

Struttura per memorizzare le informazioni sul client

```
typedef struct {  
    struct sockaddr_in address; // Indirizzo del client  
    int sockfd;                // File descriptor della socket del client  
    int uid;                   // ID univoco del client  
} client_t;
```

Questa dichiarazione consente di utilizzare il tipo `client_t` direttamente senza dover scrivere `struct`.

Questa dichiarazione associa un alias `client_t` al tipo `struct client_t`, permettendoti di usare `client_t` direttamente.

```
struct client_t {  
    struct sockaddr_in address; // Indirizzo del client  
    int sockfd;                // File descriptor della socket del client  
    int uid;                   // ID univoco del client  
};
```

In questo caso, dovrai usare `struct client_t` ogni volta che fai riferimento a questa struttura.

WHILE(1)

Il ciclo `while (1)` è utilizzato per creare un loop infinito, che nel contesto di un server è comune per mantenere il server in esecuzione continua, accettando nuove connessioni e gestendo richieste fino a quando il server non viene esplicitamente terminato. Vediamo in dettaglio perché e come viene usato:

Perché Usare `while (1)` in un Server

1. **Accettazione Continua di Connessioni:** Un server di rete deve essere in grado di accettare connessioni da nuovi client in qualsiasi momento. Utilizzando un ciclo infinito, il server può rimanere in ascolto continuamente per nuove connessioni.
2. **Servizio Ininterrotto:** Un server spesso deve fornire un servizio continuo e ininterrotto. L'uso di `while (1)` garantisce che il server non termini dopo aver servito un solo client.
3. **Gestione Concorrenziale:** In un server multi-client, il ciclo infinito consente di accettare connessioni concorrenti, creare nuovi thread per ogni client, e continuare a funzionare senza interruzioni.