

Estructura de dato: Cola

Una Cola o Queue (en inglés) es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción se realiza por un extremo y la operación de extracción por el otro. Este tipo de estructura se le conoce como un tipo FIFO (First In – First Out); Primero en entrar, primero en salir. Esto quiere decir, que, si metemos un primer elemento, este será el primero en retirarse con la operación de extracción. Podemos verlo como una fila de las tortillas, el primero en ser atendido, será el primero que llegó, y así con más personas que lleguen.

La particularidad de una estructura de datos de cola es el hecho de que sólo podemos acceder al primer y al último elemento de la estructura. Así mismo, los elementos sólo se pueden eliminar por el principio y sólo se pueden añadir por el final de la cola.

Cola estática

Una cola estática es similar a una pila estática, este tiene un tope de elemento que se pueden almacenar. Como notamos en nuestro archivo **cola.h** y donde la aplicación de este está en **colaEstatica.c**, esta cola se apoya de un arreglo de un tamaño específico, donde nosotros vamos a guardar los valores de dicha cola. Veamos más en concreto nuestra estructura que está definida de la siguiente forma:

```
typedef struct{
    int queue[maxCola];      // Cola de un tamaño de 10 elementos
    int pri;                 // Índice del primer dato de la cola
    int ult;                 // Índice del último dato de la cola
    int elemCola;            // Número de elementos insertados en la cola
}Cola;
```

Algoritmo 1. Estructura de la cola estática

Como notamos en el Algoritmo 1 tenemos una estructura, la cola el cual contiene un arreglo, con un número máximo que en este caso es de 10 elemento. La variable *pri* representa una variable el cual nos indica que es el “apuntador” al primer elemento de la cola y *ult* es un “apuntador” al último elemento de la cola, es decir, que cuando ingrese un nuevo elemento, este se va a dirigir a él, o bien, guardará el índice del último elemento. Y por último un contador de elementos que tiene la cola actualmente.

Ahora notemos algo, esta definición de estructura es un poco diferente a las que hemos estado trabajando, en este caso, usamos la palabra reservada typedef, su función es asignar un nombre alternativo a tipos existentes, a menudo cuando su declaración normal es aparatosa, potencialmente confusa o probablemente variable de una implementación a otra. En otras palabras, podemos llamar a una variable de un tipo de dato cualquiera en el código, o bien, en nuestro caso al momento de instanciar nuestra estructura, declararla como una variable, es decir en lugar de poner en el *main*: `struct Cola cola00;`, pondremos la línea `Cola cola00;` Que podemos verlo como si fuera un tipo de variable Cola, esto a simple vista puede ser un poco más sencillo y rápido detectar ese tipo de declaraciones tanto para una instancia normal como el apuntador de ello.

No se explicará en esta parte las funciones, ya qué, al ser una estructura estática, es un poco más fácil intuir que es lo que está haciendo la estructura, además de que su tarea es describir las funciones, en el código de se describe un poco mejor la tarea.

Una vez hecho esto, avisarme para poder actualizar de nuevo el repositorio.

Cola dinámica

Ahora, veamos más o menos como será la estructura de nuestra cola según nuestro programa ***colaDinamica_errores.c***.

Primero veamos nuestro nodo (Figura 1), imaginemos que tenemos una caja que tiene dos compartimentos, estos son el dato, qué, si lo vemos bien, no importa qué tipo de dato sea, puede ser algo abstracto, tanto pude ser un dato primitivo, o bien puede ser algo más complejo, como otra estructura que contenga varios tipos de datos primitivos.

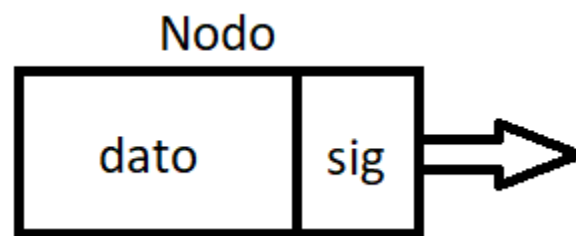


Figura 1. Nodo

La otra sección de la caja es un apuntador al mismo tipo de estructura, que es nuestro nodo, es decir, esta variable (llamada *sig*) tiene la capacidad de dirigirse o apuntar a otro nodo. Esto nos ayuda a saber quién es el siguiente en la cola; volvamos a retomar la fila de las tortillas. Digamos que en la fila están tres personas: Abner, Brenda y Carlos. Ellos llegaron a la fila en esa forma, comparando con una cola, cada persona es un nodo y su nombre puede ser el dato, ahora, ¿cuál es el apuntador? Realmente en las personas no tenemos algo que simule esto, pero podemos decir que el conocimiento de la persona que está antes de nosotros es el apuntador. Es decir, que el apuntador hace que Abner sepa que Brenda está detrás de él y que Brenda sepa que detrás de ella está Carlos.

Ahora veamos la otra estructura, que la llamamos *ptrCola* (Figura 2), el cual, contiene dos apuntadores del tipo nodo, es decir, va a dirigirse solamente a tipos de dato estructura nodo. Volvámoslo a ver como una caja, con dos compartimentos, uno de esos compartimentos, solo va a poder apuntar al primer nodo de la cola, y el otro compartimento al último nodo de la cola.

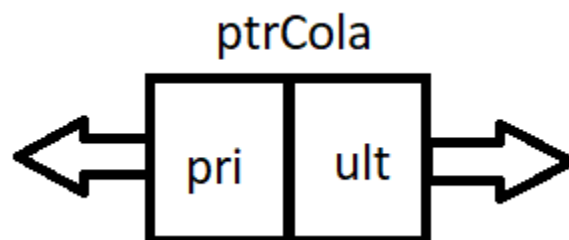


Figura 2. ptrCola

De nuevo vamos a ver nuestra fila de las tortillas con los mismos clientes, digamos que nuestro vendedor de tortillas será *ptrCola* y lo único que sabe son los nombres del primer cliente y del último, es decir, de Abner y de Carlos, pero ¿qué pasa si llega alguien más? Veamos, vamos a decir que llega un nuevo cliente, llamado Daniela. Aquí pasan dos cosas, uno es que Carlos, en lugar de no tener a nadie atrás ahora lo tiene, entonces, el apuntador *sig* de Carlos verá al nuevo cliente, que es Daniela. La segunda cosa por ocurrir es que el vendedor, en lugar de ver a Carlos, ahora ve a Daniela para saber quién es el último de la fila, entonces el vendedor solo verá a Abner y a Daniela.

Regresando a la estructura, entonces *ptrCola* son vistas al último y primer nodo que hay en la cola, por lo que podemos verlo más o menos como se muestra en la figura 3. Algo que mencionar, es que con otros autores y fuentes de información no existe *ptrCola*, simplemente son los apuntadores *pri* y *ult* sueltos, no están en una caja como nosotros lo estamos viendo. Esto lo hacemos solo para verlo con ejemplos que se ven en la vida diaria, como en la fila de las tortillas.

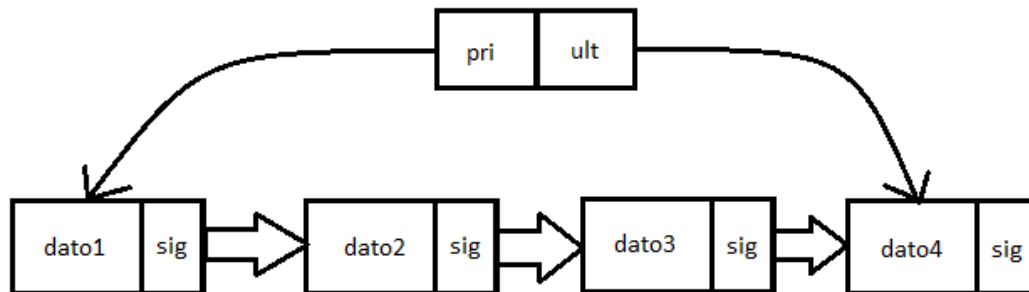


Figura 3. Estructura Cola

Ahora podemos ver las funciones meter, sacar y recorrer, pero esto lo podemos entender leyendo el código, ya que sabemos apuntadores, pero explicaremos un poco más a fondo la función meter como ayuda extra.

Iniciamos con los apuntadores *pri* y *ult* con valor NULL, y queremos ingresar un dato, como estos apuntadores están vacíos, el primer nodo entrante será el primer y último dato al mismo tiempo, es decir que *pri* y *ult* apuntan al nodo entrante, este paso es el fácil. Ahora bien, ingresemos otro nodo, pero la cola ahora tiene un dato, ¿qué se tiene que hacer?

Bueno, como vemos en la Figura 4, nosotros tenemos que dirigirnos con el apuntador *ult* de la estructura *ptrCola* al último dato que hay en la cola, para poder saber quién es el último en la fila, entonces, nos tenemos que dirigir con a ese apuntador, ya que él es el que tiene conocimiento de ese último dato que entro, no nos sirve ir al primero, ya que siempre será el primero y si tenemos más datos de entrada, no sabremos nunca quien es el último, sino el dato siguiente del primer dato de la cola.

El segundo paso, es dirigir ese apuntador *ult* al nuevo dato, que es el paso más fácil y así saber que realmente ese es el nuevo dato entrante, para poder finalizar con una cola con más de un elemento. Retomando la Figura 4, la primera parte son los pasos que se tienen que realizar, y después el resultado que se debe obtener.

Para la función de desplazar, solo se toma un apuntador auxiliar a donde está apuntando *pri*, para después poder guiarse con el dato siguiente que hay de ese nodo primero y llegar al último nodo.

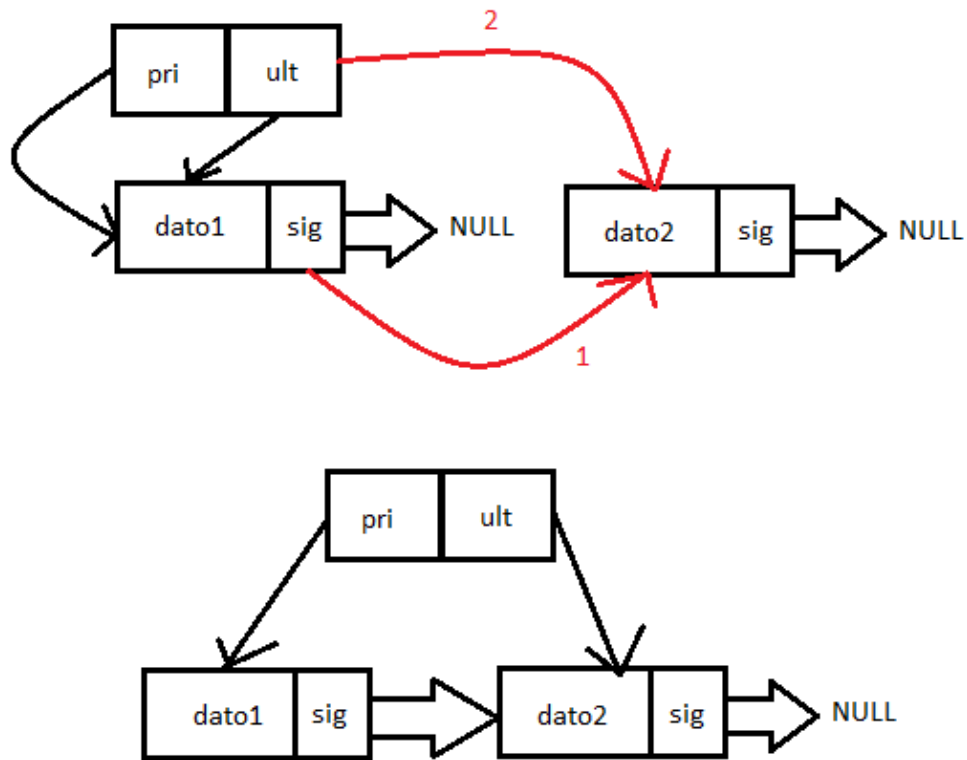


Figura 4. Ingresar un dato a la cola

Ahora, hacer la tarea correspondiente a ***colaDinamica_errores.c***. Dentro del archivo, están a detalle las instrucciones.