

RELAZIONE PROGETTO SISTEMI OPERATIVI 2019/20

1. Compilazione progetto

La compilazione del progetto avviene tramite Makefile

Recarsi nella cartella sorgenti con il Terminal e digitare "make"

2. Esecuzione progetto

L'esecuzione del progetto avviene tramite Makefile

Recarsi nella cartella sorgenti con il Terminal e dopo aver compilato digitare "make run"

3. Modifica parametri di configurazione

I parametri di configurazione vengono letti da file.

Per la modifica dei parametri recarsi nella cartella dei sorgenti, poi in `./config` e modificare il file `config.txt` cambiando solamente i valori dei parametri (non modificare l'ordine)

NB E' presente un separatore tra il nome del parametro e il numero, serve per la lettura da file. Non modificarne la struttura, in caso di modifica del separatore sarà necessario modificare il sorgente e ricompilare.

(Nel codice è presente una parte commentata in `util.c` per la lettura da file senza il separatore).

4. Scelte progettuali

Il codice è suddiviso in 5 file: un file per il master, uno per i giocatori, uno per le pedine e due file di appoggio: `util.h` `util.c`

Si utilizzerà la funzione `execve` per sostituire il codice da eseguire dei processi figli con quello contenuto nel file indicato.

Il progetto utilizza 3 shared memory

Una shared memory per la scacchiera:

La scacchiera è un vettore in memoria condivisa di grandezza `SO_BASE*SO_ALTEZZA`, ogni cella della scacchiera è una struttura in cui vengono mantenute le informazioni sulle pedine (presenti o meno), sulle bandierine (presenti o meno) e su chi ha piazzato un'eventuale pedina su quella cella.

Una shared memory per i semafori:

La shared memory per i semafori è una struttura condivisa contenente tutti i set di semafori.

`setSemaforiScacchiera`: set di dimensioni `SO_BASE*SO_ALTEZZA` che permette di accedere in mutua esclusione a ogni cella della scacchiera

`setSemaforiPosizionamentoPedine`: questo set viene utilizzato per piazzare a turno le pedine (si può trovare una spiegazione più dettagliata successivamente)

`semaforoFineTurnazionePedine`: utilizzato per sapere quando tutti i giocatori hanno piazzato le loro pedine

`semaforoIndicazioniPedine`: utilizzato per sapere quando tutti i giocatori hanno finito di dare le informazioni alle loro pedine

`gameStart`: quando tutti i giocatori hanno piazzato le pedine e dato indicazioni sugli spostamenti, il master dà a tutti il permesso di iniziare.

`endRound`: utilizzato per la sincronizzazione di fine round, quando tutte le bandierine vengono mangiate il round ricomincia.

Quando finisce il gioco il master cambia il valore ad una variabile in memoria condivisa `runGiocatori` che servirà per far uscire i giocatori dal ciclo del round.

Stessa metodologia è stata utilizzata con la variabile `runPedine`, cambiata di valore dal primo giocatore che riceve il segnale di fine gioco.

Una shared memory per le mosse residue delle pedine:

In questa shared memory di dimensione numeroPedineTotali vengono mantenute le informazioni sulle mosse di ogni pedina.

Note:

- per la logica con cui sono state implementate le shared memory le pedine sono distinguibili in base al giocatore
- Viene utilizzata la funzione waitForZero con il compito di attendere il valore zero di un semaforo a conteggio.

MASTER

Dopo le inizializzazioni il processo master genera SO_NUM_G processi giocatore e tramite una waitForZero sul semaforo semaforoFineTurnazionePedine attende che i giocatori abbiano finito di piazzare le pedine, il semaforo e' inizializzato a SO_NUM_G e ogni giocatore dopo aver piazzato le sue pedine decrementa il valore del semaforo.

ROUND: Il processo master distribuisce un numero casuale di bandierine tra SO_FLAG_MIN e SO_FLAG_MAX da posizionare sulla scacchiera, sempre con coordinate casuali.

Permette poi ai giocatori di dare informazioni alle pedine tramite i due semafori semaforoIndicazioniPedine (è un set di due semafori). Il primo viene inizializzato a 1 e rappresenta lo stato di attesa per i giocatori a dare informazioni alle pedine. Il master successivamente decrementerà il semaforo, i giocatori effettuando una waitForZero su quest'ultimo potranno dare indicazioni alle pedine quando il semaforo assumerà il valore atteso. Il secondo semaforo rappresenta invece l'attesa, da parte del master, che i giocatori abbiano finito di dare indicazioni alle pedine. Viene inizializzato ad SO_NUM_G e viene decrementato da ogni giocatore dopo aver suggerito ad ogni sua pedina quale mosse effettuare. Il master effettuerà una waitForZero su questo semaforo in modo da poter procedere solamente quando tutte le pedine avranno ricevuto la mossa da effettuare.

Successivamente dopo aver avviato il timer, il master permette a tutti i giocatori di cominciare il round sbloccando il semaforo gameStart.

A fine di ogni round il master effettua un controllo sulle bandierine mangiate ed aggiorna i punteggi dei giocatori.

Se il timer scade viene mandato un segnale ai giocatori dicendo che il gioco e' concluso, lo stesso avverrà tra giocatori e pedine. Una volta terminate le pedine seguiranno i giocatori e infine il master.

GIOCATORI

I giocatori devono piazzare le pedine a turno.

Il semaforo setSemaforiPosizionamentoPedine e' un set di SO_NUM_G semafori, ad ogni giocatore viene associato un indice per accedere al set (Il giocatore 0 ha assegnato l'indice 0 del set, il giocatore 1 l'indice 1, ecc.)

Il set e' così inizializzato:

0 1 1 1

Il giocatore con il valore 0 del semaforo corrispondente può piazzare la pedina, gli altri aspetteranno il valore atteso dal giocatore precedente.

Quando un giocatore ha piazzato una pedina in una posizione casuale della scacchiera decrementa il valore del semaforo successivo in modo da passare il turno al prossimo giocatore.

Si procede finchè tutti i giocatori non hanno piazzato le loro pedine.

Ogni giocatore comunicherà al master di aver piazzato tutte le sue pedine decrementando di 1 il semaforo semaforoFineTurnazionePedine.

ROUND: La comunicazione di informazioni alle pedine avviene tramite code di messaggi.

Ogni giocatore condivide una coda di messaggi usata per tutte le sue pedine (NON per ogni pedina).

Ad inizio round dopo che il master ha dato il permesso di dare informazioni alle pedine il giocatore invia le sue informazioni e lo notifica al master tramite il semaforo prima specificato. Resta poi in attesa che ricominci un nuovo round per comunicare nuovamente alle pedine i movimenti.

I movimenti consistono in uno spostamento casuale tra i quattro possibili. Sarà compito della pedina valutare se è possibile compiere lo spostamento in base alla sua posizione e alle pedine che la circondano.

Quando riceve il segnale di fine gioco da parte del master notifica le sue pedine con un altro segnale, attende la loro terminazione poi termina anch'esso.

PEDINE

A inizio round le pedine ricevono informazioni sulla coda di messaggi del loro giocatore.

Dopo che il master ha dato il via al gioco si muoveranno fin quando una delle tre condizioni non sarà più vera: possibilità di effettuare ulteriori mosse, tempo a disposizione (non è ancora scaduto il timer), non è ancora stata conquistata una bandierina.

A seconda della condizione non verificata avvengono eventi differenti:

- Se le mosse esauriscono la pedina non effettuerà più spostamenti e rimarrà nell'ultima cella raggiunta
- Se scade il timer terminano l'esecuzione
- Se conquistano una bandierina rimangono ferme nel punto in cui l'hanno conquistata fino al prossimo round

5. Strategia

Le pedine si muovono casualmente nella scacchiera.

Nel caso in cui venissero conquistate tutte le bandierine le pedine riceveranno un nuovo suggerimento dal giocatore e potranno conquistare le nuove bandierine con lo stesso criterio.

Segue sotto uno schema rappresentante il corso degli avvenimenti durante l'esecuzione del programma.

