DRAFT

Progetto Lab S.O. (UNIX)

2016/2017

Tema

L'obiettivo del progetto è sviluppare un simulatore di una partita di calcetto. Questo documento descrive alcuni dettagli per l'implementazione, sui quali è però possibile elaborare soluzioni personali, usando tutto il materiale visto durante il laboratorio. I progetti verranno valutati in base alla correttezza, completezza, eleganza e chiarezza del codice, creatività e, ovviamente, alla sensatezza delle soluzioni adottate. Il punteggio massimo di un esame di laboratorio sarà 3 punti. Il voto è individuale, anche nel caso di gruppi. Valgono infine tutte le regole illustrate durante il corso e descritte nei documenti pubblicati su I-Learn.

File di configurazione

Il progetto deve utilizzare un file di configurazione, contenente almeno i seguenti elementi:

1. Durata partita in secondi;
2. Parametri di "dinamica partita", con valori da 1 a 100:
   2.1. goal: probabilità di successo di tiro (goal, appunto);
   2.2. infortunio: probabilità che un giocatore si infortuni durante una "giocata", descritta in seguito;
   2.3.dribbling: probabilità di riuscire ad effettuare un dribbling e di continuare una "giocata", descritta in seguito.

Questi parametri agiscono sul processo "fato", anch'esso descritto in seguito.

Funzionamento

Tutto comincia da un processo "generatore" o "arbitro", che genera due processi "squadra" A e B, un processo "fato", e memorizza il punteggio corrente. Usa un signal handler per gestire i goal e aggiornare il punteggio.

Ogni processo squadra genera 5 processi "giocatore".

Il processo “fato” riceve messaggi da una coda e risponde casualmente ed in maniera binaria ad eventi generati durante la partita dai giocatori (serve a randomizzare il successo di goal, dribbling ed infortuni con certi coefficienti di “dinamica partita" valorizzati nel file di configurazione). La coda deve essere usata per scambio messaggi privato tra “giocatore" e “fato” usando adeguatamente il campo msgtype.

Il pallone è un semaforo: tutti i giocatori cercano continuamente (in un loop) di accedervi. Chi riesce ad accedere “possiede il pallone” e può dedicarsi alla “giocata”, cioè la generazione casuale di uno dei seguenti eventi:

1. infortunio -> messaggio inviato a “fato”, che risponde 0 o 1 per segnalare la morte del processo e liberazione slot su semaforo associato a relativa squadra che ricrea un nuovo processo giocatore da sostituire (il valore deve sempre essere 5 usando un semaforo) e rilascia la risorsa “pallone"
2. tiro -> messaggio inviato al processo “fato”, che risponde 0 o 1 su stessa coda di messaggi in caso di insuccesso o successo (goal). Se avviene un goal, si genera un segnale che viene catturato dal processo “arbitro” che aggiorna il punteggio della partita.
3. dribbling -> come per il tiro, messaggio inviato a “fato” che risponde con 1 o 0 per indicare se riuscito oppure no: se è riuscito, si cicla nuovamente per un nuovo tentativo di tiro o di infortunio o di dribbling, altrimenti viene liberata la risorsa “pallone" e tutti gli altri giocatori possono accedervi (essendo in loop continuo di tentativo di accesso al semaforo).

Tutti gli eventi devono essere scritti in un file di log completo gestito dal processo “fato”, e visualizzati su terminale in modo chiaro (con un minimo di “formattazione”).

Terminato il programma, tutte le strutture condivise devono essere de-allocate, e non devono essere presenti processi zombie.