

Cuestiones (3 puntos)**Cuestión 1: (1,5 puntos)**

Dado el siguiente código (similar al ejemplo 3 visto en clase):

```
#include <pthread.h>
#include <stdio.h>

pthread_mutex_t mutex;

void imprimir(int *n)
{
    int local;
    pthread_mutex_unlock(&mutex);
    sleep(2);
    local=*n;
    printf("Thread %d %d \n", pthread_self(), local);
}

int main(void)
{
    pthread_attr_t attr;
    pthread_t thid;
    int num=1;
    pthread_attr_init(&attr);
    pthread_attr_setdetachstate(&attr,
PTHREAD_CREATE_DETACHED);
    pthread_mutex_init(&mutex, NULL);
    while(num)
        {
            printf("Escribir numero entero :\n");
            pthread_mutex_lock(&mutex);
            scanf("%d", &num); /* espera */
            pthread_create(&thid, &attr, imprimir, &num);
        }
    pthread_attr_destroy(&attr);
    return(0);
}
```

¿Qué se podría hacer para mejorarlo?

Marque con una cruz una, varias o ninguna de las siguientes opciones:

- a) Usar una variable condición
- b) Cambiar return(0) por pthread_exit(NULL)
- c) Cambiar se sitio local=*n y ponerlo justo antes de pthread_mutex_unlock dentro de imprimir.
- d) Intercambiar entre sí pthread_mutex_unlock y pthread_mutex_lock

Cuestión 2 (1,5 puntos)

Dado el siguiente fichero con extensión .x:

```
const NUM_MAX_TIT=5;
const LON_MAX_NOM=255;
enum codigo_cuenta {
    CORRIENTE=0,
    AHORRO=1,
    VIVIENDA=2
};
union tipo_cuenta switch (codigo_cuenta codigo) {
    case CORRIENTE:
        void;
    case AHORRO:
        float interes;
    case VIVIENDA:
        bool desgrava;
};
typedef string nombre <LON_MAX_NOM>;
struct cuenta {
    nombre titulares <NUM_MAX_TIT>;
    tipo_cuenta tipo;
    double saldo;
};
struct lista_cuentas {
    cuenta info;
    lista_cuentas * siguiente;
};
```

Al pasarle el fichero anterior a rpcgen, se obtiene un fichero que puede ser el siguiente:

```
#include <rpc/rpc.h>
#define NUM_MAX_TIT 5
#define LON_MAX_NOM 255
enum codigo_cuenta {
    CORRIENTE = 0,
    AHORRO = 1,
    VIVIENDA = 2
};
typedef enum codigo_cuenta codigo_cuenta;
struct tipo_cuenta {
    codigo_cuenta codigo;
    union {
        float interes;
        bool_t desgrava;
    } tipo_cuenta_u;
};
typedef char *nombre;
struct cuenta {
    struct {
        nombre *titulares_val;
    } titulares;
    tipo_cuenta tipo;
    double saldo;
};
typedef struct cuenta cuenta;
struct lista_cuentas {
    cuenta info;
    struct lista_cuentas siguiente;
};
typedef struct lista_cuentas lista_cuentas;
```

Marque con una cruz una, varias o ninguna de las siguientes opciones:

- a) Falta una línea: typedef struct tipo_cuenta tipo_cuenta;
- b) Sobra la línea: typedef struct cuenta cuenta;
- c) Falta un * justo delante de siguiente.
- d) Falta un campo que indique la longitud dentro de titulares.
- e) Falta un campo del tipo void dentro de la union.