

PROGETTO DI SISTEMI OPERATIVI

Struttura del progetto:

Il progetto è stato suddiviso in 3 file: `opt_conf.h` contenente le definizioni per le variabili globali del codice come `POP_SIZE`, `sim_time`, preferenze per la dimensione dei gruppi, ecc...; `progetto.c` con il codice eseguito dal processo ufficio studenti (padre) e dagli studenti (figli); ed infine il file `util.c` il quale raccoglie tutte le librerie necessari per il funzionamento del codice e la definizione di alcune funzioni che hanno lo scopo di facilitare la lettura e la comprensione generale del codice. In questo file è stata inoltre definita la struttura studente costituita di vettori di interi, di dimensione `POP_SIZE`, e una matrice di dimensione `POP_SIZE` x 3 (numero massimo di studenti che è possibile invitare in un gruppo). Contengono le seguenti variabili:

matricola: determina il numero di matricola tramite il PID di ogni processo studente;

turno: ottenuto dalla divisione tra pari e dispari del numero di matricola

voto_AdE: voto di architetture degli elaboratori, un numero casuale compreso tra 18 e 30;

voto_SO: voto finale del progetto, calcolato dal ufficio studenti alla fine della simulazione;

nof_elements: numero casuale con probabilità pesata compreso tra 2 e 4;

ruolo: identifica il ruolo di ogni studente all'interno di un gruppo (1: leader; 0: membro; -1: non definito);

inviti e rifiuti: contatori del numero di inviti o rifiuti fatti da ogni studente;

nAdepti: conta il numero di studenti sotto la direzione di un leader di un gruppo;

Adepti: matrice che identifica gli studenti sotto la direzione del leader di un gruppo.

E' stata, infine, creata una funzione *preferenze* che ha lo scopo di generare il numero di elementi del gruppo di ogni studente basandosi su determinate percentuali definite in `opt_conf`, dai valori `P2`, `P3`, `P4`.

Per eseguire il progetto basta compilare il file `progetto.c` (`gcc progetto.c -o progetto`) ed eseguire il file generato (`./progetto`).

Funzioni, oggetti e scelte progettuali:

Per la realizzazione del codice si è deciso di utilizzare:

- 3 semafori: per la sincronizzazione tra processi;
- 1 code di messaggi: usata dagli studenti per l'invito dei colleghi al gruppo di lavoro;
- 1 memoria condivisa: contenete la struttura studente

La memoria condivisa contiene la struttura studente in cui è possibile accedere, tramite contatore del numero di processi, per modificare le proprie variabili.

La ricerca e scelta dei compagni di lavoro è gestita da una coda di messaggi, in cui ogni studente che non ha ancora un ruolo (etichettato con -1), dopo essersi accertato di non avere già degli inviti in sospeso, manda agli studenti del suo stesso turno un messaggio, contenete la propria posizione all'interno della struttura studente. Se uno degli studenti invitati accetta di unirsi al gruppo, questo andrà a modificare la variabile *ruolo* sia propria (con 0), sia quella dello studente che lo ha invitato (con 1) e le variabili *nAdepti* e *Adepti* dello studente ora leader; dopo di che provvederà a rifiutare ogni altro invito che gli viene mandato. Lo studente che invece diventa leader continuerà a cercare altri colleghi per raggiungere la propria preferenza di elementi del gruppo, e rifiuta ogni altro invito che gli viene mandato.

Per sincronizzare gli invii dei messaggi ed evitare la creazione di deadlock gli studenti non possono invitare altri studenti che stanno invitando a loro volta, questo è possibile grazie ad un semaforo che vale 2 durante l'invio del messaggio e che viene decrementato per attendere la risposta al messaggio e decrementato ulteriormente dopo la ricezione di una risposta.

La scelta di uno studente di accettare o rifiutare un invito avviene confrontando i valori delle variabili dello studente che manda l'invito, e accetta se una o più delle seguenti condizioni sono verificate:

- voto di architetture dello studente invitato minore o uguale al voto di architetture dello studente che invia richiesta
- studente che invita ha una preferenza di elementi del gruppo uguale o maggiore dello studente invitato;
- lo studente invitato ha raggiunto il numero massimo di rifiuti
- lo studente che invita è un leader e lo studente invitato non ha ancora un ruolo.

In caso contrario rifiuta l'invito e incrementa il proprio contatore di rifiuti.

La simulazione ha fine quando un segnale allarm, lanciato dal processo padre, viene intercettato; questo va a modificare la variabile `esc` che non permette ulteriori modifiche alle variabili all'interno della memoria condivisa.

Voto finale e Conclusioni:

Il voto finale del progetto si è deciso di farlo scrivere nella memoria condivisa direttamente dal processo ufficio studenti nella variabile `voto_SO` relativa ad ogni studente, così che gli studenti non debbano far altro che andare a leggere il valore per poi stamparlo.

Queste decisioni portano ad avere un numero di studenti che, alla fine del progetto, non trovano un gruppo mediamente in torno al 1-5% degli studenti totali, portando la media dei voti di sistemi operativi ad essere uguale o 1-2 voti superiore a quella di architetture.