

Membri : Fuqiang Guo

Il progetto è composto da 3 programmi

per compilare : make taxigame WIDTH=<VAL> HEIGHT=<VAL>

per cancellare : make clean

Per eseguire : make run

## Implementazione

### MAPPA

La mappa, allocata in una porzione di memoria condivisa tra i processi main taxi e source, è implementata come un array di **struct cella** composta ciascuna da (tempo di attraversamento, capacità, numero attraversamenti, is\_hole).

Le celle sono risorse protette da un array di semafori(celle\_sem) che vengono inizializzate alla capacità massima delle celle, un altro semaforo per la scrittura in sezione critica sulle varie celle serve per aggiornare i valori delle celle come il numero di attraversamenti

### RICHIESTE VIAGGI

Le richieste vengono generate dalle source e inviati tramite una coda di messaggi ai taxi. Ogni source genera periodicamente (ogni 3 secondi) una richiesta, inviando un segnale SIGUSR1 a un processo source, viene generata un'ulteriore richiesta nel prossimo ciclo di generazione(ogni 3 secondi).

Ogni richiesta è composta dalla posizione della source che la genera e la destinazione.

Una richiesta è in attesa dalla sua generazione fino a quando il taxi, che avendo ricevuto la richiesta, ha raggiunto la source e quindi può evadere la richiesta.

Quando il taxi comincia ad evadere la richiesta di viaggio, quest'ultima può terminare in due modi :

-aborted : quando il viaggio viene annullato perchè è scaduto il timeout del taxi per attraversare una cella o quando la simulazione termina nel mentre;

-completed : quando raggiunge la destinazione.

### MOVIMENTO TAXI

I taxi, dopo aver prelevato una richiesta dalla coda di messaggi, si muove verso la fonte che l'ha generata per raccogliere il passeggero per poi andare verso la destinazione. il taxi si muove prima orizzontalmente e poi verticalmente, sempre evitando le celle HOLE e avviene come segue :

-attraverso la cella(nanosleep)

-transizione di cella rilasciando la cella in possesso e acquisendo la cella in cui muoversi (semtimeop)

durante lo spostamento i taxi aggiornano i vari parametri di attraversamento delle celle, e delle statistiche utili per la stampa di terminazione.

## INIZIO E FINE SIMULAZIONE

All'inizio della simulazione il processo main imposta un alarm che lo segnala della fine della simulazione, il processo main dealloca le risorse IPC(semaphori celle e coda di messaggi delle richieste) e attende la terminazione dei processi taxi e source, che avviene quando una delle system call sulle IPC condivise fallirà e stamperà su terminale le statistiche della simulazione.

## STAMPA

Ogni secondo il processo main stampa l'occupazione delle varie celle ottenuto prendendo le capacità massime delle varie celle e sottraendogli il valore dei semafori delle celle(con semctl( , , , GETALL)), inizializzate durante la costruzione della mappa alla capacità massima.

Le statistiche di richieste e taxi sono allocate in memoria condivisa subito dopo la mappa. Le statistiche delle richieste hanno un semaforo che gestisce la scrittura in sezione critica. Per stampare le TOPCELLS viene creato un array di puntatori a celle che viene riordinato a seconda del numero di attraversamenti e poi vengono evidenziati solo i TOPCELLS nella stampa(azzerando i valori delle altre celle). Anche le source vengono ordinate per facilitare la stampa finale.

## PARAMETRI

I parametri della simulazione vengono presi da stdin durante l'esecuzione