, come ad esempio un server che desidera accettare connessioni in ingresso su una determinata porta.
- **Parameters:**
  - `sockfd`: Descrittore del socket da associare all'indirizzo.
  - `(struct sockaddr *)&local_addr`: Puntatore alla struttura `local_addr` castata a un puntatore a `struct sockaddr`, poiché `bind()` richiede un puntatore a una struttura `sockaddr`.
  - `len`: Lunghezza dell'indirizzo del socket.
- **Note:** Valore restituito inferiore a 0, indica un errore nell'esecuzione della funzione `bind()`.
- **Example:** `bind(sockfd, (struct sockaddr *)&local_addr, len)`
- **Tag:** -

### Component name: `recvfrom()`

- **Description:** Riceve un messaggio da un socket, consentendo di specificare il mittente.
- **Utility:** Utilizzata per ricevere dati da un socket, consentendo di specificare il mittente tramite la struttura `remote_addr`.

- **Parameters:**
  - `sockfd`: Descrittore del socket.
  - `buffer`: Buffer in cui memorizzare i dati ricevuti.
  - `len`: Dimensione del buffer.
  - `flags`: Opzioni di ricezione.
  - `(struct sockaddr *)&remote_addr`: Puntatore a una struttura `sockaddr` utilizzato per memorizzare l'indirizzo e la porta del mittente.
  - `&len`: Puntatore alla lunghezza dell'indirizzo del mittente.
- **Note:** in poche parole ascolta il canale,
- **Example:** `recvfrom(sockfd, buffer, 99, 0, (struct sockaddr *)&remote_addr, &len);`
- **Tag:** -

Component name: `sendto()`

- **Description:** Invia dati tramite un socket a un indirizzo specifico.
- **Utility:** Utilizzata per inviare dati tramite un socket UDP al destinatario specificato dalla struttura `remote_addr`.
- **Parameters:**
  - `sockfd`: Descrittore del socket.
  - `sendline`: Buffer contenente i dati da inviare.
  - `strlen(sendline)`: Lunghezza dei dati da inviare.
  - `0`: Opzioni di invio.
  - `(struct sockaddr *)&remote_addr`: Puntatore a una struttura `sockaddr` che specifica l'indirizzo e la porta del destinatario.
  - `len`: Lunghezza dell'indirizzo del destinatario.
- **Note:** -
- **Tag:** -

Component name: `inet_ntoa()`

- **Description:** Converte un `indirizzo IPv4` da numerico a formato stringa. Network To Address.
- **Utility:** Utilizzata per convertire l'`indirizzo IP` memorizzato nella struttura `remote_addr` da formato numerico a formato stringa.
- **Parameters:** L'indirizzo IPv4 memorizzato nella struttura `remote_addr`.
- **Note:** Restituisce una stringa che rappresenta l'indirizzo IP in formato leggibile.
- **Tag:** -

Component name: `ntohs()`

- **Description:** Converte un `numero di porta` da network byte order a host byte order; Network To Host.
- **Utility:** Utilizzata per convertire il `numero di porta` memorizzato nella struttura `remote_addr` da network byte order a host byte order, formato leggibile.
- **Parameters:** Il numero di porta memorizzato nella struttura `remote_addr`.
- **Note:** Restituisce il numero di porta in host byte order, in formato leggibile.
- **Tag:** -

Component name: `close()`

- **Description:** Chiude un descrittore di file o un socket.

- **Utility:** Utilizzata per rilasciare le risorse associate a un descrittore di file o a un socket e interrompere la comunicazione attraverso il socket.
- **Parameters:**
  - `sockfd`: Descrittore del socket da chiudere.
- **Note:** Dopo aver chiuso un socket, non è più possibile utilizzarlo per inviare o ricevere dati.
- **Tag:** -

Component name: `inet_pton()`

- **Description:** Converte un indirizzo IP da formato testuale a formato binario.
- **Utility:** Utilizzata per convertire un indirizzo IP specificato in formato testuale (es. "192.168.1.1") in un formato binario appropriato per essere memorizzato nella struttura `sockaddr_in`.
- **Parameters:**
  - `AF_INET`: Famiglia di indirizzi IP (IPv4).
  - `argv[1]`: Stringa che rappresenta l'indirizzo IP in formato testuale.
  - `&(dest_addr.sin_addr)`: Puntatore al campo `sin_addr` della struttura `dest_addr` in cui verrà memorizzato l'indirizzo IP convertito.
- **Note:** Restituisce 1 in caso di successo e 0 se l'indirizzo non è valido o -1 in caso di errore.
- **Tag:** -

Component name: `inet_ntop()`

- **Description:** Converte un indirizzo IP da formato binario a formato testuale.
- **Utility:** Utilizzata per convertire un indirizzo IPv6 dalla sua rappresentazione binaria nella struttura `sockaddr_in6` in un formato di stringa leggibile.
- **Parameters:**
  - `AF_INET6`: Famiglia di indirizzi IP (IPv6).
  - `&(remote_addr.sin6_addr)`: Puntatore al campo `sin6_addr` della struttura `remote_addr`, contenente l'indirizzo IPv6.
  - `ipv6_addr`: Buffer in cui memorizzare l'indirizzo IPv6 in formato testuale.
  - `INET6_ADDRSTRLEN`: Lunghezza massima della stringa IPv6.
- **Note:** Restituisce un puntatore a `ipv6_addr` se la conversione ha successo, altrimenti restituisce `NULL`.
- **Tag:** -

Alessandro Ferrante