page1image16850816Examen Segundo Parcial

Consignas

La Champions:

Se juega la final del partido de futbol, la Champions.

Este partido de se definirá por penales, y pateará todos los penales Julian y Lautaro.

Cada jugador pateará un penal, y los resultados pueden ser 4 (cuatro):

-Gol

-Fuera

-Palo

-Atajo

El partido termina luego de patear 5 penales cada uno, y el ganador es el que más penales tiene con GOL.

En caso de que los penales terminen empatados (5 a 5) deberán seguir pateando hasta que uno meta GOL y otro erre.

Se contará con dos procesos.

Se debe utilizar threads, cola de mensajes, y para el sincronismo de ser necesario se puede usar mutex o semáforos.

Puede considerar de utilizar memoria compartida si lo precisa para el desarrollo.

ME OLVIDE DE PONER EL MSGCTL EN EL CORRE

typedef enum {

MSG\_NADIE, //0

MSG\_JULIAN, //1

MSG\_LAUTARO, // 2

} Destinos;

typedef enum {  
EVT\_NINGUNO, //0

EVT\_GOL, //1 EVT\_FUERA, //2 EVT\_PALO //3 EVT\_ATAJA //4

} Eventos;

El partido cuenta con estos cuatro (4) eventos, y cada tiro se deberá realizar presionando una tecla por menú. Si no se presiona la tecla, no debe patear.

Los eventos se definen de manera aleatoria. Si se realiza un GOL se debe mostrar por pantalla “GOL de”. Si fue fuera, palo o atajada también se debe mostrar por pantalla la acción que sucedió.

Se debe mostrar lo que sucede en los dos procesos, y cual jugador gano y el resultado.

Se debe comenzar el partido presionando una tecla cuando los dos procesos estén corriendo, o mediante memoria compartida.

Pensar bien que/quien es cada proceso!

Si utiliza memoria dinámica debe liberar la memoria!

Utilizar memoria compartida o un bucle, para siempre poder realizar las inicializaciones correspondientes, y no depender de correr un proceso determinado primero.

Utilizar Makefile con la estructura de archivos vista en la cursada.

Utilizar los delay (sleep) que correspondan.