

Recuperatorio del segundo examen parcial

Consignas

Penales

Se trata de un proceso ARQUERO y otro PATEADORES con varios hilos.

Utiliza cola de mensajes con los siguientes IDs:

```
typedef enum
{
    MSG_NADIE,      //0
    MSG_ARQUERO,    //1
    MSG_PATEADOR,   // 2 Cada hilo le suma uno a este define.
} Destinos;

typedef enum
{
    EVT_NINGUNO,    //0
    EVT_TIRO,       //1
    EVT_GOL,        //2
    EVT_PALO,       //3
    EVT_TRAVESAÑO,  //4
    EVT_FUERA       //5
} Eventos;
```

Proceso ARQUERO (MSG_ARQUERO)

1. Borrar los mensajes anteriores, para empezar de 0.
2. Espera recibir EVT_TIRO de cada pateador.
3. Selecciona al azar un evento (EVT_PALO, TRAVESAÑO, GOL o FUERA).
4. Espera 500 ms.
5. Envía el evento seleccionado en el punto 3 al pateador (remitente).
6. Una vez recibidos **5 tiros** (en total) finaliza y muestra la cantidad de goles recibidos y el número de pateador de los que los hicieron. Si no se llega a 5 tiros, vuelve al punto 2 (es un tiro por pateador).

Proceso PATEADORES

El hilo principal (main):

- Lanza **5hilos**HiloPateador, uno por pateador (son 5 pateadores).
- Se queda esperando que terminen los hilos.

- Al finalizar los hilos se debe mostrar la cantidad de goles. Además, emitir un listado con NRO_PATEADOR RESULTADO_TIRO (GOL, TRAVESAÑO, FUERA, PALO) (uno por cada uno).

HiloPateador (para ordenar los tiros se utiliza un mutex)

1. Espera obtener el MUTEX.
2. Espera 500 ms.
3. Envía EVT_TIRO a MSG_ARQUERO (un solo tiro por pateador).
4. Espera recibir un evento del proceso ARQUERO.
 - Si recibe un mensaje EVT_GOL, escribe por pantalla "GOOOOOL".
 - Si recibe cualquier otro evento, escribe por pantalla "UFFFFFF".
5. Finaliza devolviendo el resultado.

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