UNIVERSITÀ DEGLI STUDI DI TORINO
DIPARTIMENTO DI INFORMATICA

Relazione del progetto di Sistemi Operativi

a.a. 2019/2020

Gandolfi Alessandro
859130
alessandro.gandol@edu.unito.it

D'Auria Mario
869420
mario.dauria@edu.unito.it

Indice

Schema generale

MASTER	GIOCATORE	PEDINA
Crea oggetti IPCS		
Inizializza parametri per processi giocatore		
Crea una scacchiera (SO_BASE * SO_ALTEZZA)		
Crea SO_NUM_G processi giocatori		
	Inizializza parametri per processi pedina	
	for i ∈ {0,...,SO_NUM_P}: Cerca una posizione libera sulla scacchiera per piazzare una pedina. Crea processo pedina	
		Legge i parametri passati dal processo giocatore
Piazza le bandierine sulla scacchiera		
	Assegna le bandierine, come obiettivi da raggiungere, ai processi pedina	
		Calcola il percorso per raggiungere la bandierina assegnata

FINE GIOCO		
Invio segnale di fine gioco alle squadre		
		Termina
	Termina	
Rimuove oggetti IPCS		
Termina		

Master.c

Il processo Master crea gli oggetti IPCS:

- **Memoria Condivisa** di grandezza SO_BASE * SO_ALTEZZA per la scacchiera.
- **Coda di messaggi** per lo scambio di informazioni tra i processi (master, giocatore e pedina).
- **Semafori:**
    1. un set di semafori per la scacchiera costituito da SO_BASE * SO_ALTEZZA semafori inizializzati a 0 (tutte le posizioni della scacchiera sono libere).
    2. un set di semafori per i processi giocatori costituito da SO_NUM_G semafori: usato dai giocatori per piazzare uno alla volta le proprie pedine sulla scacchiera e per far partire tutte insieme le pedine.

Dopo aver creato SO_NUM_G processi giocatori, il master rimane in attesa del messaggio di fine piazzamento pedine da parte dell'ultimo giocatore che ha piazzato le pedine.
Quando l'ultimo giocatore invia il messaggio atteso, il master si occupa di piazzare sulla scacchiera un numero random (tra SO_FLAG_MIN e SO_FLAG_MAX) di bandierine. Per ogni bandierina piazzata assegna un punteggio random, in maniera sistematica in modo da distribuire al meglio SO_ROUND_SCORE (punteggio massimo per ogni round): una bandiera vale al massimo la metà dei punti rimanenti ancora da distribuire e non può valere meno di 1; l'ultima bandierina vale il restante dei punti non assegnati. Quando ha finito di piazzare le bandierine invia un messaggio ai giocatori per informarli dell'avvenuto piazzamento e si mette in attesa di ricevere un messaggio di "risposta" da parte dei giocatori (o dalle squadre giocatori + pedine) che avvisano che le pedine hanno finito di calcolare il proprio percorso per cercare di raggiungere l'obiettivo (bandierina) assegnatogli. Dopo che questo evento avviene si mette in attesa di ricevere un messaggio, per ogni bandierina piazzata, di avvenuta cattura delle stesse da parte delle pedine. Durante il gioco si occupa di stampare lo stato della scacchiera prima e dopo il round. Dopo un round stampa il punteggio attuale dei giocatori (o squadre) e le loro mosse residue a

disposizione.
*Tutto questo lo fa per ogni round, finché non finisce il gioco (quando un round non è finito prima di SO_MAX_TIME secondi o quando le pedine non hanno più mosse a sufficienza per raggiungere gli obiettivi e quindi restano ferme aspettando la fine di SO_MAX_TIME secondi).
Quando il gioco finisce il master riceve un segnale SIGALRM e si occupa di "avvisare" le squadre per la fine del gioco, tramite invio di un segnale SIGUSR1, e dopo di stampare lo stato ultimo del gioco (scacchiera e metriche), per poi mettersi in attesa che le squadre terminino. Solo dopo la terminazione di tutte le squadre si occupa di eliminare tutte le risorse (oggetti IPCS) creati e termina.

Giocatore.c

Quando vengono creati, i processi giocatore, leggono i parametri passati dal master utili ad avere accesso agli oggetti IPCS ed essere in possesso delle informazioni necessarie che il master ha assegnato ad ognuno dei processi. Dopodiché si occupano di creare SO_NUM_P processi pedina per formare la propria squadra: per ogni pedina il processo giocatore cerca una posizione libera sulla scacchiera e una volta trovata la assegna alla pedina che appena dopo verrà creata. La ricerca di una posizione libera viene effettuata, dal processo giocatore, in maniera random, rispettando le grandezze della scacchiera, finchè non trova una posizione libera che sia abbastanza distante dalle pedine della propria squadra precedentemente piazzate. La creazione dei processi pedina avviene a turni dettati dal master: un processo giocatore per volta esegue la creazione dei processi pedina. L'ultimo giocatore che effettua il piazzamento e la creazione delle pedine si occupa (anche da parte degli altri giocatori) dell'invio di un messaggio al master di fine piazzamento. In questo momento tutti i giocatori (o le squadre) restano in attesa di un messaggio dal master di fine piazzamento bandierine.
Ad ogni round i giocatori si occupano di assegnare, alle proprie pedina, le bandierine (obiettivi). L'assegnazione avviene secondo il seguente criterio: se una pedina si trova in vicinanza di una bandierina, e la stessa pedina

non è stata assegnata ad una bandierina che vale di più di quella attuale (di cui si sta effettuando lo scan), allora l'attuale bandierina viene assegnata come obiettivo della pedina. Nel caso in cui è stata assegnata, ad una pedina, una bandierina con minor punteggio di un'altra di cui si sta effettuando lo scan allora viene effettuata la riassegnazione dell'obiettivo con la bandierina con un punteggio maggiore. Tutto ciò avviene se la pedina in questione ha abbastanza mosse da raggiungere l'obiettivo assegnato. Il giocatore quindi, nel processo di assegnazione, tiene conto dei punteggi delle bandierine, delle distanze tra le pedina e le bandierine e delle mosse che ogni pedina ha a disposizione. La comunicazione tra giocatore e pedina riguardo l'assegnazione dell'obiettivo avviene tramite invio di un messaggio (da parte del giocatore) di avvenuta assegnazione. In questo momento il giocatore sarà in attesa di un messaggio da parte di ogni pedina di avvenuto calcolo percorso (che intercorre tra la loro posizione e l'obiettivo assegnatogli).
Dopo essersi assicurato che ogni propria pedina ha calcolato il percorso, invia un messaggio al master per avvisare che la squadra è pronta per l'inizio del round.
Alla fine del gioco il processo giocatore (o meglio la squadra) riceverà un segnale da parte del master in modo che smetta di fare qualsiasi operazione mettendosi in attesa della terminazione delle pedine. Solo quando tutte le pedine, appartenenti ad esso, terminano, termina anche lui.

Pedina.c

Quando vengono creati, i processi pedina, leggono i parametri passati dal giocatore utili ad avere accesso agli oggetti IPCS ed essere in possesso delle informazioni necessarie assegnate al momento della creazione per la distinzione delle varie pedine.
Una volta creati restano in attesa di un messaggio da parte del giocatore di avvenuta assegnazione dell'obiettivo (bandierina) da raggiungere. Dopo aver ricevuto tale messaggio calcolano il percorso che intercorre tra la loro posizione e la posizione dell'obiettivo assegnatogli. Alla fine del calcolo del percorso inviano un messaggio al giocatore (a cui

appartengono) di fine calcolo percorso e aspettano l'inizio del gioco (e successivamente del round). Iniziato il gioco (o il round) iniziano a muoversi sulla scacchiera tentando di arrivare all'obiettivo assegnatogli e di prenderlo: ad ogni passo, se l'obiettivo non risulta stato già preso da un'altra pedina, tentano di occupare una casella della scacchiera e se tale casella risulta già occupata da un'altra pedina ritenta di occuparla dopo (SO_MIN_HOLD_NSEC / 2) nanosecondi. Se una pedina prende l'obiettivo assegnatogli invia un messaggio al processo master per informarlo che tale bandierina è stata presa.
Quando il gioco finisce i processi pedina (e l'intera squadra) ricevono un segnale SIGUSR1 dal master e terminano.

**Il concetto di squadra è stato definito come l'insieme di processo giocatore e processi pedina che il processo giocatore ha creato. A livello di codice è stato settato un pid di gruppo in modo che la fine (o l'interruzione) del gioco fosse comunicata dal master all'intera squadra.

Struttura del progetto

Il progetto è stato strutturato in modo da distinguere i file sorgenti, i file oggetto (eseguibili) e i file per i parametri di configurazione: una cartella src contenente i file sorgenti (.c e .h), una cartella bin contenente i file eseguibili e una cartella config contenente i file con i parametri di configurazione. I file di configurazione sono stati divisi per modalità di gioco: un file per la modalità easy, un file per la modalità hard e un file per la modalità debug utilizzato in fase di progettazione per dubugging. Nei primi due sono stati inseriti i parametri di configurazione per modalità di gioco suggeriti nella consegna del progetto. La scelta della modalità di gioco viene definita dall'utente in fase di esecuzione inserendo il parametro "mode=hard" oppure "mode=easy". Tale scelta implicherà la lettura dei parametri di uno dei due file (a seconda della scelta) da parte di tutti i processi che eseguiranno la propria computazione con i relativi parametri.

Come eseguire il progetto

Per eseguire il progetto ci siamo avvalsi di un utility make in cui sono dettate le regole per la compilazione e l'esecuzione. Pertanto basti eseguire il comando "make" per eseguire la compilazione dei sorgenti e poi il comando "make run mode=<modalità di gioco>" per eseguire i file oggetto prodotti dalla compilazione, oppure eseguire il comando "make all mode=<modalità di gioco>" per eseguire la compilazione e l'esecuzione. La compilazione viene eseguita settando i seguenti flag: -std=c89, -pedantic, -Wunused-variable.