



Cuestión 1. Otra vez Indi (8 puntos)

Los amigos de Indiana Jones (*indianos*) deben cruzar el puente colgante huyendo del templo maldito. Les persiguen los seguidores de Mola Ram (*molaRam*). Se quiere simular el comportamiento de los hilos cumpliendo las siguientes restricciones:

- 1) Un *indiano* sólo entra si el puente está vacío o sólo hay indianos cruzando.
- 2) Nunca debe haber más de 4 personas en el puente en total.
- 3) Un *molaRam* entrará en el puente si ya hay al menos un *indiano* cruzando.
- 4) Nunca hay más de un *molaRam* en el puente.
- 5) Si *indianos* y/o *molaRams* están esperando para cruzar, no tiene por qué mantenerse el orden de llegada una vez sea posible cruzar.
- 6) Se deben usar cerrojos y variables condicionales

Codifica el programa completo creando 10 hilos *indianos* y 4 hilos *molaRam*.

```
// Declaración de variables globales (se pueden usar más,claro)
int molRamsEnPuente;
int indianosEnPuente;
pthread_mutex_t mtx;
pthread_cond_t esperando;

void* indiano(void* arg) {
    int index = (int) arg;
    /* comprobar si puede cruzar el puente */
    mientras (hayaMolRamsenPuente || haya4Personas) {
        printf("Indiano %d se queda a la espera\n",index);
        <<< ESPERAR >>>
    }
    /* Otras acciones (COMPLETAR) */

    printf("Indiano %d se entra en el puente\n",index);
    sleep(2);

    /* Acciones de salida del puente (COMPLETAR) */

    printf("Indiano %d sale del puente\n",index);
    si (soyElUltimoENSalirDelPuente)
        <<< AVISAR A TODOS LOS QUE ESPERAN >>
}

void* molaRams(void* arg) {
    int index = (int) arg;
    /* comprobar si puede cruzar el puente */
    mientras (hayaMolRamsenPuente || haya4Personas || noHayaIndianosEnPuente) {
        printf("molRam %d se queda a la espera\n",index);
        <<< ESPERAR >>>
    }
    /* Otras acciones (COMPLETAR) */

    printf("molRam %d se entra en el puente\n",index);
    sleep(3);

    /* Acciones de salida del puente (COMPLETAR) */

    printf("molRam %d sale del puente\n",index);
    si (soyElUltimoENSalirDelPuente)
        <<< AVISAR A TODOS LOS QUE ESPERAN >>
}
```

```

int main(int argc, char* argv[]) {

    pthread_mutex_init(&mtx, NULL);
    pthread_cond_init(&esperando, NULL);

    // Crear 10 hilos para indianos

    // Crear 4 hilos para molaRams

    // Esperar a que finalicen los hilos indianos

    // NO esperamos por molaRams, porque alguno se puede quedar sin pasar

    // Liberar memoria si necesario y finalizar
}

```

Cuestión 2. Generando Log (2 puntos)

En el código anterior, incluye un nuevo hilo que se encargará de escribir en un fichero, cada segundo la siguiente información: número de segundos transcurridos (tal y como los devuelve `time()`), número de *indianos* en el puente en ese momento, número de *molaRams* en el puente en ese momento. La información debe escribirse en el fichero en BINARIO (no en ASCII). Debe usarse la librería estándar (`fopen / fread / fwrite....`)

La estructura del código podría ser similar a:

```

void* thread_log(void* argv) {

    << abrir fichero log.bin. Abrir para escritura, truncando si ya existe >>
    mientras no finalice la simulacion {
        time_t ts = time(NULL);
        << escribir ts (en binario) >>
        << escribir indianosEnPuente >>
        << escribir molaRamsEnPuente >>
        sleep(1);
    }
    << Cerrar fichero >>
    return NULL;
}

```

Asimismo, escribe una nueva aplicación en un fichero llamado `readLog.c` que lea el fichero *log.bin* e imprima por pantalla la información (en ASCII) en un formato similar a :

Timestamp	Indianos	molaRams
12345000:	2	1
12345001:	4	0
.....		

La estructura del código podría ser similar a:

```

int main(int argc,int* argv[]) {
    << abrir fichero log.bin. Abrir para lectura >>
    printf("Timestamp \t Indianos \t molaRams\n");
    mientras no lleguemos a final de fichero {
        << leer ts (en binario) >>
        << leer indianosEnPuente >>
        << leer molaRamsEnPuente >>
        printf(".....");
    }
    << Cerrar fichero >>
    return NULL;
}

```

NOTA: se puede hacer este apartado sin hacer el anterior (aunque no se llegaría al aprobado). Para ello, se podrán tomar valores aleatorios para `indianosEnPuente` y `molaRamsEnPuente`. En todo caso, el código de escritura debe estar en un hilo, no en el programa principal.