

Università di Roma Tor Vergata
Facoltà di Ingegneria
Sistemi Operativi - Prof. Francesco Quaglia

“**C:NEMA**, sistema di gestione di sale
cinematografiche scritto in C”
- *Relazione* -

Emanuele Savo
#0189363
emanuele.savo@gmail.com

Indice

Introduzione contenente le specifiche del progetto	5
Discussione delle scelte di progetto e realizzative e delle tecniche e metodologie generali usate	5
Il server	5
Le strutture	7
Sviluppi futuri e migliorie	7
Il client	7
Breve manuale d'uso dei programmi (come compilare, come installare)	8
Compilazione e rimozione	8
Esecuzione del server	8
Esecuzione del client	9
Licenza del software	9

Introduzione contenente le specifiche del progetto

Il progetto richiedeva la realizzazione di un servizio di prenotazione posti per una sala cinematografica, richiedendo esplicitamente la realizzazione sia dell'applicazione client che di quella server.

Ciascun posto è caratterizzato da un numero di fila, un numero di poltrona, e può essere libero o occupato. Il server accetta e processa sequenzialmente o in concorrenza (a scelta) le richieste di prenotazione di posti dei client (residenti, in generale, su macchine diverse).

Un client deve fornire ad un utente le seguenti funzioni:

1. Visualizzare la mappa dei posti in modo da individuare quelli ancora disponibili.
2. Inviare al server l'elenco dei posti che si intende prenotare (ciascun posto da prenotare viene ancora identificato tramite numero di fila e numero di poltrona).
3. Attendere dal server la conferma di effettuata prenotazione ed un codice univoco di prenotazione.
4. Disdire una prenotazione per cui si possiede un codice.

Discussione delle scelte di progetto e realizzative, e delle tecniche e metodologie generali usate

Nonostante fosse richiesta anche un'applicazione client il server è stato sviluppato in modo da essere utilizzato trivialmente anche attraverso netcat - non fornendo tuttavia sufficienti protezioni nei confronti di input errati od incompleti. Utilizzando una connessione di tipo TCP, infatti, non è necessario controllare la completezza e l'ordine del traffico dati quanto la loro correttezza sintattica.

Il server

L'applicazione server è stata inizialmente sviluppata in modalità sequenziale single-thread ma con in mente un'architettura facilmente adattabile all'approccio multithread.

Dopo aver inizializzato le strutture relative al networking viene chiesta all'utente la dimensione della sala cinematografica e viene inizializzata una struttura di tipo cinema.

Non appena ha impostato l'ascolto sulla porta desiderata il server accetta la connessione ed il processo interpreta le stringhe in ingresso tramite la socket come comandi (la funzione analizza esclusivamente il primo carattere poiché il controllo si può implementare più facilmente lato client). In particolare i comandi implementati sono i seguenti:

lI	Stampa a schermo le prenotazioni distinguendo i posti riservati alla prenotazione di codice I (nel caso viene omessa mostra posti liberi ed occupati)
gI	Risponde 0 se il codice I è disponibile od 1 se è utilizzato
i	Risponde con 0 se non sono più disponibili prenotazioni o con il primo codice disponibile
y	Risponde con 0 se non sono più disponibili prenotazioni o con il primo codice disponibile, riservandolo
rI,R1,C1,R2,C2[...].	Riserva i posti Rk,Ck (Riga, Colonna) al codice I. Qualora alcuni posti non siano disponibili ne riserva altri
fI,N	Riserva i primi N posti disponibili al codice I
cI,R1,C1,R2,C2[...].	Cancella i posti Rk, Ck dalla prenotazione di codice I, controllando eventualmente se la prenotazione rimane vuota
uI,N	Cancella N posti alla prenotazione di codice I
dI	Cancella l'intera prenotazione di codice I
h	Stampa un messaggio riassuntivo di questi comandi (utilizzabile attraverso netcat)
z	Cancella eventuali prenotazioni vuote
x	Chiude la connessione con il server

Il passaggio al multithread è stato relativamente semplice, poiché è stato necessario implementare esclusivamente una mutex per sequenzializzare l'accesso alla struttura condivisa cinema; una ulteriore mutex è stata utilizzata per limitare il numero massimo di thread tramite un contatore comune.

Le strutture

Fondamentale è la struttura cinema, formata da due unsigned int indicanti le dimensioni della sala e due array monodimensionali delle dimensioni righe*colonne e righe*colonne+1. Questi ultimi due, allocati in memoria mediante calloc (che si occupa di impostare la memoria allocata a zero) indicano rispettivamente i posti della sala e i codici prenotazione disponibili. Grazie a calloc non è stato necessario inizializzare con dei cicli for le due aree di memoria: il valore 0 infatti indica rispettivamente

- * Un posto libero
- * Un codice prenotazione inutilizzato.

Nonostante la sala sia bidimensionale il vettore che la gestisce è monodimensionale poiché si può accedere alle informazioni riguardanti il posto utilizzando l'indice

$$i = \text{Riga} * (\text{Dimensione Colonne}) + \text{colonne}.$$

La struttura dedicata ad ogni thread contiene le componenti indispensabili, ovvero i puntatori alla struttura cinema relativa, al file descriptor del client correlato, al thread id ed ai mutex.

Sviluppi futuri e migliorie

Una specifica aggiuntiva facilmente implementabile consiste nel gestire un multisala, potendo istanziare più di una struttura di tipo cinema e ponendo più thread in ascolto di più porte, o chiedendo alla connessione a quale sala interfacciarsi.

Altre specifiche implementabili consistono nella possibilità di salvare periodicamente lo stato della sala su un file di testo in modo da poter recuperare le prenotazioni in caso di crash del server, l'autenticazione tramite password ad una prenotazione oppure alla gestione della sala da parte di un "amministratore"

Il client

Il client si limita a guidare l'utente nell'eseguire suddetti comandi attraverso alcuni menù guidati ed un controllo di input ed output.

Breve manuale d'uso dei programmi (come compilare, come installare)

Compilazione e rimozione

Il codice sorgente dei programmi è accompagnato da un Makefile che include le seguenti opzioni

make	help	Stampa le opzioni disponibili
	client	Compila il client
	server	Compila il server
	All	Compila entrambi gli applicativi
	clean	Elimina eventuali applicativi compilati

In caso l'utility GNU Make non sia installata è possibile compilare il client con

```
gcc -o client client*
```

ed il server con

```
gcc -o server server* -lpthread
```

ed ovviamente è possibile eliminare gli applicativi con i comandi `rm client` e `rm server`.

Esecuzione del server

Il server si esegue mediante il comando `./server` ed accetta come parametri:

-h	help	Messaggio di aiuto
-p	porta	La porta TCP sulla quale porsi in ascolto.
-r	righe	Dimensioni della sala cinematografica (-l crea una sala quadrata)
-c	colonne	
-s	lato	

Nel caso tale porta non sia specificata si assume di default il valore 4321. Qualora la porta specificata non sia disponibile il server prova sequenzialmente tutto il range di porte 1-65535, ponendosi in ascolto sulla prima porta disponibile (o terminando qualora non vi siano porte disponibili).

Qualora vi siano dimensioni mancanti il server le richiede a schermo.

Esecuzione del client

Il client si esegue mediante il comando `./client`, specificando tramite i parametri

<code>-a</code>	<code>indirizzo</code>	L'indirizzo del server
<code>-p</code>	<code>porta</code>	La porta del server

Al posto del client si può utilizzare il tool di sistema netcat mediante il comando

```
nc <indirizzo_del_server> <porta>
```

(ricordando che via netcat c'è il comando di aiuto rapido "h").

Licenza del Software

```
<c:nema, a simple TCP cinema reservation manager>  
Copyright (C) <2019> <Emanuele Savo>
```

```
This program is free software: you can redistribute it and/or modify it under the  
terms of the GNU General Public License as published by the Free Software  
Foundation, either version 3 of the License, or (at your option) any later  
version.
```

```
This program is distributed in the hope that it will be useful, but WITHOUT ANY  
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE. See the GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License along with this  
program. If not, see <https://www.gnu.org/licenses/>.
```