

UNIVERSITA’ DEGLI STUDI DI NAPOLI

“FEDERICO II”

LABORATORIO SISTEMI OPERATIVI

Prof. CLEMENTE GALDI

PROGETTO: cORRIERE ESPRESSO





MARCO CARROZZO N86/1240

MAURIZIO DEL PRETE N86/783

# Guida d’uso

## Compilazione

La compilazione lato client dovrà essere effettuata nel seguente modo:

**gcc client.c library.c library\_client.c –o client**

Se il processo di compilazione è andato a buon fine il compilatore non invierà alcun messaggio all'utente.

**Screenshot: Compilazione Client :**

****

Dato che il server è un’applicazione multithread è necessario aggiungere alla compilazione il flag “-pthread”, quindi la compilazione lato server dovrà essere effettuata nel seguente modo:

**gcc –pthread server.c library.c library\_server.c –o server**

Se il processo di compilazione è andato a buon fine il compilatore non invierà alcun messaggio all'utente.

**Screenshot: Compilazione Server :**

****

E’ possibile eseguire la compilazione del Client e del Server attraverso l’utilizzo di un script shell “./compile” incluso nell’archivio.

**Screenshot: Compilazione Client e Server con script Shell:**

****

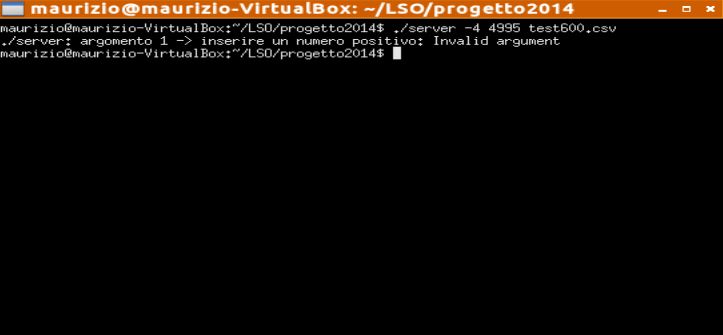
## Uso e Test del Server

Una volta compilato correttamente il Server è possibile eseguire il file di output ottenuto utilizzando la seguente sintassi:

“./server N K NOMEFILE

Dove N indica il numero massimo di operatori attivi, K il numero di oggetti che il server deve inviare al client e NOMEFILE il file sorgente per inizializzare il database del server.

**Screenshot: Inserimento parametri errati (N negativo):**



**Screenshot: Inserimento parametri errati (N char):**

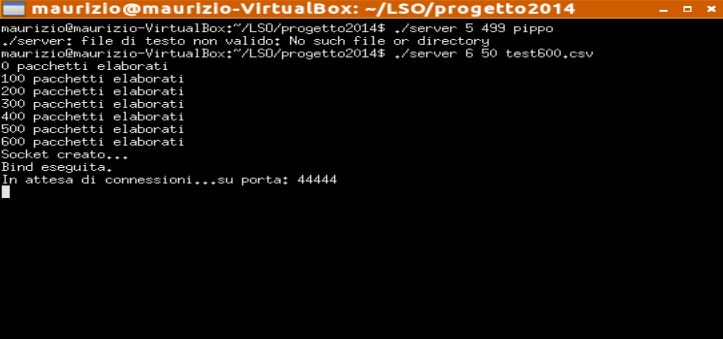


**Screenshot: Inserimento parametri errati (NOMEFILE errato):**



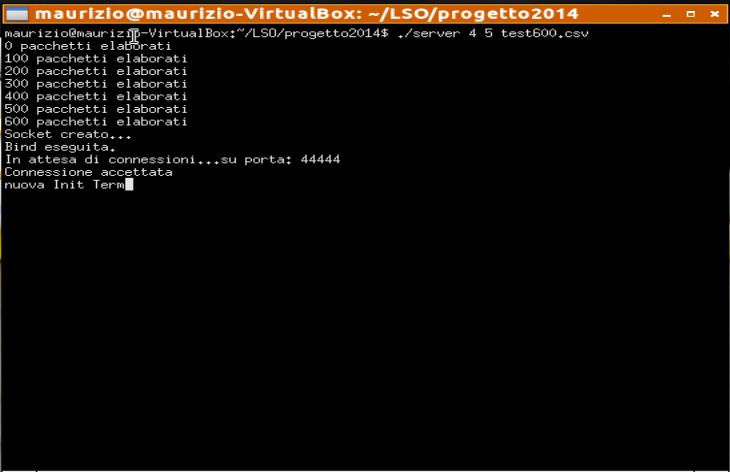
**Screenshot: Inserimento parametri corretti**

**(Inizializzazione database e creazione del soket Server):**

****

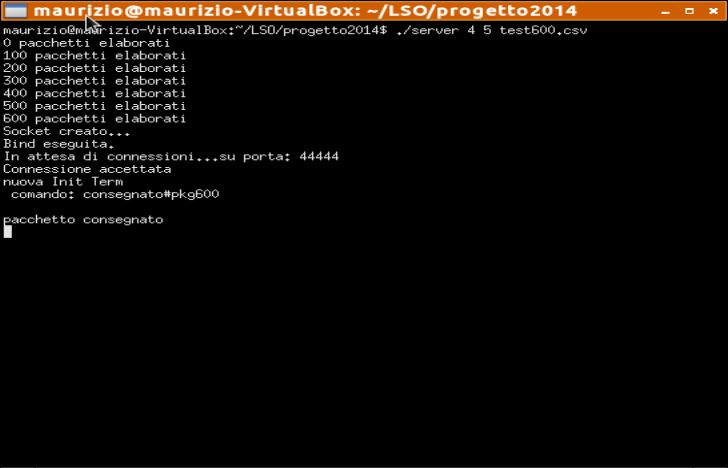
**Screenshot: Accetta connessione**

**(Se il numero di client attivi è inferiore al numero massimo fissato all’atto della creazione del server, allora si accetta una nuova connessione, e si esegue l’inizializzazione del database del client)**

****

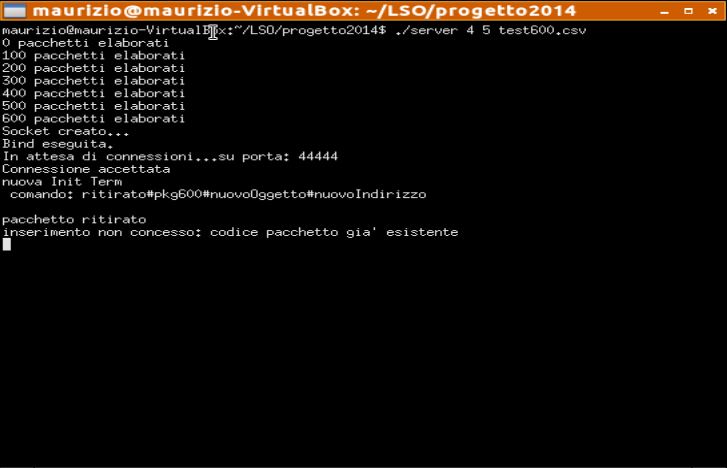
**Screenshot: Ricezione comando “consegnato”**

**(Il server modifica lo stato dell’oggetto identificato dal codice, da “in consegna” a “consegnato”):**

****

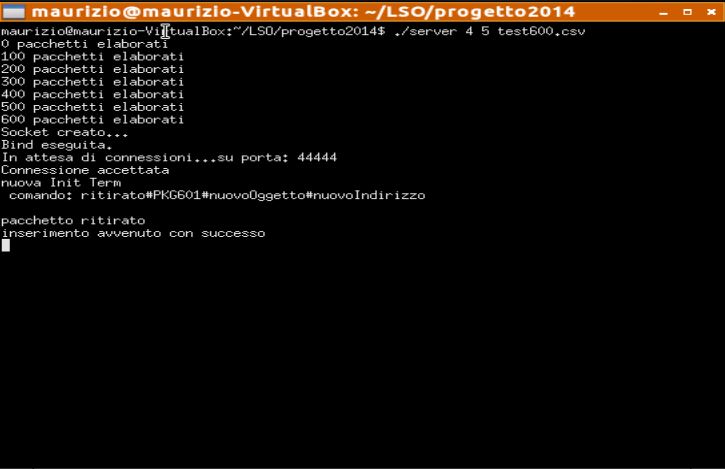
**Screenshot: Ricezione comando “ritirato” errato (codice prodotto già presente nel database )**

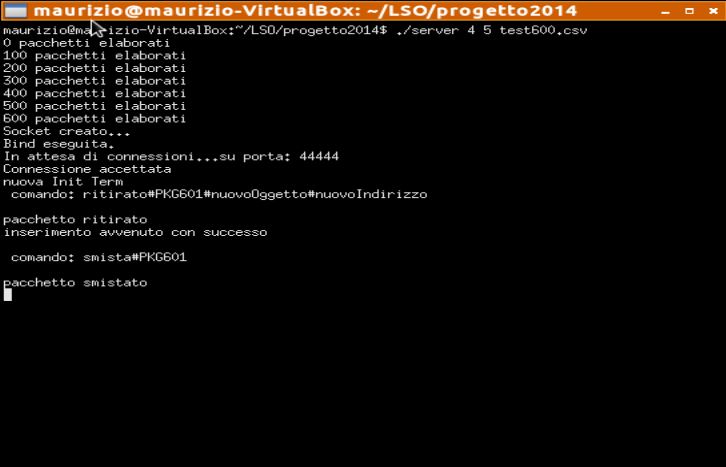
**(Il server esegue un controllo sul database per verificare che il codice del nuovo oggetto non sia già presente, se il codice è presente non esegue nessuna azione e ritorna al client un messaggio di errore):**

****

**Screenshot: Ricezione comando “ritirato” (codice prodotto non presente nel database del server)**

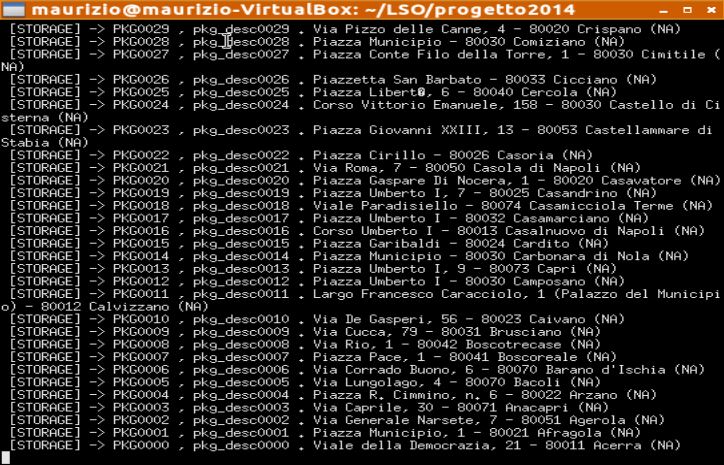
**(Il server aggiunge un nuovo oggetto nel suo database con stato “ritirato”):**

****

**Screenshot: Ricezione comando “smista”(Il server modifica lo stato dell’oggetto identificato dal codice, da “ritirato” a “in magazzino”):** 

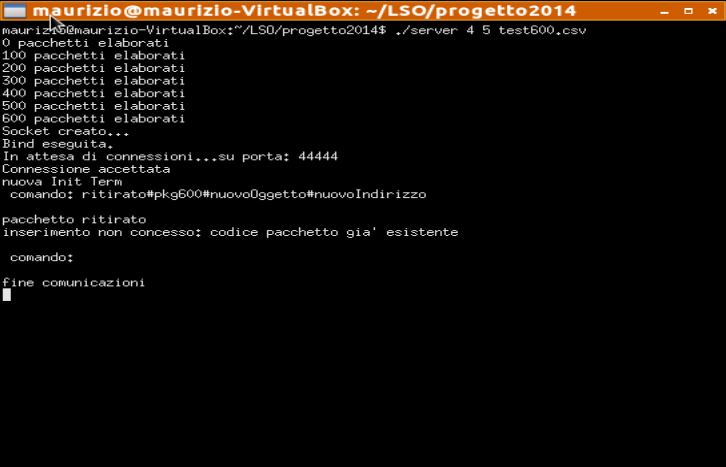
**Screenshot: Ricezione comando “elencaserver” :**

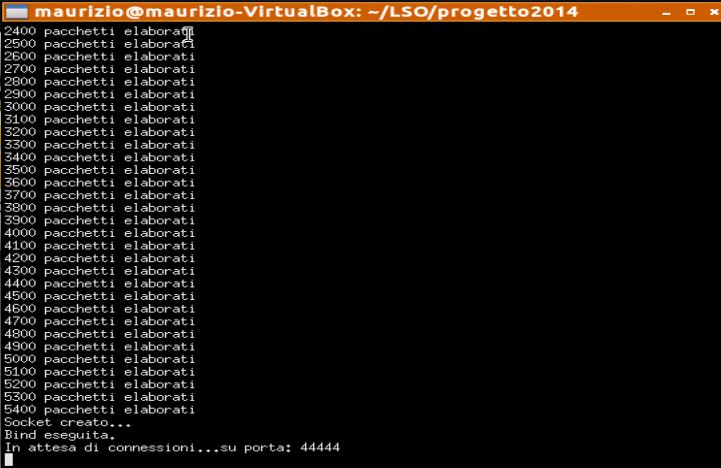
**(il server oltre ad invia al client l’intero elenco del database, stampa a video elenco):**

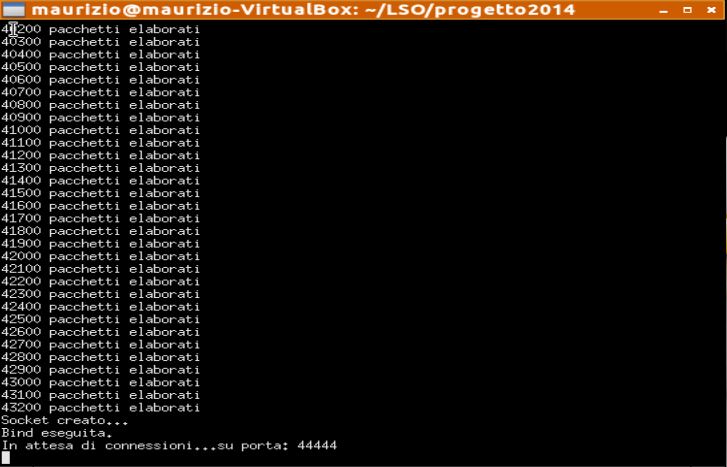
****

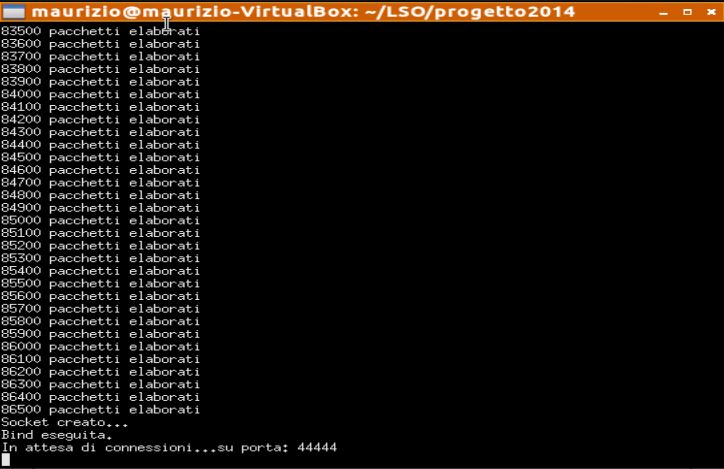
**Screenshot: Comunicazione Terminata con Client**

**(Il client termina quando non ha più oggetti nel suo database locale, di conseguenza si avrà la chiusura del canale di comunicazione):**

****

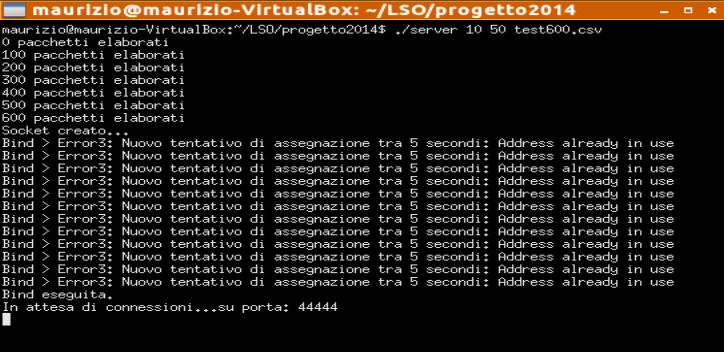
**Screenshot: Test database(caricamento da file di 5400 oggetti nel database del server) :**

**Screenshot: Test database (caricamento da file di 43240 oggetti nel database del server) :**

**Screenshot: Test database (caricamento da file di 86560 oggetti nel database del server) :**

**Screenshot: errore bind**

**(Quando la “Bind” ritorna errore causa “indirizzo già in uso”, attende 5 secondi e riprova fino alla riuscita):**

****

## Uso e Test del client

Una volta compilato correttamente il Client è possibile eseguire il file di output ottenuto utilizzando la seguente sintassi:

“./client Ip porta

Dove ip indica l’indirizzo ip dell’elaboratore e porta indica il numero di porta che individua il servizio sull’elaboratore .

**Screenshot: Inserimento parametri errati (valore porta negativo):**

**Screenshot: Inserimento parametri errati (valore porta non composto da soli digit):**



**Screenshot: Inserimento parametri errati (Ip non valido):**

****

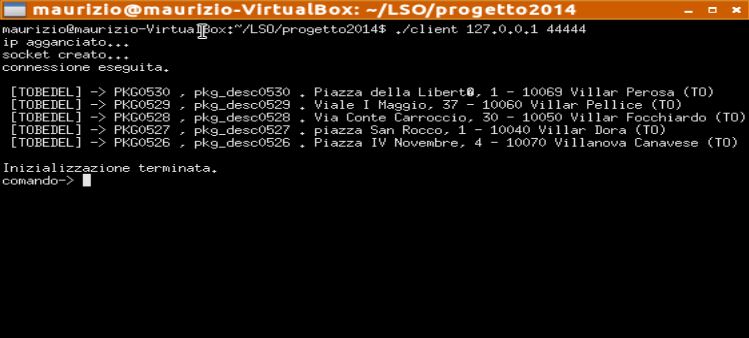
**Screenshot: Inserimento parametri corretto Ma numero massimo client superato**

**(il server accetta la nuova connessione solo quando qualche client già connesso termina, liberando un posto):**

****

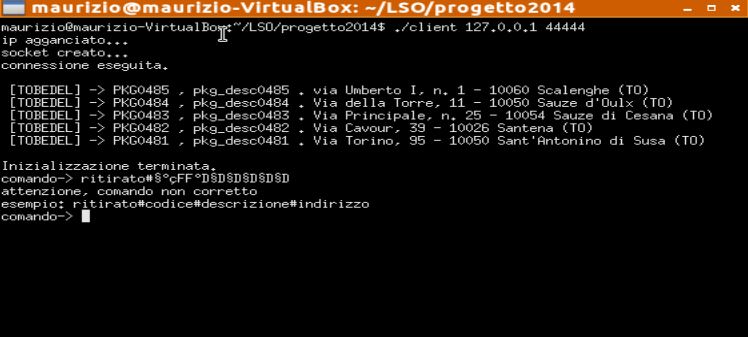
**Screenshot: Inserimento parametri corretti :**

**(Client creato ed inizializzato, attende l’inserimento dei comandi da parte dell’utente):**

****

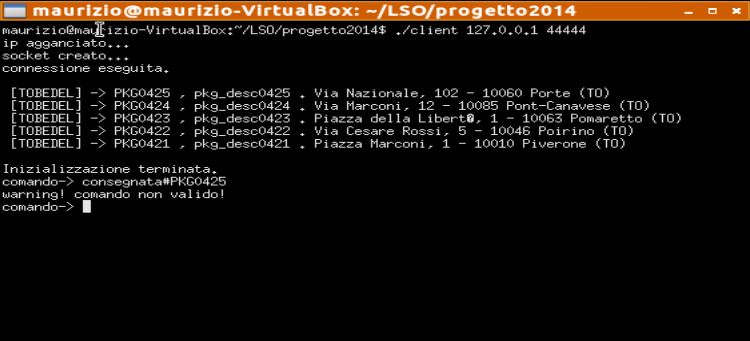
**Screenshot: comando non valido**

**(Il cliente esegue un primo controllo sulla struttura del comando inserito dall’utente, controllando che il numero dei campi sia corretto):**

****

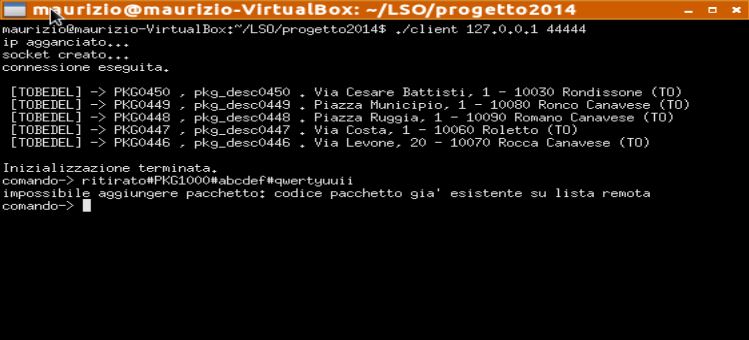
**Screenshot: comando non valido**

**(il client esegue un controllo sintattico del comando, attraverso la comparazione con una lista di comandi validi):**

****

**Screenshot: Comando “ritirato” (con codice già presente nel database del server)**

**(il client attende che il server controlli che il codice dell’oggetto ritirato non sia già presente nel suo database, e in caso di risposta negativa non esegue nessuna operazione):**

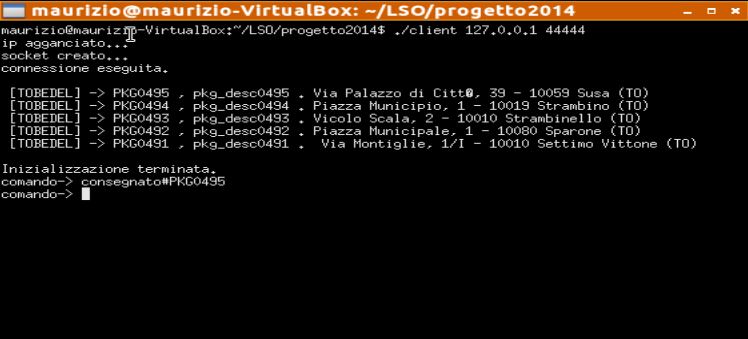


**Screenshot: Comando “ritirato” (con codice non presente nel database del server)**

**(il cliente aggiunge un nuovo aggetto con stato “ritirato” nel database locale);**

****

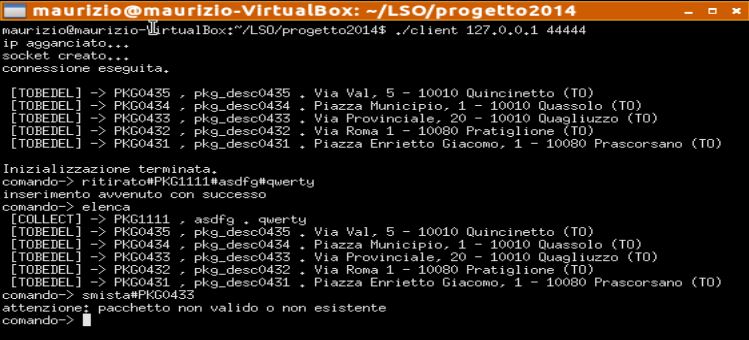
**Screenshot: Comando “consegnato”**

**(il client controlla che il codice dell’oggetto consegnato sia presente nel suo database locale, se è presente elimina l’oggetto):**

**Screenshot: Comando “consegnato”(di un oggetto non presente nella Lista Locale):**

**Screenshot: Comando “smista” ( oggetto non presente con stato ‘ritirato’)**

**(Il client controlla nel suo database locale che l’oggetto sia presente ed abbia lo stato “ritirato” ed solo in tal caso elimina l’oggetto presente)**



## Debugging

# Protocollo di comunicazione

# Dettagli implementativi

# Appendice