

Progetto di Laboratorio di Sistemi Operativi

a.a. 2014/15

Introduzione

Il progetto consiste nello sviluppare un sistema di negoziazione per l'acquisizione di risorse, simile alle aste (auctions), in cui un processo Banditore mette (virtualmente) in vendita risorse; risorse che più processi Clienti intendono comprare. I Clienti, per mezzo di un insieme di processi Agenti, creati appositamente, faranno avere al Banditore le loro offerte. Il Banditore deciderà quindi a chi assegnare le risorse in accordo alle richieste ricevute.

Processi da sviluppare

Banditore

Il processo Banditore ha il compito di assegnare risorse ai processi Clienti. Le risorse da assegnare sono lette da file. Per ogni risorsa, il file deve indicare quante risorse sono effettivamente disponibili, e qual è il costo minimo per unità. Esempio, seguendo lo schema: <nome-risorsa; unità disponibili; costo per unità>, un possibile file delle risorse potrebbe essere:

CPU; 4; 100

Memory-slot; 4096; 10

Per ogni risorsa, il Banditore crea un "tavolo delle offerte" (TAO) dedicato a quella risorsa. Il TAO è un'area di memoria condivisa che contiene, per ogni offerta fatta da un Client, le seguenti informazioni:

- pid-cliente
- quantità richiesta
- offerta per unità

Il Banditore può creare al più 3 TAO contemporaneamente, se ci sono più risorse da assegnare, il Banditore dovrà preoccuparsi di deallocare i TAO che non servono più e quindi di allocarne uno nuovo per le risorse non ancora assegnate.

Ogni TAO può contenere un numero limitato di offerte (es. 5). Nel caso ci siano più offerte del numero di slot disponibili, solo le offerte migliori vengono mantenute.

Il Banditore mantiene una lista dei Clienti registrati al sistema. Per ogni Cliente il Banditore memorizza le risorse cui il Cliente è interessato.

Quando il Banditore apre un nuovo TAO per una determinata risorsa, il Banditore informa tutti i Clienti interessati a quella risorsa mandando loro, via code di messaggi, l'id della shm corrispondente al TAO e l'id di un semaforo che regolerà il tempo di apertura dell'asta (si consiglia di usare l'attesa dello zero), insieme al prezzo base dell'asta.

Per ogni TAO, il Banditore gestisce quindi un'asta che deve avvenire nel seguente modo. Dopo aver informato tutti gli interessati, il TAO farà partire un timer di 3 secondi. Al termine del quale darà avvio

all'asta, e quindi consentirà ai Clienti di fare le loro offerte. L'asta può durare un tempo variabile in funzione del numero dei Clienti interessati (l'implementazione di questa parte è libera). Allo scadere del tempo, il Banditore chiude l'asta, legge il contenuto del TAO e assegna le risorse secondo le offerte migliori.

L'assegnazione consiste nel mandare un messaggio ai Clienti che hanno acquisito le risorse.

Il Banditore non esegue mai delle sleep.

Esempio 1.

Ad un certo punto dell'esecuzione il Banditore ha creato i due TAO per le due risorse Memory-slot e CPU.

TAO - CPU

TAO – Memory-slot

Cliente

Il processo Cliente legge da file le risorse che deve acquisire e la loro quantità. Il file riporta anche il budget disponibile a quel Cliente.

All'avvio, il Cliente si registra al Banditore mandando un messaggio contenente il proprio pid e l'elenco delle risorse cui è interessato. I Clienti possono essere interessati ad un numero variabile di risorse; potete risolvere questo problema usando più messaggi (uno per ogni risorsa), o usando una struttura del messaggio più articolata in modo da trasmettere più risorse in un unico messaggio, ma mantenendo la dimensione del messaggio fissa.

Il Cliente si mette quindi in attesa che il Banditore lo informi dell'apertura di un TAO per una delle risorse di suo interesse. Alla ricezione di questo messaggio, il Cliente avvia un processo Agente che si occuperà di fare effettivamente l'offerta (vedi dopo).

Avviato l'Agente, il Cliente torna in attesa di un nuovo messaggio da parte del Banditore.

Se il messaggio inviato dal Banditore si riferisce all'acquisizione di una risorsa, il Cliente scriverà su un file di log a lui dedicato la risorsa acquisita, la quantità e il costo complessivo.

Agente

Il processo Agente è creato dal Cliente ed ha il compito di acquisire una certa quantità di risorse.

Alla sua creazione, l'Agente riceve dal Cliente la quantità di risorse che deve acquisire e il budget massimo per quella risorsa.

L'Agente riceve dal Cliente inoltre le informazioni necessarie per accedere al TAO, quali la base d'asta, l'id del segmento di memoria condivisa e l'id dei semafori necessari.

L'accesso al TAO deve avvenire nel seguente modo:

- 1) nel periodo dei tre secondi prima dell'asta, tutti i processi agenti sono in attesa
- 2) quando l'asta si apre, gli agenti accedono in mutua esclusione al TAO, e scrivono in una riga del TAO la loro offerta.
- 3) un agente può scrivere una entry del TAO solo in due condizioni:
 - a. la entry è vuota
 - b. la entry contiene un'offerta inferiore rispetto a quella che si sta per fare
- 4) un agente che ha già un'offerta presente nel TAO non può aggiungerne un'altra: per ogni agente al più un'offerta deve essere presente nel TAO.
- 5) Gli agenti "competono" nel senso che continuano ad accedere al TAO fino a quando l'asta non si chiude. Ogni volta che accedono al TAO verificano che la loro offerta sia ancora tra le migliori. In caso affermativo rilasciano il TAO, in caso negativo fanno una nuova offerta che deve essere almeno migliore della peggiore attualmente presente nel TAO (purché il loro budget permetta loro di incrementare l'offerta).

Al termine dell'asta il processo Agente termina.

Esempio 2.

Durante la fase d'asta i processi Agente fanno offerte per conto dei rispettivi processi Cliente. Ad un certo punto dell'esecuzione la situazione potrebbe essere

14258	2	102
24587	4	101
24517	1	103
75845	3	101
21475	1	100

TAO - CPU

Una nuova offerta deve può essere fatta da un ulteriore processo Agente, purché il prezzo dell'offerta sia maggiore dell'offerta minima attualmente presente, cioè 100.

Potete gestire le offerte nel TAO come preferite: potete tenerle in ordine decrescente per offerta, oppure non ordinate, in ogni caso le due soluzioni hanno vantaggi e svantaggi che commenterete nella vostra relazione.

A questo punto, il processo Banditore chiude l'asta e decide a chi assegnare la CPU. In questo caso potrebbe decidere di assegnare una CPU al processo 24517, una al processo 14258, e due al processo 24587, oppure una al processo 24587 e una al processo 75845: i tiebreak possono essere risolti come meglio credete.

Strutture di IPC

- Coda di messaggi tra Banditore e Clienti
- Per ogni TAO
 - o Segmento di memoria condivisa
 - o Pool di semafori di sincronizzazione

Gestione dei segnali e Terminazione pulita

Il sistema non deve creare zombie: questa condizione deve essere vera mentre il sistema è in funzione, non quando è stato ormai chiuso.

I periodi di apertura/chiusura delle aste possono essere gestiti dai segnali di alarm. Attenzione però che può essere impostato un solo alarm per volta.

Un Cliente termina naturalmente quando ha partecipato ad un'asta per ciascuna delle sue risorse. Può però anche essere terminato anzitempo quando riceve un segnale di SIGINT. In tal caso deve anche:

- deregistrarsi dal Banditore
- terminare i suoi agenti, nel caso ve ne fossero

(Il Banditore ignorerà offerte provenienti da Clienti deregistrati.)

Il Banditore termina naturalmente quando ha fatto un'asta per ogni risorsa letta dal file. Può anche terminare anzitempo intercettando un segnale di SIGINT.

In ogni caso, alla terminazione del Banditore:

- tutti i Clienti devono essere deregistrati,
- le strutture di IPC deallocate

Commento generale

Questo documento è una traccia del lavoro che deve essere svolto, molti dettagli sono mancanti perché dipendono dalle vostre decisioni implementative.

Ad esempio, i processi Cliente possono essere creati manualmente, oppure potete decidere di implementare un Generatore, che crea un determinato numero di processi Cliente (scelta consigliata).

Va comunque assunto che i processi Agente sono “onesti”, nel senso che accedendo ad un’area di memoria condivisa, hanno anche accesso ai dati immessi da altri processi, e potenzialmente possono modificarli. Ovviamente la modifica in un TAO deve avvenire esclusivamente nelle modalità indicate.