**WATCHNEIGHBORS - Assembler**



**Membri:**

* **Marco di Capua (matricola: 727673)**
* **Filippo Pelosi (matricola: 726602)**
* **Riccardo Zorzi (matricola: 726748)**

INDICE

1. Descrizione del progetto

B. Strutture dati

C. Descrizione del codice

0. Crea Profilo

1. Crea Rete Quartiere

2. Lista Segnalazioni

3. Aggiungi Segnalazione

4. Esci

D. Alcune scelte progettuali attuate

E. Fase di debugging

**A. DESCRIZIONE DEL PROGETTO**

Il progetto prevede l’implementazione di un software scritto in linguaggio Assembler, operante sul simulatore SPIM, che consenta all’utente di monitorare la sicurezza di una città suddivisa in quartieri.

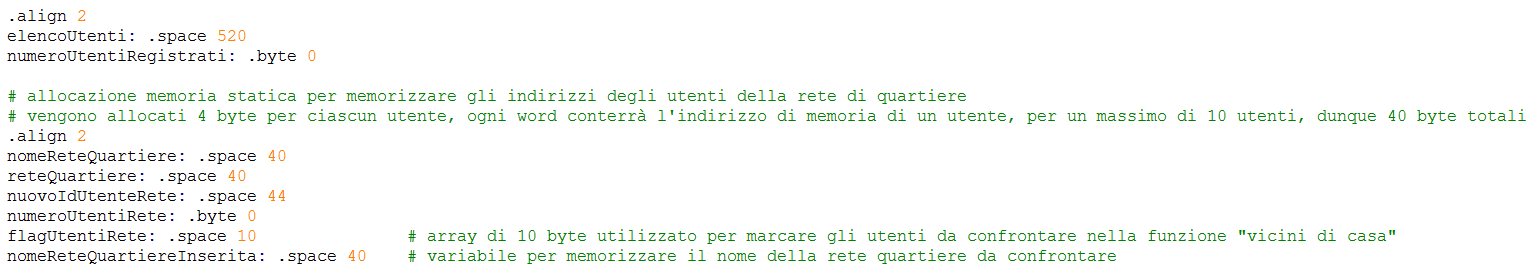
L’applicazione, nello specifico, è descritta dalle seguenti caratteristiche:

* Creazione di un nuovo profilo utente, specificando nome, cognome e coordinate per la geolocalizzazione;
* Creazione di una rete quartiere, la quale sarà costituita dagli utenti presenti in quel determinato quartiere;
* L’utente interagisce con l’applicazione visualizzando le segnalazioni attive in quel determinato momento ed è in grado di constatare se due utenti siano vicini di casa tramite la specifica funzione;
* Possibilità da parte dell’utente registrato di poter inserire una nuova segnalazione rendendola visibile agli altri utenti;
* Capacità da parte del software di gestire eccezioni generate durante la fase di input da parte dell’utente;
* Il programma deve essere in grado di arrestarsi attraverso la specifica funzione, per mezzo di un comando dell’utilizzatore.

**B. SCELTE PROGETTUALI**

Durante la fase di realizzazione del progetto abbiamo intrapreso scelte legate alle strutture dati al fine di gestire in maniera più consona i moduli che compongono l’applicazione.

In particolare, il programma deve essere in grado di gestire al più 10 profili utente contemporaneamente registrati, motivo per il quale la scelta è ricaduta su una struttura dati statica per allocare a priori lo spazio necessario.



Il frammento di codice mostra come viene allocata la memoria statica per memorizzare gli indirizzi degli utenti della rete quartiere.

Vengono allocati 4 byte per ciascun utente, ogni word contiene l’indirizzo di memoria di un determinato utente. Avendo un massimo di 10 utenti si avranno 40 byte totali.

Nello specifico sono 52 byte così distribuiti:

* 44 byte: nome utente (string);
* 4 byte: latitudine (float);
* 4 byte: longitudine (float);

Contrariamente ai profili utente, le segnalazioni non possono essere determinate a priori richiedendo quindi una struttura dati dinamica, la quale è caratterizzata da una variabile per memorizzare il primo oggetto della lista delle segnalazioni.

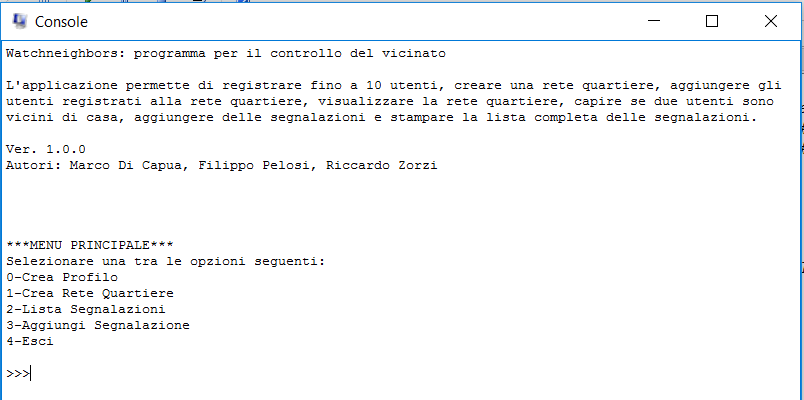


Una segnalazione è composta da 172 byte:

* 40 byte: nome e cognome;
* 24 byte: latitudine e longitudine della segnalazione;
* 24 byte: timestamp;
* 40 byte: motivo della segnalazione;
* 40 byte: stato della segnalazione;
* 4 byte: indirizzo della segnalazione successiva.

**C. DESCRIZIONE DEL CODICE**

L’utente avviando il programma, affronta il menù principale:

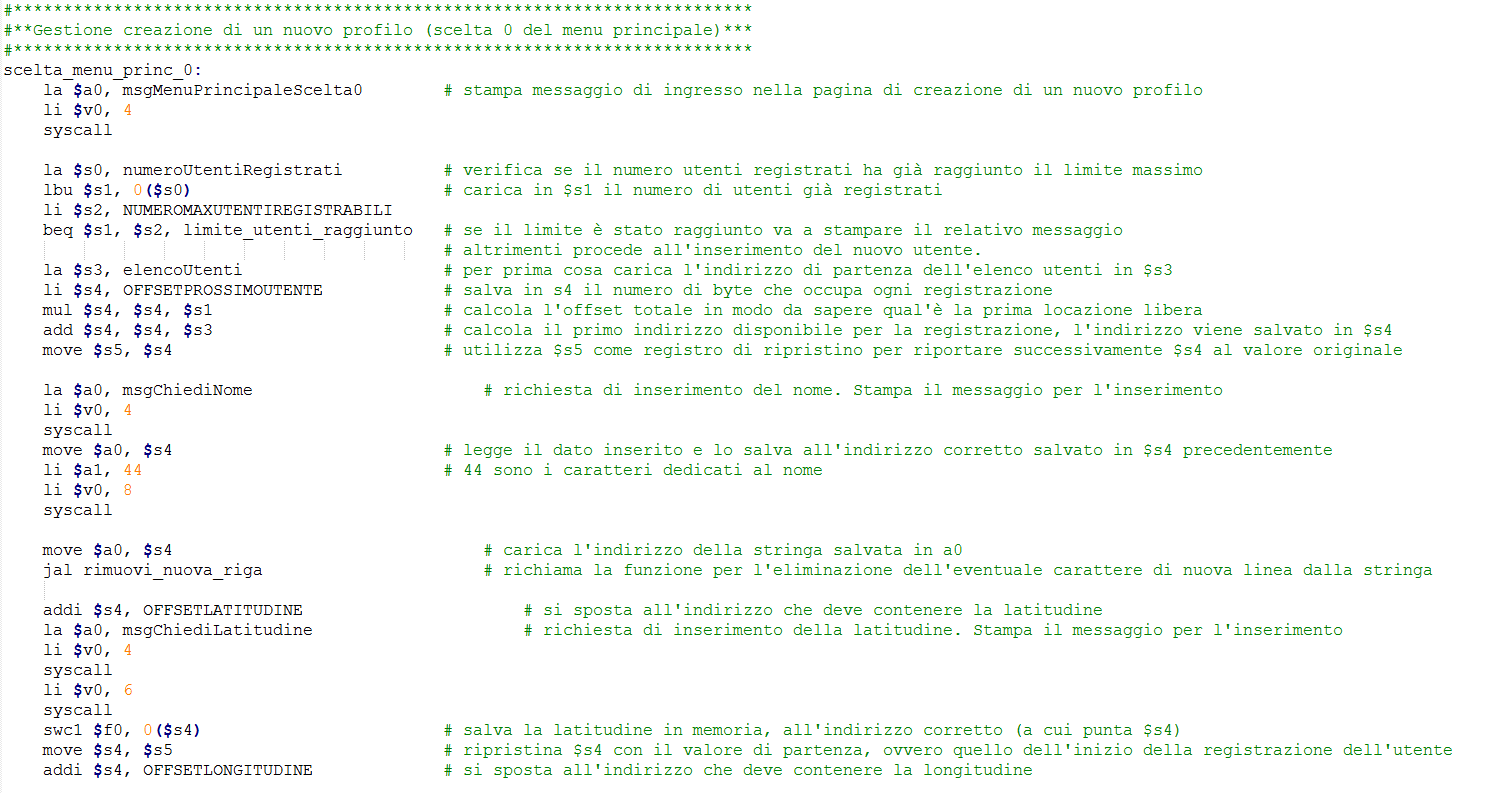


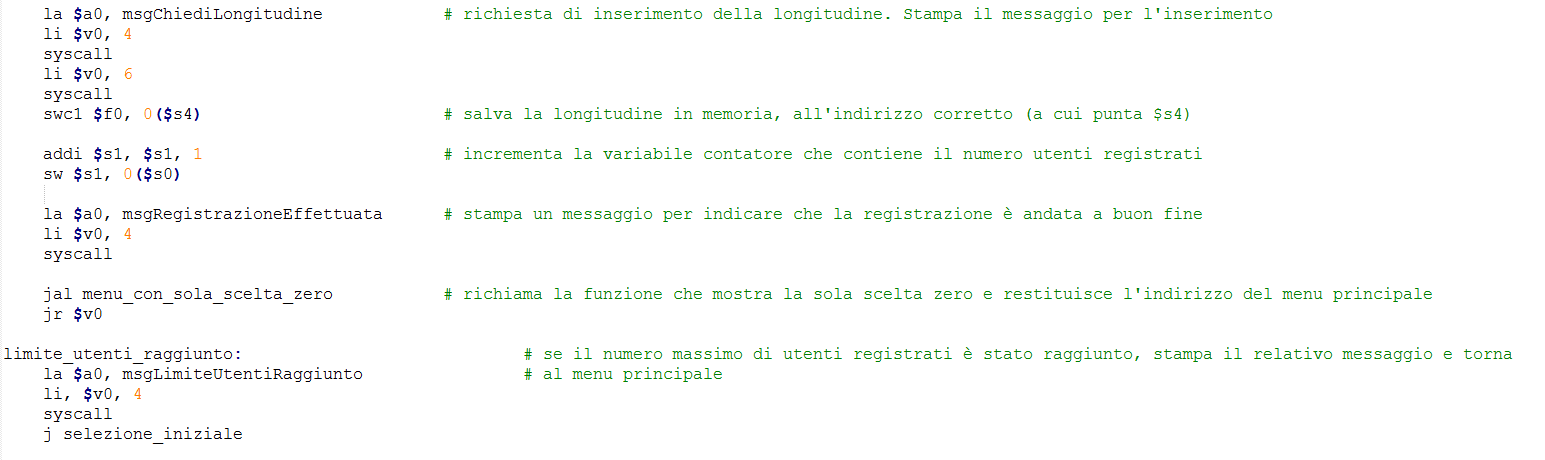
Di seguito vengono analizzate le possibili scelte effettuabili dall’utente in base alla scelta di quest’ultimo.

**0. CREA PROFILO**

La scelta comporta la creazione di un nuovo profilo utente, in seguito all’inserimento del nome e del cognome dell’utente.

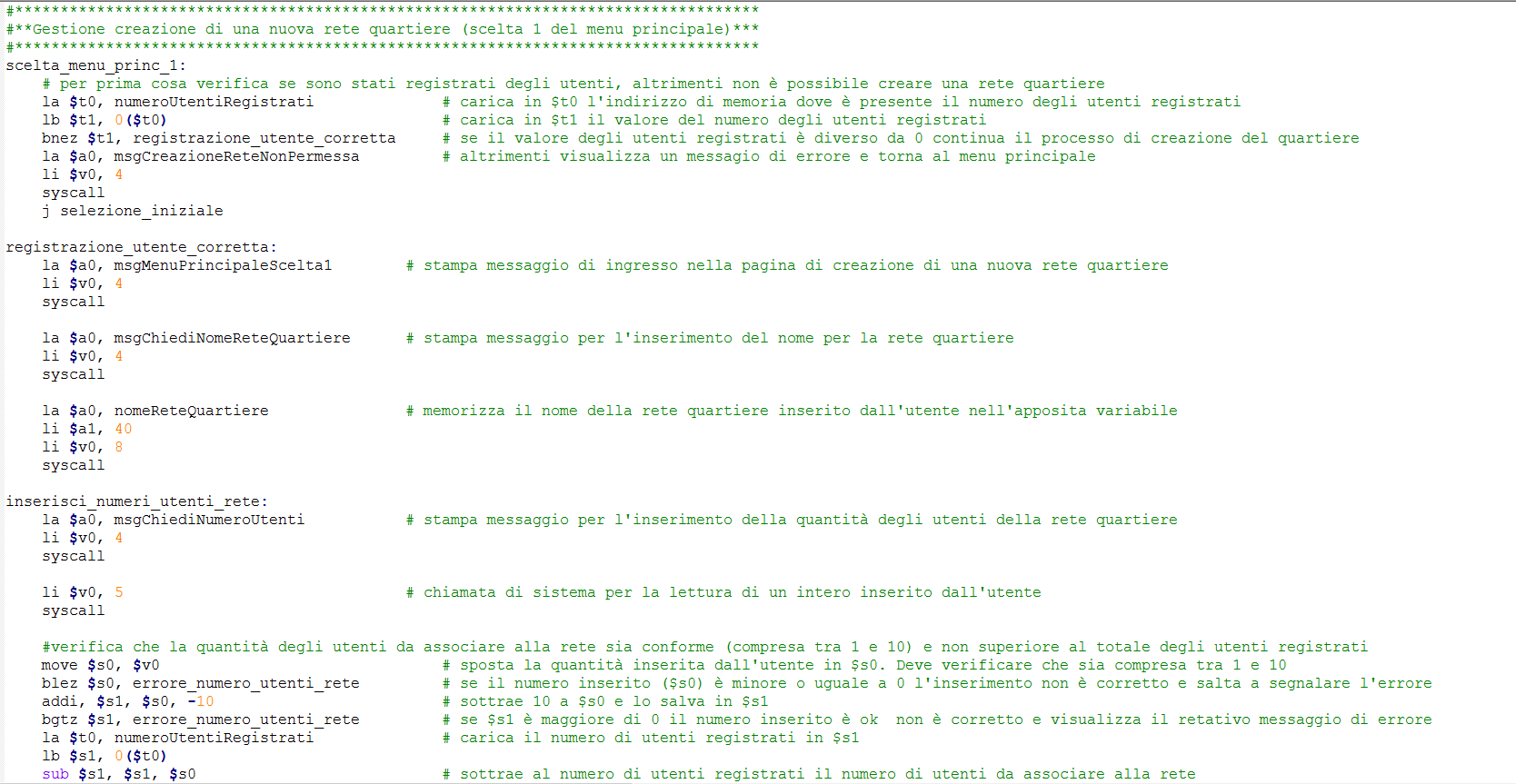
Viene gestita inoltre l’eccezione relativa alla creazione di un numero non possibile di utenti avendo il programma un massimo di 10 profili utente.

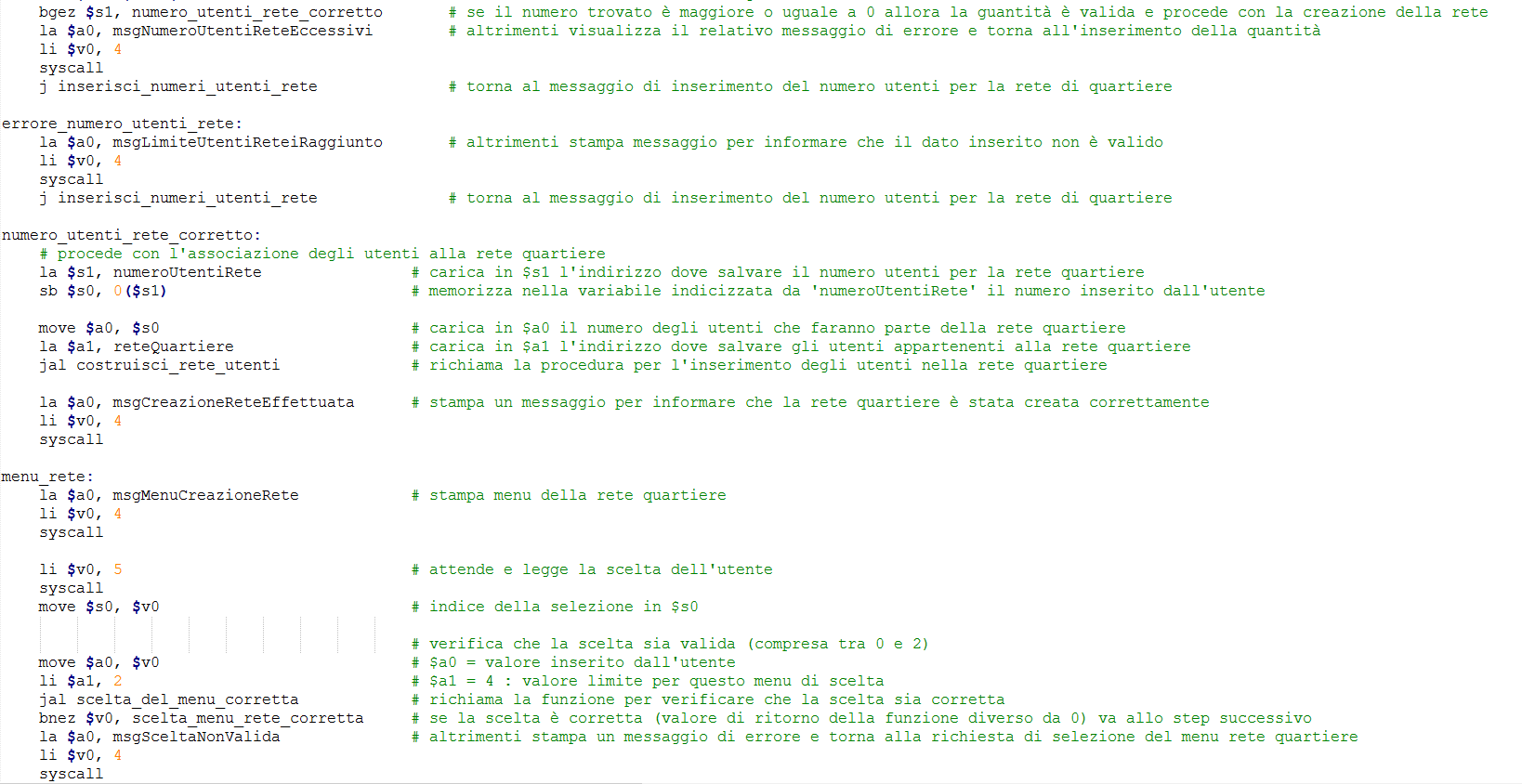


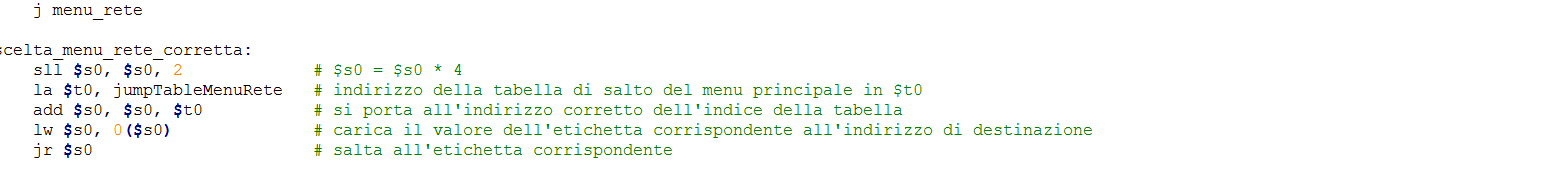


**1. CREA RETE QUARTIERE**

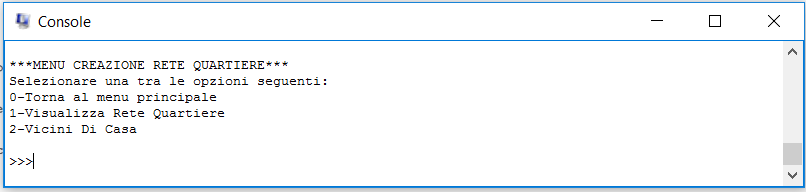
La funzione permette di creare una nuova rete quartiere a condizione che sia presente almeno un utente registrato da poter aggiungere alla rete, in caso contrario verrà stampato un messaggio d’errore:





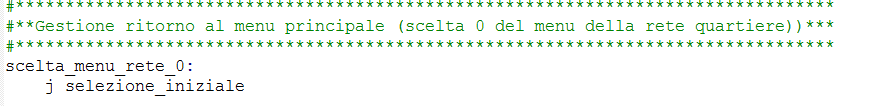


Dopo aver creato la rete quartiere, l’utente potrà interagire con il seguente menu di selezione:



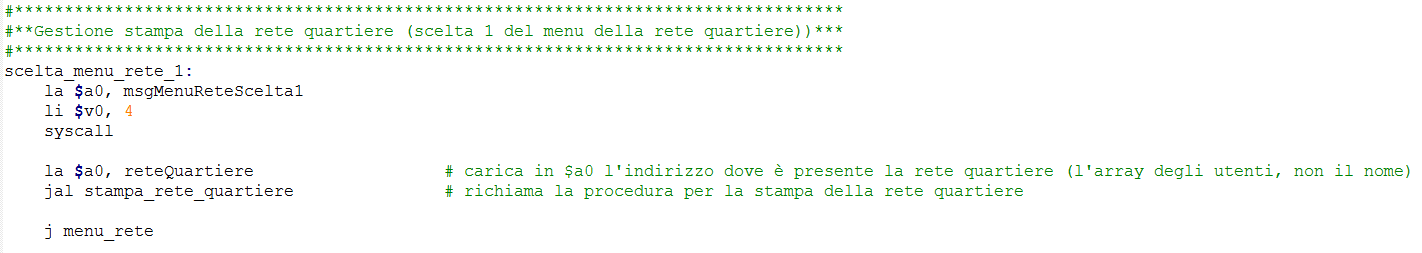
**0-Torna al menù principale**

Il ritorno al menu principale viene gestito con le seguenti istruzioni:



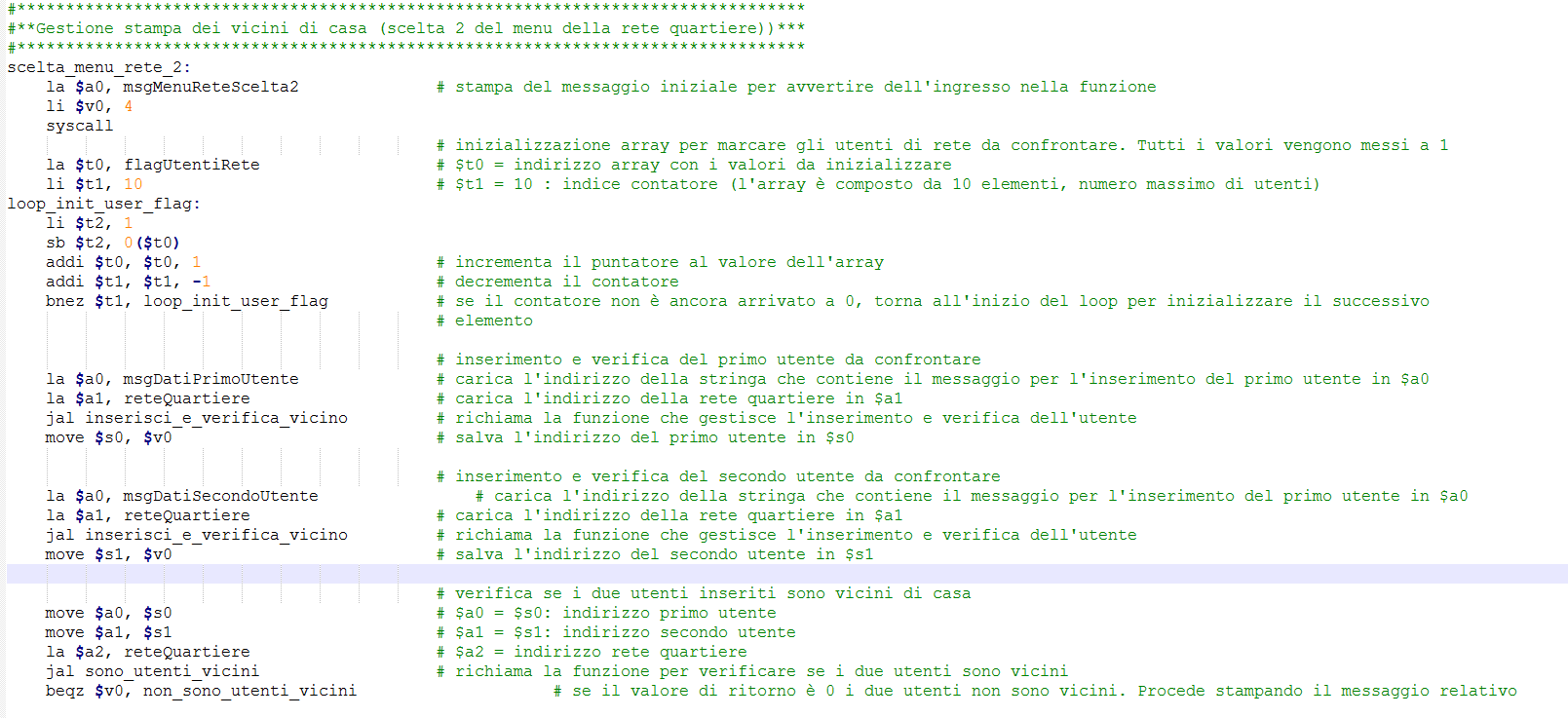
**1-Visualizza Rete Quartiere**

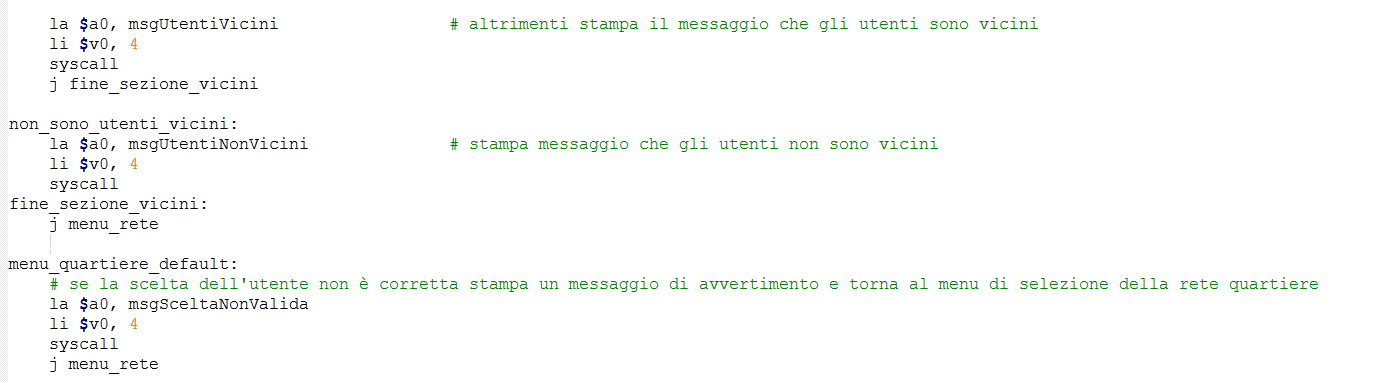
È possibile stampare le specifiche del quartiere creato, cioè le informazioni relative agli utenti che ne fanno parte con le relative coordinate per la geolocalizzazione:



**2-Vicini di casa:**

La seguente scelta utilizza una specifica funzione per determinare se due utenti sono vicini di casa; vengono inoltre gestite le eccezioni riguardo a scelte errate da parte dell’utente:



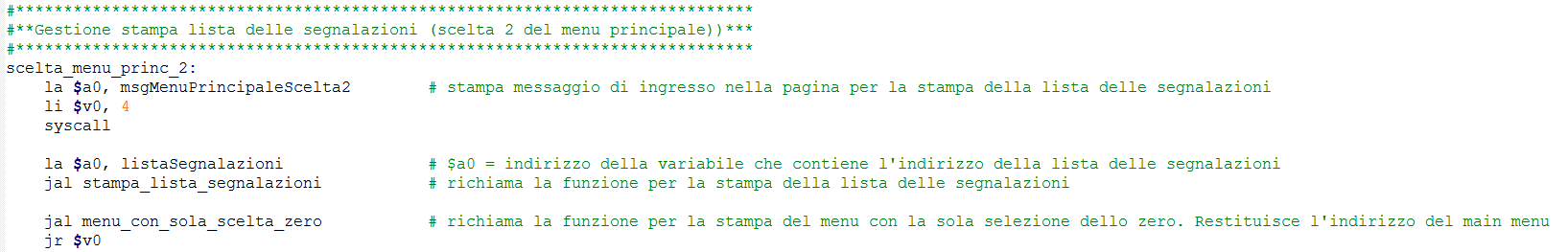


**2. LISTA SEGNALAZIONI**

La seguente opzione permette di visualizzare la lista delle segnalazioni attive, distinguendo in ciascuna i seguenti campi:

* Nome utente che ha registrato la segnalazione;
* Coordinate relative alla segnalazione;
* Data e ora della segnalazione;
* Motivo della segnalazione;
* Stato della segnalazione.

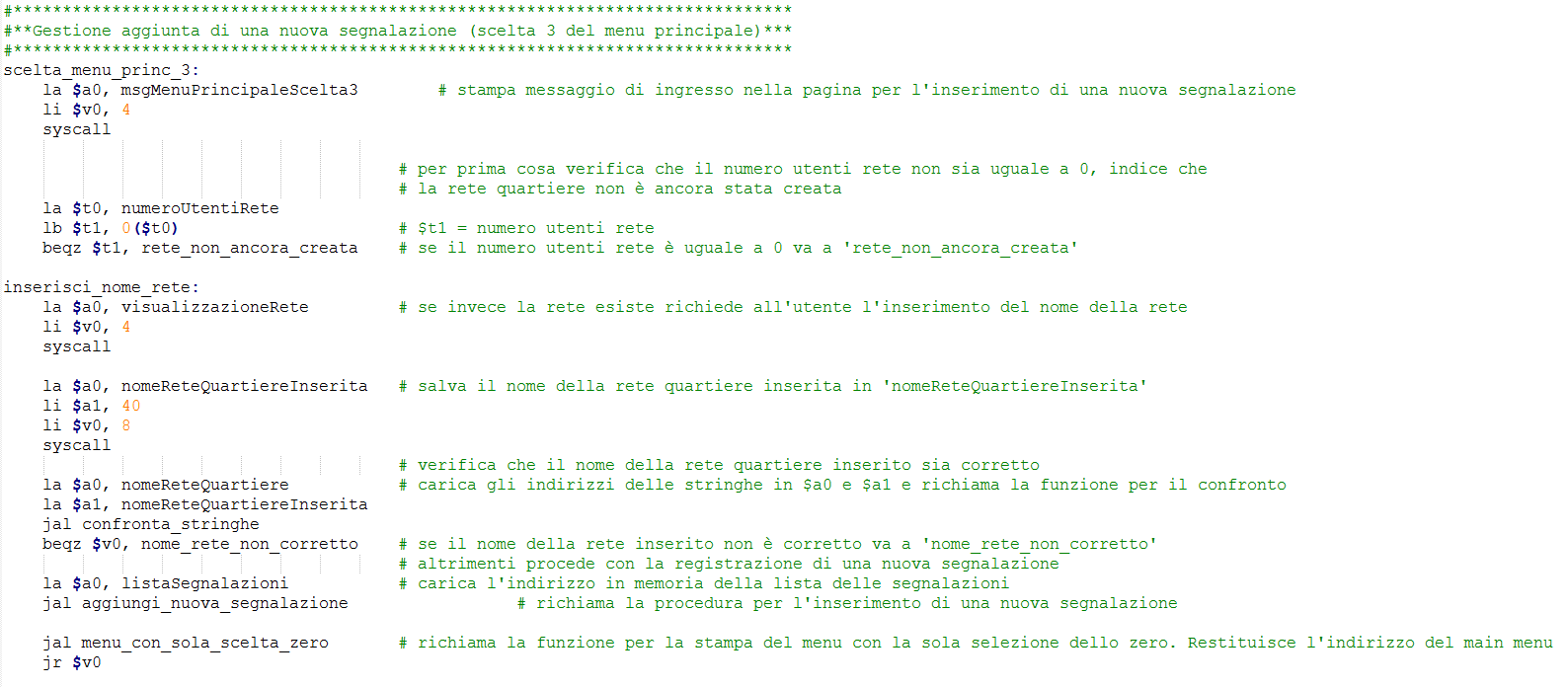
Di seguito il codice:

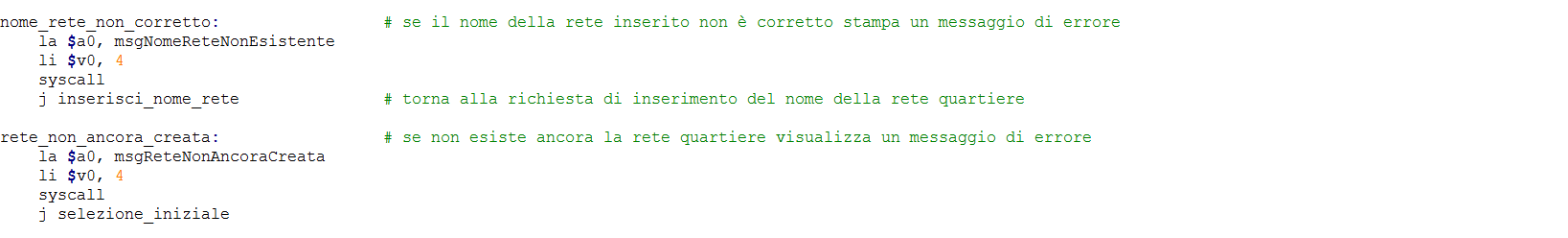


**3. AGGIUNGI UNA SEGNALAZIONE**

Con la seguente funzione è possibile registrare una nuova segnalazione, di modo che venga visualizzata dagli altri utenti;

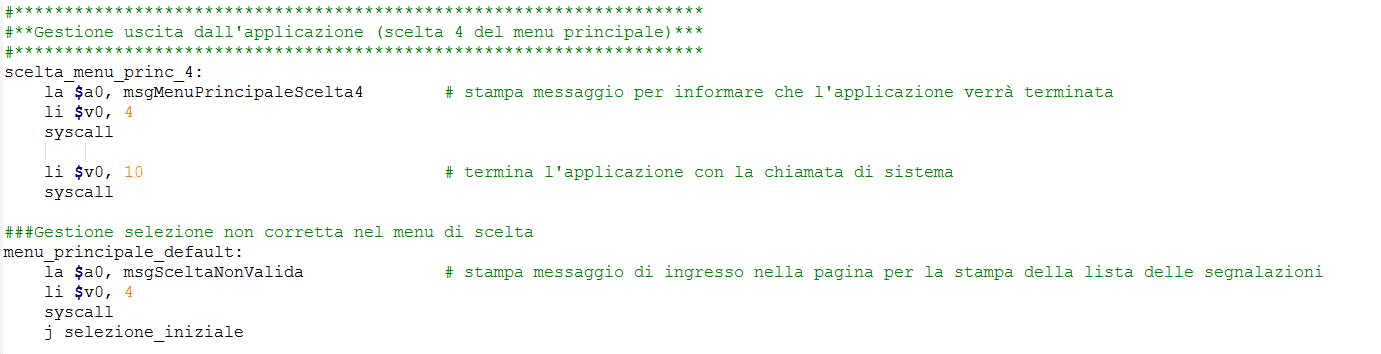
i campi da inserire sono uguali a quelli che verranno visualizzati dalla funzione precedente (Lista segnalazioni); viene inoltre gestita l’eccezione relativa al nome errato della rete quartiere o del quartiere non ancora creato.





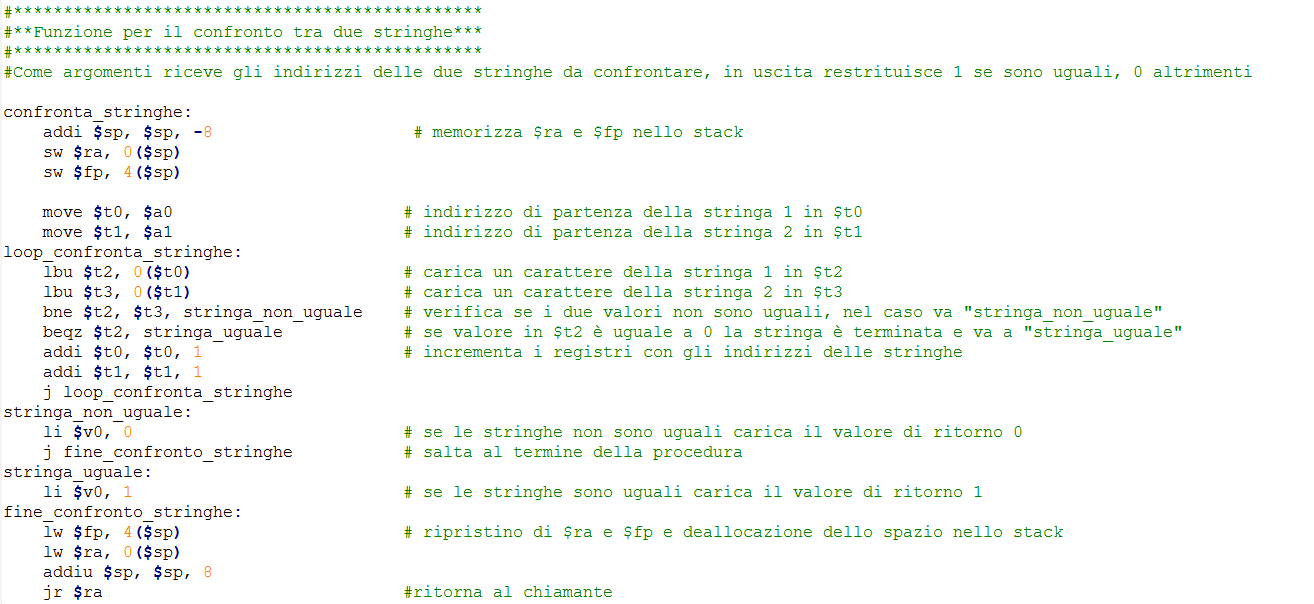
**4. ESCI**

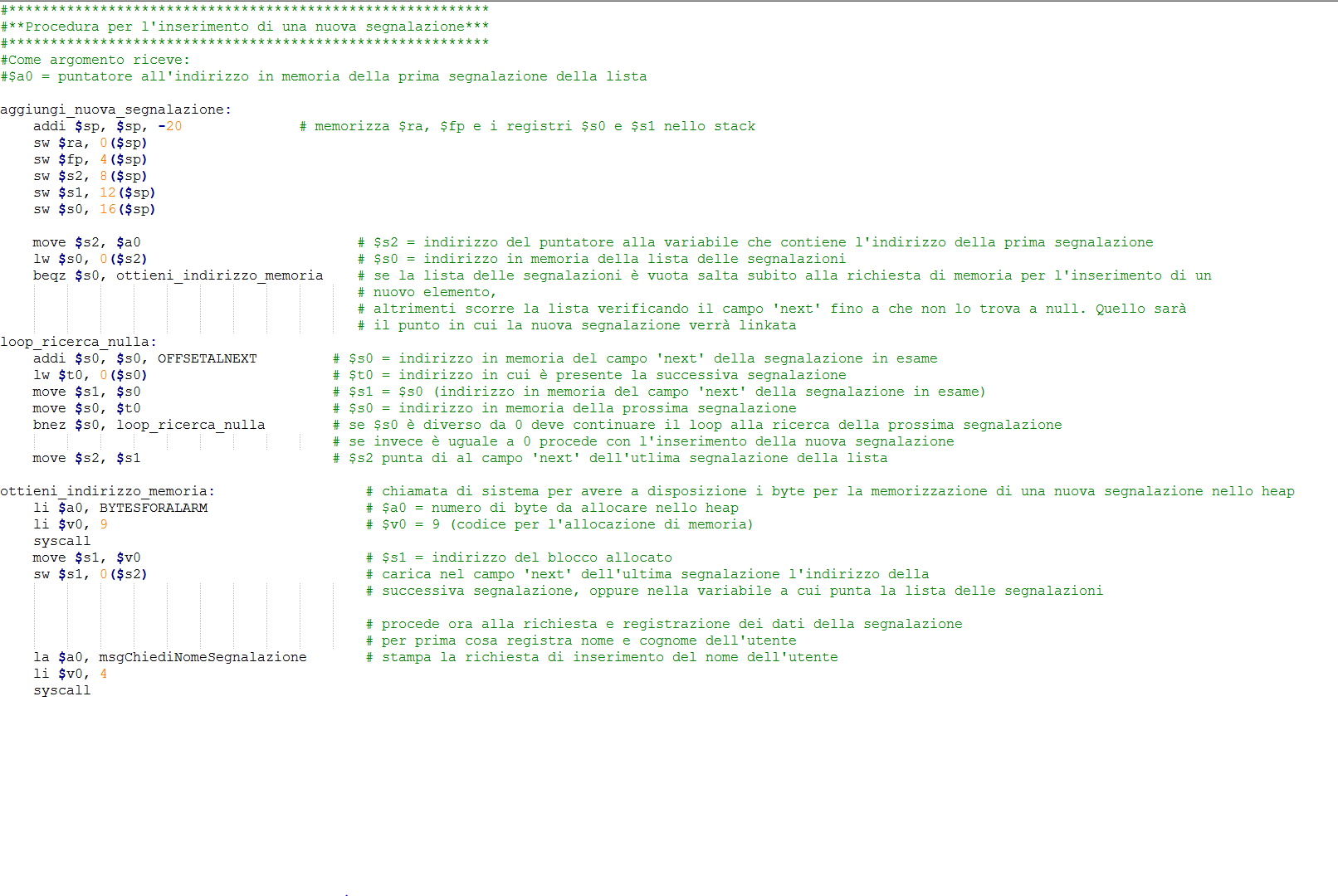
Il programma deve essere in grado di arrestarsi, ciò avviene grazie alla selezione di questa funzione.

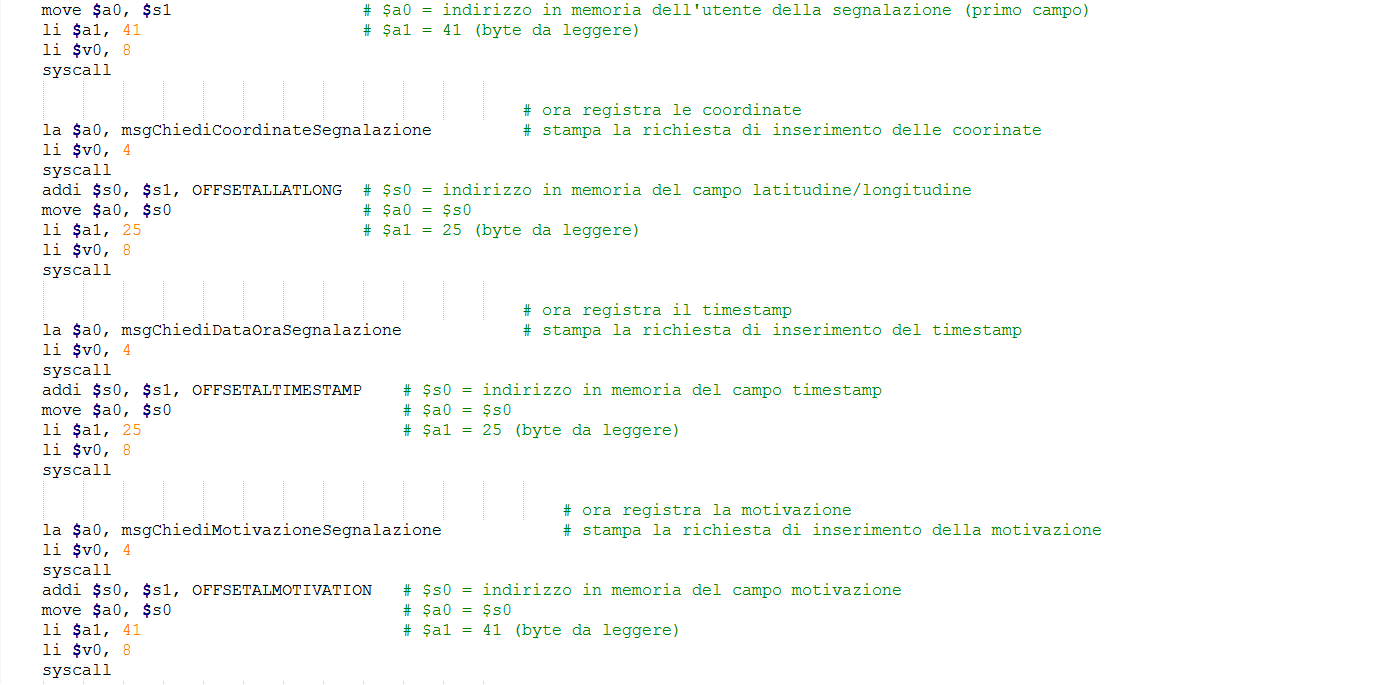


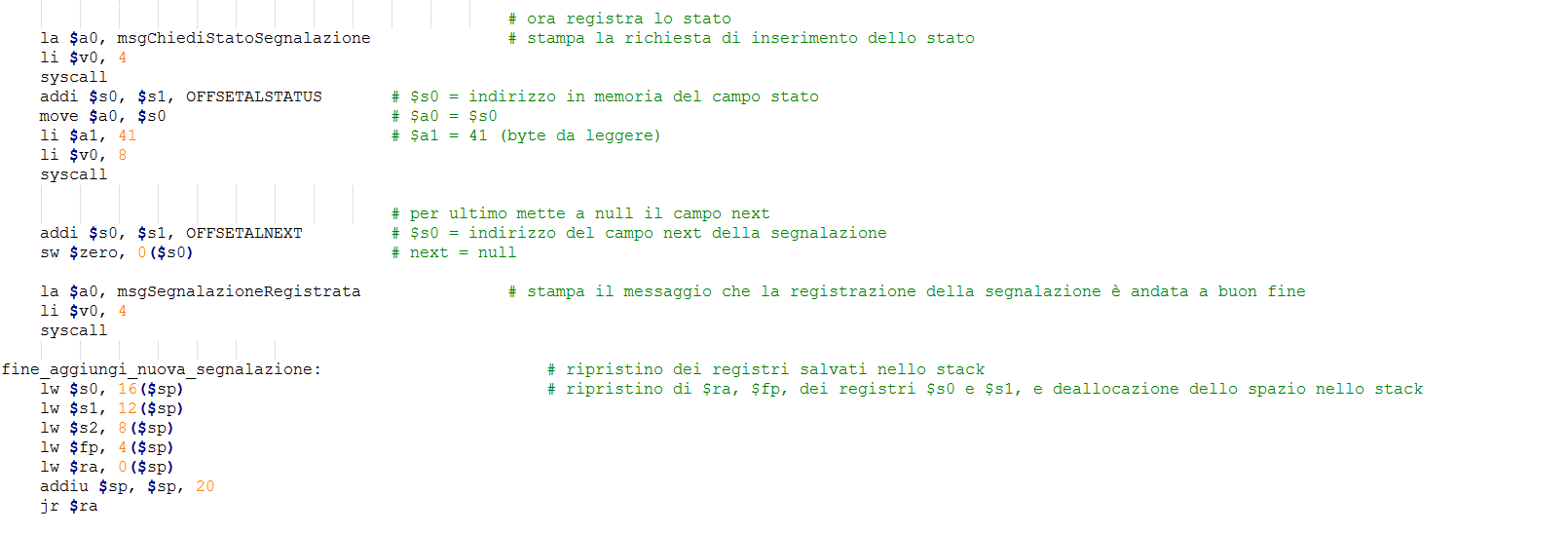
**D. SCELTE PROGETTUALI**

Abbiamo utilizzato la seguente funzione di confronto fra stringhe per controllare l’inserimento fatto dall’utente



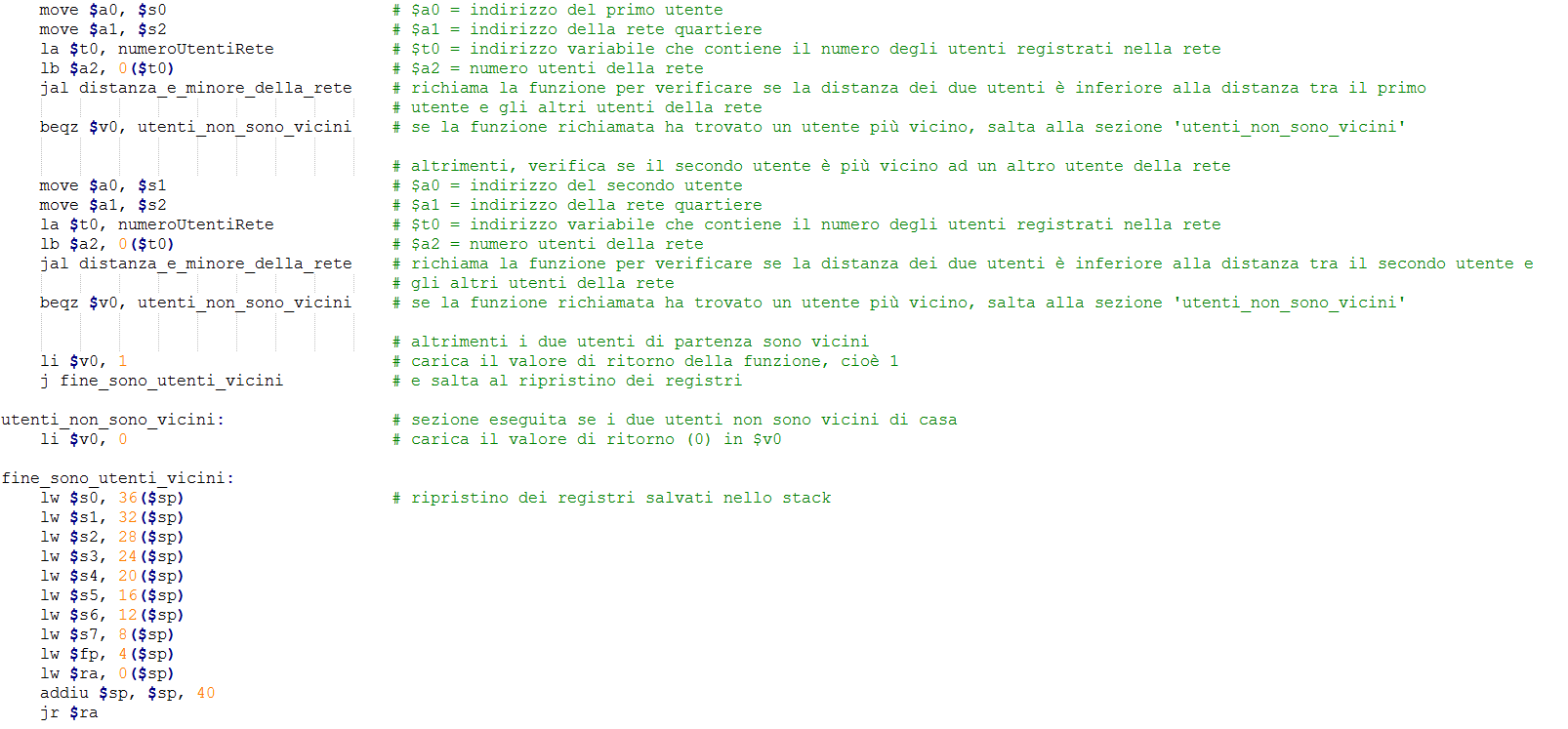
Per quanto riguarda l’aggiunta di una nuova segnalazione invece di seguito viene esposto il codice relativo: 





É presente inoltre una funzione per determinare se due utenti appartenenti a una stessa rete siano vicini:





Un’altra scelta progettuale è stata l’utilizzo di una jumptable (tabella dei salti) per i menù, la quale è una raccolta di indirizzi di codice destinati a essere indicizzati da un valore di selezione.

**E. DEBBUGGING**

Il programma è stato testato e non ha provocato nessuna incongruenza nei vari moduli analizzati.