UNIVERSITÀ DI PISA

Corso di laurea in informatica

Relazione progetto

Laboratorio di Sistemi Operativi

Autore:

Filippo Renai 530478

Anno Accademico 2020/2021

1 Supermercato

La funzione principale del supermercato è la funzione **main** i cui compiti principali sono: settare la configurazione, predisporrere la cattura dei segnali,predisporrere la connessione con il direttore, inviare il pid del suo processo al direttore e riceverlo, creare ed aspettare il thread responsabile delle casse ed infine comunicare al direttore la chiusura.

La funzione inizio del responsabile delle casse è la funzione **responsabileCasse** il cui compito è di gestisce tutti gli aspetti delle casse: creare e aspettare i thread cassa, creare l'array degli status delle casse che sarà inviato al direttore, di modo che possa prendere una decisione corretta, comunicare con il direttore la situazione delle casse, in base alle tempistiche di ogni cassa ed attraverso la sua risposta aprire o chiudere casse. Infine creare e aspettare il thread responsabile dei clienti.

La funzione inizio del thread cassa è la funzione **cassiere** il cui compito è simulare le operazioni di una cassa, dal servire i clienti, attraverso l'utilizzo della struct **coda**, all'attendere quando è chiusa. Al momento della ricezione del segnale da parte del supermercato si comporta in due casi diversi: nel caso **SIGQUIT** si arresta per la variabile sigquit, nel caso **SIGHUP** si arresta grazie alla variabile atomica varCass. Infine quando la cassa ha finito scrive nel file log le proprie statistiche.

La funzione inizio del responsabile dei clienti è la funzione **responsabileClienti** il cui compito è creare e aspettare i thread clienti, ed attivarne degli altri quando è possibile, essendo l'entrata al supermercato contingentata.

La funzione inizio del thread cliente è la funzione **cliente** il cui compito è simulare le operazioni del cliente. Nel caso speciale in cui il cliente non ha acquistato prodotti deve informare il direttore ed aspettare l'autorizzazione di uscita attraverso un invio del segnale **SIGUSR1**, altrimenti entra nella prima cassa disponibile, in maniera randomica, ed aspetta il suo turno. Durante l'attesa, in base all'algoritmo, puo' scegliere, se possibile, una cassa più conveniente,

mentre nel caso in cui la cassa sia stata chiusa sceglie la prima cassa disponibile, in maniera randomica; entrambe le problematiche sono gestite mediante un goto che semplicemente torna indientro al momento della scelta di una cassa. Infine il cliente esce in due casi: è stato servito e quindi scrive le sue statiche nel file log, altrimenti nel caso della chiusura immediata nel supermercato attraverso il segnale **SIGQUIT**, senza essere servito, semplicemente non scrive nulla nel file log.

Infine ricordo la funzione **sighandler** che gestisce la cattura di segnale inviato dal direttore.

2 Direttore

La funzione principale del supermercato è la funzione **main** i cui compiti principali sono: settare la configurazione, predisporrere la cattura dei segnali,predisporrere la connessione e comunicazione con il supermercato,nel caso del **test2** deve lanciare il processo supermercato. Attraverso la situazione delle casse mandatagli dal supermercato deciderà se aprire e chiudere casse comunicandogli la risposta.La terminazione avviene sotto segnalazione,via socket,del supermercato nel main.

Il direttore gestisce i segnali inviatogli dal Makefile attraverso la funzione static **sighandler**, nel momento che intercetta un segnale invia a sua volta lo stesso segnale al supermercato attraverso il suo pid precedentemente inviatogli.

3 Comunicazione supermercato direttore

La comunicazione avviene attraverso una singola connessione socket ed utilizzo del segnale **SIGUSR1**.

3.1 Socket

Inizialmente i reciproci main si scambieranno il loro pid, in seguito il **responsabileCasse** comunicherà, periodicamente, la situazione delle casse attraverso un prima writen contenente il codice **CODCASSA**, settato nel file .h ed una seconda writen contenente l'array status casse. Infine il main del supermercato per chiudere il direttore userà una sola writen contenente il codice **CODCHIUSURA**.

3.2 Segnale

Il segnale è utilizzato nel caso in cui il cliente non ha acquistato prodotti e chiedere al direttore il permesso di uscire. La richiesta avviene mediante l'invio del segnale SIGUSR1, per risposta il direttore rimanda il segnale di modo che il cliente capisca che ha ottenuto il permesso per uscire.

4 Libreria

Ho creato un libreria statica, tenendo irrilevante la scelta tra dinamica e statica in questo caso, con il suo file header e il suo file c.

Nel file **mylib.h** troviamo: tutte le include utilizzate, tutte le define utilizzate in particolare, quelle utili per la comunicazione tra supermercato e direttore,le varie struct e prototipi di funzione. La define **CLIENTIMAX** è utilizzata soltanto per evitare un fileLog con troppi clienti.

Nel file **mylib.c** troviamo l'implementazione dei prototipi delle funzione, descritte nel file .h, riguadanti la configurazione, le operazioni della coda che sono thread-safe,la readn e la writen thread-safe usate nelle comunicazioni tra direttore e supermercato attraverso la socket ed infine la funzione utile per la questione del tempo.

5 Scelte Implementative

5.1 Concorrenza

Entrambi i processi sono **thread safe** attraverso l'uso di lock, variabili di condizioni, variabili atomiche e alle due funzioni readn e writen della libreria che vanno a trattare il caso speciale errn uguale a **EINTR** (in questo caso non è un errore vero e proprio).

Nel processo **supermercato** ho utilizzato le lock: **fileLogLock** per il problema di concorrenza riguardante il file log, la lock **casseStatusLock** per la variabili condivise casseStatus ed infine la lock **uscitaZeroProdottiLock** utilizzata per far si che le richieste di uscita dei clienti con zero prodotti siano diversificate. Le variabili atomiche vengono utilizzate per snellire il progetto da eccessive lock e sono utilizzate: nel segnare le statische nel file log, nelle condizioni di richiesta di comunicazione delle casse,nel far entrare nuovi clienti e nel caso **SIGHUP** per comunicare alle casse e il loro responsabile che possono chiudere.

Nel processo **direttore** non ho problemi di concorrenza non essendo multi-thread.

5.2 Interazione supermercato direttore

La scelta di utilizzare una sola socket tra il supermercato e il direttore è per velocizzare l'intero progetto. Il responsabileCasse comunica con il direttore sotto solecitazione di ogni singola cassa, mentre il main comunica il pid e la chiusura. Infine attraverso l'uso dei segnale, nei casi dei clienti con zero acquisti, ci evitiamo la comunicazione socket.

6 Lanciare il progetto

Il **Makefile** viene utilizzato per:

- Generare gli eseguibili del programma: **make**

- Eliminazione dei file generati: **make clean**
- Lanciare il primo test: **make test1**
- Lanciare il secondo test: **make test2**

Inserendo i seguenti comandi in grassetto nella shell in base a cosa si voglia fare.