Final – La tormenta

Consignas

Sistema meteorológico:

Una entidad que controla el clima y la temperatura, quiere desarrollar un sistema para poder medir los rayos que caen, los relámpagos y los truenos.

Los rayos cuando suceden son de 0 a 4 (o sea puede no haber rayos, esto se decide aleatoriamente), mientras que los relámpagos pueden ser de 1 a 3 (también aleatoriamente) y los truenos son de 0 a 1 (o sea puede que no haya truenos y si hay siempre vienen después del relámpago, si puede darle ese sincronismo mejor).

Hay 4 hilos, la lluvia comienza cuando por lo menos hubo 1 trueno.

Se contará con un proceso para el panel y otro proceso con 4 threads para cada evento meteorológico.

Para comunicar puede/debe utilizar cola de mensajes (por ejemplo MSJ\_CLIMA+i, o hardcodeado) y/o memoria compartida y para el sincronismo de ser necesario se puede/debe usar mutex o semaforos.

Cada 400ms debe informar lo que sucede en la tormenta.

Los dos procesos deben ir informando todo lo que va sucediendo.

Usar los eventos que crea necesario.

Ejemplos:

typedef enum {

MSG\_NADIE, //0

MSG\_PANEL, //1

MSG\_RAYOS, // 2

MSG\_TRUENOS, // 3

MSG\_RELAMPAGOS, // 4

MSG\_LLUVIA, // 5

} Destinos;

typedef enum {  
EVT\_NINGUNO, //0

EVT\_INIT, //1

EVT\_INFO, //2

EVT\_FIN, //3

} Eventos

Puede diferir, pero tenga en cuenta que de otro modo, puede afectarse la cantidad de hilos.

El evento fin puede poner la condición que crea necesaria, cuando por lo menos hayan ejecutado todos los hilos alguna/s vez/ces.

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.