page1image16850816Examen final – Primer llamado

Consignas

La carrera de autos:

Se trata de un proceso “circuito” y otro “autos” con varios hilos, uno por cada auto.

Es la primer carrera del año, y se deberá realizar la competencia de autos. La cantidad de autos que corren se recibe por parámetro, al iniciar el proceso.

El circuito tiene un recorrido total de 100 a 1000 metros, definido aleatoriamente al comienzo de la carrera.

Cada auto recorre por turno de 1 a 10 metros, este numero es aleatorio por vez.

Se pueden utilizar los eventos definidos debajo, o los que prefiera (más o menos eventos); el evento inicio si se utiliza, puede recibir la cantidad de metros a recorrer, o estar definido en el proceso circuito.

typedef enum {

MSG\_NADIE, //0

MSG\_CIRCUITO, //1

MSG\_AUTOS, // 2(+nro\_autos)

} Destinos;

typedef enum {

EVT\_NINGUNO, //0

EVT\_INICIO, //1   
EVT\_AVANZA, //2

EVT\_GANO\_BANDERIN //3

} Eventos

Si un auto supera la cantidad de metros que tiene el circuito de la carrera, ese auto gana, y puede enviar el evento EVT\_GANO\_BANDERIN.

Utilizar sleeps entre cada avance de auto (300ms) y para evitar problemas de sincronismo.

Se debe utilizar threads, cola de mensajes, y para el sincronismo de ser necesario se puede usar mutex o semáforos. Recordar borrar los mensajes al inicio y otras inicializaciones.

Puede considerar de utilizar memoria compartida si lo precisa para el desarrollo. Es recomendado que se utilice memoria compartida para que se informe y se controle si primero se ejecuto el proceso de inicialización, o no.

Se debe mostrar por pantalla todo el tiempo los metros recorridos y al finalizar el auto ganador.

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