Ejercicio 1.1 (Manejo de listas)

Una librería necesita conocer los libros que tiene disponibles. Para ello, lleva cargados en memoria los títulos de las diferentes obras que tiene disponible (no existen dos libros con el mismo nombre).

- Mediante el comando: "***ListarLibros***" se debe imprimir en pantalla los datos de cada obra.

- Mediante el comando: "***CargarLibro [Nombre]***" se debe agregar a la lista de obras a *[Nombre]*

Ejercicio 1.2 (Manejo de Estructuras)

La librería ahora necesita guardar más datos para cada libro:
-Nombre
-ISBN (String de 13 caracteres)
-Precio
-Stock

- Actualizar el modelo del ejercicio 1.1 y modificar el comando *CargarLibro* de tal forma que permita agregar un libro con todas las características antes detalladas.

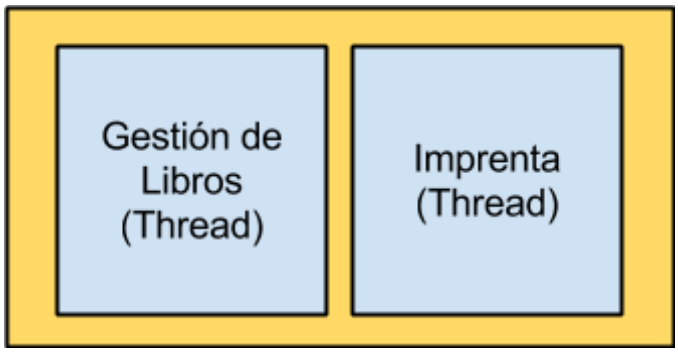
- Añadir el comando "***QuitarLibro [Nombre]***" que permita borrar de la lista el libro indicado en ***[Nombre]***.

- Añadir la comando "***AumentarStock [Nombre] [Unidades]***" y "***DisminuirStock [Nombre] [Unidades]***".

Ejercicio 2 (Threads) – *Necesita 1.2*

Los dueños, sorprendidos del éxito de su local, deciden instalar su propia imprenta, la cual generará libros. Como aún no está relevada la forma de generar libros, pero queremos atacar el problema técnico de realizar concurrentemente la operación, se deberá agregar un thread que cada un tiempo especificado por archivo de configuración imprima el mensaje "generando libros…"

El archivo de configuración se llamará ***"biblio.conf"*** donde estará el tiempo entre cada generación de libros, el tiempo se encuentra especificado en segundos:
imprenta_tiempo_generacion=60



Ejercicio 3.1 (Sincro: Productor/Consumidor)

Nuestro analista ha hecho un genial trabajo, y nos envió la siguiente descripción:

“Se deberá generar, en el tiempo especificado anteriormente, una cantidad X de libros con los siguientes datos:

-Nombre: pepin_[numero_libro_producido]
-ISBN: “AAAA-AAA-AAAA” (Si, todos iguales, no es un ID)
-Precio: 1 - X
-Stock: X – 1

La cantidad se especifica a través del archivo de configuración, con el parametro imprenta_cantidad_libros_por_tanda

Por ejemplo, si en el archivo de configuración se indica la impresión de 3 libros y ya se han impreso 12 libros, se deben generar los libros:

-pepin_13 AAAA-AAA-AAAA, 1, 3
-pepin_14, AAAA-AAA-AAAA, 2, 2
-pepin_15, AAAA-AAA-AAAA, 3, 1”

- Dicha imprenta funcionará como un Thread separado al registro de libros (para que se pueda imprimir y acceder a la vez). Cada vez que la imprenta termine de imprimir un libro, lo cargará en una estructura global para que se encuentre disponible al resto del sistema.

- Además, se agrega un Thread Depósito. Dicho depósito deberá obtener de la estructura global los libros a medida que vayan saliendo de la imprenta y los mostrará por pantalla, luego lo dejará en otra lista.

- Realizar un diagrama de la estructura final del programa.

Nota: Analizar las diferentes zonas críticas, sincronizado de tal manera que todos los sistemas funcionen correctamente (tener en cuenta que la imprenta no puede guardar un libro en la estructura global mientras el depósito está leyendo dicha estructura, ya que se pueden generar inconsistencias).

Ejercicio 3.2 – *Necesita 2*

Integrar el sistema de gestión de libros con el depósito y la imprenta. Implementar el comando "***TraerDeDeposito***" que cargue en el thread de gestión los libros que se encuentran actualmente en el depósito.

Nota: Sincronizar de tal manera que el depósito no se encuentre obteniendo libros de la imprenta cuando se están trayendo los libros al sistema de gestión.

Ejercicio 4.1 (Sockets)

El local creció tanto que los dueños tuvieron que comprar el negocio de enfrente para poder sostener el afluente de gente. Los empleados, cansados de verse constantemente involucrados en accidentes de tránsito al tener que cruzarse para poder hablar, nos demandan la creación de un sistema de chat simple.

Se deberá crear un programa nuevo, que pueda comunicarse con otra instancia del mismo, mediante TCP/IP. Cuando una instancia reciba un mensaje, imprimirá su contenido en pantalla. Cuando el usuario escriba un mensaje y presione <*enter*>, deberá enviárselo a la otra instancia.

Consejo: Pensar qué pasará cuando el programa esté esperando un input por pantalla y reciba un mensaje ¿Lo mostrará? ¿Por qué? En el caso que no, ¿Cómo lo solucionarías? ¿Y cuando espere un mensaje, se dará la situación inversa?.

Ejercicio 4.2

Los empleados, en su ánimo de moverse cada vez menos, desean tener un chat global que se encuentre en todas las computadoras de los locales. Para ello, contaremos con una computadora que actuará de servidor, a la cual se le conectarán los distintos clientes de chat.

Todos los clientes de chat serán parte de la misma conversación. Cuando un cliente envíe un mensaje, lo hará al servidor, y se reproducirá en todos los clientes (incluido él mismo).
El servidor debe recepcionar dicho mensaje y luego enviárselo a todos los clientes.

Realizar un diagrama de conexiones y envío de mensajes.

Ejercicio 4.3 (Serialización)

Bajar el programa de: *Comming Soon*

Este programa es un proceso servidor, cuando alguien se le conecte en el puerto 6660, le enviará un mensaje de saludo y luego una estructura serializada con los últimos 10 mensajes recibidos, los mensajes tendrán el siguiente formato:

Saludo:
```
{
  tipo: uint8_t (1 = saludo)
  largo: uint8_t (largo del mensaje, sin incluir el \0)
  mensaje: 64 chars
}
```

Mensajes (10 estructuras consecutivas):

```
{
   tipo: uint8_t (2 = mensaje)
   largo: uint32_t (longitud del resto de los datos)
   hora: uint8_t
   minuto: uint8_t
   segundos: uint8_t
   mensaje: variable (tamaño = largo - sizeof(hora) - sizeof(minuto) - sizeof(segundos)
}
```

Si el cliente desea enviar un mensaje al servidor para que lo almacene, deberá respetar la misma estructura definida arriba, este lo propagara a todos los clientes que tenga conectados en el momento.

Ejercicio 5.1 (Integrador) – *Requiere 3, Requiere 4*

Los dueños se enteraron del chat que están usando los empleados y, gracias a su visión capitalista, se dieron cuenta que podían aplicarlo al sistema de gestión.

Como método de expansión, compraron distintos locales para cada uno de los sectores, de tal manera que ahora se encuentran todos geográficamente separados, exceptuando por la Imprenta y el Depósito que siguen estando juntos. Si, tienen un local en el que se encargan, como sede administrativa, de controlar a los empleados.

- Generar, a partir del punto 3, los procesos Gestión de Libros y Producción, que se comuniquen entre sí a través del protocolo TCP/IP. Producción incluye a los Threads Imprenta y Depósito.

- Realizar un diagrama en el cual se pueda analizar el intercambio de mensajes.

Nota: Los procesos deben cumplir con los mismos requerimientos que se han dispuesto hasta el punto 3.