

Relazione Progetto Sistemi Operativi 2021-2022

Componenti del gruppo

- Lorenzo Bafunno, N° 944233, *lorenzo.bafunno@edu.unito.it*
- Alessio Hu, N° 950386, *alessio.hu@edu.unito.it*
- Francesco Targa, N° 944743, *francesco.targa@edu.unito.it*

Strutture IPC

1. Memorie condivise:
 - a. libro mastro(array di transazioni e non di blocchi di transazioni);
 - b. size del libro mastro;
 - c. array di utenti;
 - d. array di nodi;
 - e. size dei nodi (massimo $SO_NODES_NUM^2$);
2. 1 Set di semafori (SYS-V):
 - a. scrittura nel libro mastro da parte dei nodi;
 - b. sincronizzazione partenza processi nodo;
 - c. sincronizzazione partenza di tutti i processi utente;
3. Code di messaggi (SYS-V):
 - a. creazione di un nuovo processo nodo;
 - b. rifiuto delle transazioni degli utenti;
 - c. ricezioni di transazioni nei nodi;
4. Pipe:
 - a. comunicazione per l'aggiunta di una amico;

Implementazione

Processo master

Il master si occupa di generare tutti le strutture IPC ad eccezione delle code dei nodi (3.c).

Attesa dei processi nodi e utenti

Il master utilizza solo due semafori del set che genera:

- il (2.b) per attendere che tutti i nodi siano pronti così da poter generare la lista di amici
- il (2.c) per attendere che tutti i processi abbiano completato la loro prima configurazione

Generazione della lista degli amici e invio

All'atto della generazione dei nodi il master crea una pipe (4.a) per ciascun nodo esistente: questa viene utilizzata per permettere ai nodi di aggiungere un nuovo nodo amico.

Generazione di un nuovo nodo ed invio/rifiuto di una transazione

Quando il master riceve un segnale segnale di *SIGUSR1*, viene notificato per la lettura di una transazione nella coda per la generazione di un nuovo nodo (3.a). Il master legge dalla coda tutti i messaggi presenti così da risolvere il merge di segnali, creando un nodo per ogni transazione. La transazione viene inserita nella coda di transazioni (3.c) del nodo generato. Inoltre, attraverso le pipe, a *SO_NUM_FRIENDS* nodi viene ordinato di aggiungere come amico il nodo appena creato. Se il numero di nodi ha raggiunto il massimo, il master rifiuta la transazione: viene inserita nella coda di rifiuto (3.b) indicando il msg type uguale al pid dell'utente sender, inviandogli un *SIGUSR1*.

Stampa periodica e terminazione dell'esecuzione da segnale

Per la stampa periodica utilizziamo *alarm()*. Nella gestione del segnale *SIGALRM*, vengono effettuate anche aggiornamenti ai bilanci dei nodi e degli utenti.

L'esecuzione del master può essere terminata prematuramente con un segnale di *SIGINT*.

Processo utente

Generazione e invio di una transazione

I processi utenti si avvalgono delle code di messaggi create dai nodi (3.c) per cercare di inviare transazioni. Se la trova piena o ha un bilancio minore di 2, aumenta il numero di tentativi consecutivi d'invio, altrimenti se riesce ad inviarla, lo azzerava e invia un segnale *SIGUSR1* al nodo scelto.

Rifiuto di una transazione

Quando una transazione viene rifiutata dal master oppure da un nodo, l'utente viene notificato tramite un segnale *SIGUSR1*. Il processo utente legge dalla coda di rifiuto (3.b) un messaggio con tipo uguale al proprio pid, aggiunge il totale (quantità + reward) al proprio bilancio corrente.

Processo nodo

Il nodo si occupa di gestire la propria coda di messaggi e la transaction pool interna, un array di transazioni non ordinato. Se il numero di transazioni presenti è minore della dimensione di un blocco da processare, si sospende con una *pause()* in attesa di un segnale.

Ricezione delle transazioni

Quando viene ricevuto un segnale di *SIGUSR1* il nodo legge dalla coda tutti i messaggi presenti così da risolvere il merge di segnali. Se il nodo ha posto nella pool accetta le transazioni e le inserisce. Se una transazione ha raggiunto *SO_HOPS* salti, il nodo recupera la coda di messaggi per la creazione dei nodi (3.a) e inserisce la transazione, inviando un segnale *SIGUSR1* al master. La coda di rifiuto delle transazioni (3.b) viene utilizzata anche dai nodi qualora non riuscissero a inviare la transazione ad un nodo amico oppure al master. Se la transaction pool del nodo è piena viene scelto un nodo amico random al quale verrà inviata la transazione tramite la sua coda.

Invio periodico di una transazione ad un amico

Attraverso un *alarm()*, ogni secondo il nodo controlla se ha almeno una transazione nella pool, sceglie un nodo amico e gli invia l'ultima transazione nella sua pool, eliminandola in caso di invio con successo.

Aggiunta di un amico

Quando un nodo riceve un *SIGUSR2*, vuol dire che il master ha scritto il pid di un nodo appena creato nella pipe master-nodo specifica del nodo. Il nodo andrà a leggerlo e ad aggiungere l'amico nella sua lista di amici.

