

# Progetto Lab S.O. (UNIX)

## 2016/2017

### Tema

L'obiettivo del progetto è quello di sviluppare un simulatore di una partita di calcetto. Vengono descritti in questo documento alcuni dettagli su cui basare 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, e creatività. 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 definite durante il corso e formalizzate nei documenti pubblicati sul portale I-Learn.

### File di configurazione

Il progetto avrà bisogno di usare un file di configurazione, contenente almeno i seguenti parametri:

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 random di uno dei seguenti eventi:

1. infortunio -> messaggio che viene 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 che viene 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”).

Vi si chiede l'utilizzo della memoria condivisa per gestire la comunicazione tra due o più processi in un qualsiasi punto del flusso di gioco e rispetto ad un obiettivo a totale decisione vostra.

Terminato il programma, si chiede che tutte le strutture condivise siano disallocate automaticamente, e che non ci siano processi zombie.

## Modalità di Esame

### **1) Scritto: obbligatorio. Permette di prendere un voto massimo di 27/30.**

Lo scritto è diviso nelle parti di Teoria, Laboratorio Unix, e Linguaggio C (da quest'ultima parte sono esonerati gli studenti a cui sono stati riconosciuti i 3 cfu di linguaggio C)

### **2) Discussione sull'esercitazione finale di laboratorio: obbligatoria.** Permette di incrementare di al massimo di 3 punti il voto conseguito allo scritto.

### **3) Orale vero e proprio: facoltativo.** Integra il voto ottenuto sostenendo le parti di esame 1) e 2), e permette di prendere un voto massimo di 30/30 e lode. Per sostenere l'esame orale è necessario aver preso almeno 24/30 allo scritto.

## NOTE:

- L'esame scritto si intende superato se si prende almeno 18.
- Sostenendo solo le parti 1) e 2) è possibile ottenere un voto massimo di 30/30.
- Sostenendo la prova orale facoltativa (3), è possibile prendere anche 30 e lode pur non avendo conseguito il punteggio massimo nella prova scritta (ovviamente si può però anche essere bocciati).
- È possibile presentarsi a tutti gli appelli (cinque all'anno) e consegnare fino a tre scritti per anno accademico. La consegna di uno scritto annulla automaticamente qualsiasi voto conseguito in scritti precedenti.
- Uno scritto nel quale vi ritirate non conta, è come se non vi foste neanche iscritti all'appello.
- L'orale facoltativo (che riguarda tutti gli argomenti del corso: teoria, Unix, C) può essere dato in qualsiasi momento dell'anno su appuntamento, ma solo dopo aver discusso l'esercitazione finale e passato lo scritto.
- Per la discussione dell'esercitazione finale riceverete istruzioni dettagliate più avanti dal docente di laboratorio.
- Una volta passato lo scritto, il voto conseguito resta valido per un anno, se però si ridà lo scritto e si consegna il nuovo voto annulla qualsiasi voto precedente.
- L'esercitazione di laboratorio deve essere sviluppata e discussa entro Dicembre 2017, altrimenti occorrerà sviluppare e discutere l'esercitazione successiva, che verrà assegnata intorno a Novembre 2017.
- Non è possibile sostenere la prova scritta senza aver discusso prima l'esercitazione finale di laboratorio.

## Modalità di consegna e discussione dell'esercitazione

- Il progetto può essere individuale o di gruppo (massimo tre persone). La discussione del progetto di laboratorio deve essere svolta da tutti i componenti del gruppo contemporaneamente. Almeno tre giorni lavorativi prima della discussione il gruppo deve consegnare in formato elettronico:
  - una breve relazione contenente una descrizione degli aspetti caratterizzanti la soluzione stessa (una pagina o due sono normalmente sufficienti); è gradito uno schema grafico a corredo della relazione che illustri la struttura complessiva del sistema realizzato; la relazione deve riportare chiaramente nella prima pagina i nomi dei membri del gruppo, i loro numeri di matricola e gli indirizzi istituzionali di posta elettronica.
  - copia di tutti i sorgenti che costituiscono il progetto
- Il progetto deve essere consegnato almeno 3 giorni lavorativi prima della discussione
- Tutti gli studenti del gruppo devono appartenere allo stesso turno di laboratorio
- Le richieste di discussione del laboratorio dovranno pervenire almeno 10 giorni prima dalla prova scritta

- Il voto assegnato al termine della discussione sarà individuale
  - Nel verificare la soluzione proposta, verranno considerate le competenze e le capacità acquisite dai singoli componenti del gruppo di lavoro (sia relative al linguaggio C e all'interazione con la shell, sia relative agli argomenti studiati nei moduli di laboratorio)
- Per concordare la data della discussione del progetto di laboratorio è necessario contattarmi via email scrivendo a [dicaro@di.unito.it](mailto:dicaro@di.unito.it) con oggetto "Discussione Lab Unix T3"

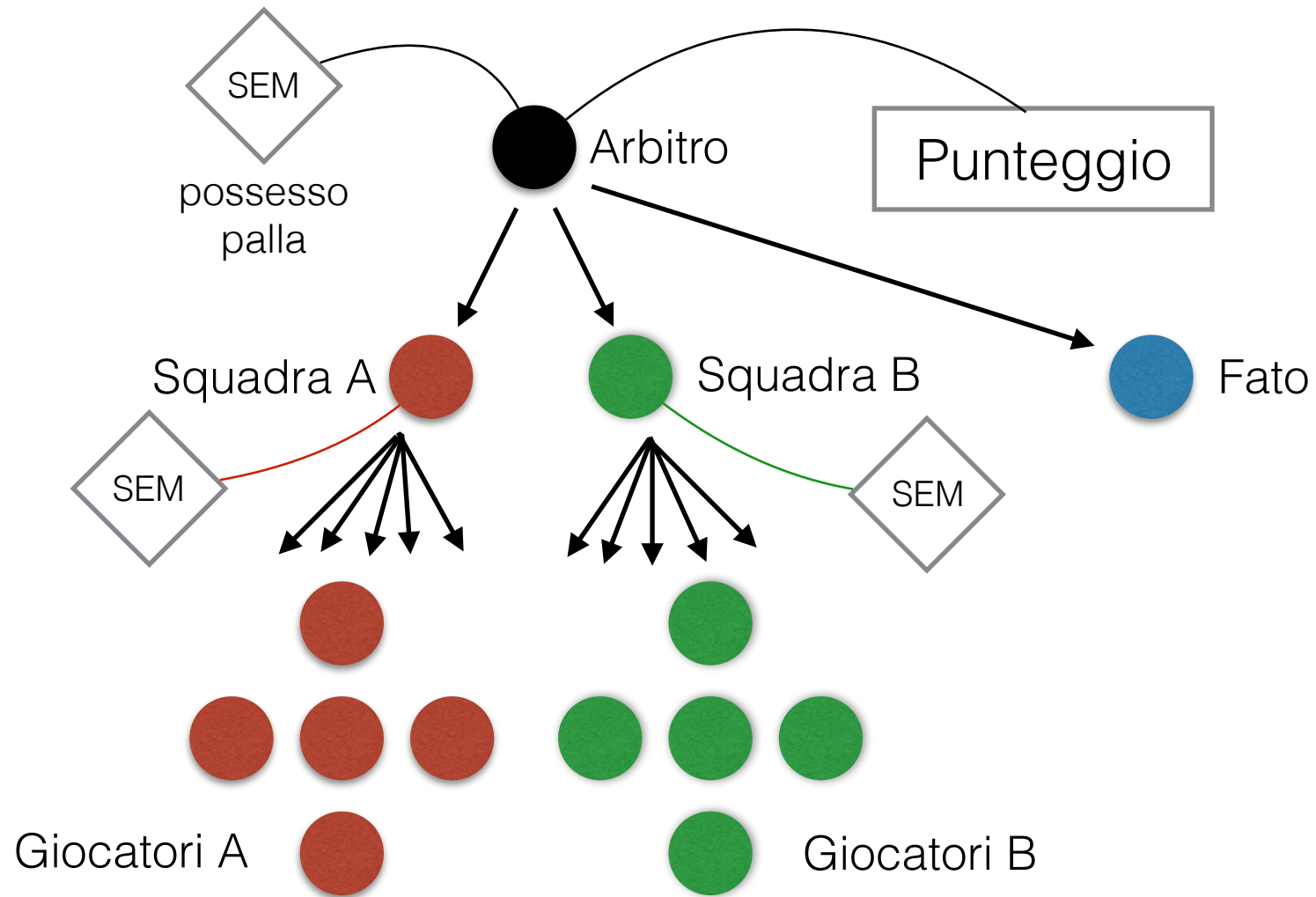
## **Plagio**

- Non saranno tollerati episodi di plagio, anche parziale, di laboratori già discussi e consegnati: in questo caso sarà annullato il laboratorio (ed eventualmente, se già verbalizzato, del voto dell'intero corso) anche agli autori del lavoro, che si assume siano i primi a consegnarlo e a discuterlo.
- La consegna di una porzione di codice comporta un'assunzione di responsabilità, in base alla quale chi presenta l'esercitazione afferma di essere autore del codice, e autore di tutto il codice del progetto.

## **Registrazione del voto di Sistemi Operativi**

- Vi iscrivete all'appello scritto e al corrispondente orale verbalizzante, che sono fissati nello stesso giorno.
- Se superate lo scritto e decidete di accettare il voto conseguito mandate alla Prof.ssa Baroglio una email in cui richiedete la registrazione del voto.
- Il voto che vi viene registrato sarà la somma tra il voto preso allo scritto che vedete on-line, e il punteggio conseguito nella discussione dell'esercitazione finale.
- Per la registrazione del voto non è quindi richiesta la vostra presenza, è necessaria una email alla prof.ssa Baroglio.
- Se vi siete iscritti ad un appello e poi non vi presentate allo scritto, o vi presentate e vi ritirate prima della consegna non succede niente, semplicemente non viene registrato nulla.
- Se invece consegnate lo scritto, quello conta come uno dei tre scritti che potete consegnare in un anno accademico.
- Una volta passato lo scritto, chi ha preso almeno 24 e desidera sostenere l'esame orale facoltativo, prende un appuntamento con la prof.ssa Baroglio.

# Architettura dei processi



PROGETTO SO-UNIX

2016-2017

**"CALCETTO"**

## Gestione eventi

### Tiro andato a segno

Il giocatore genera un segnale che viene catturato dal processo arbitro che aggiorna il punteggio. Viene rilasciato il semaforo "palla".

### Infortunio

Il processo giocatore termina e decrementa il semaforo della propria squadra. Il processo squadra si accorge che manca un processo giocatore e ne crea uno nuovo. Il semaforo "palla" viene rilasciato.

### Dribbling

Il giocatore ha a disposizione una nuova giocata (e quindi un tiro, un infortunio o un altro dribbling).

## Flusso e comunicazione

Giocatori in loop  
richiesta accesso

