



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: TISTA GARCÍA EDGAR

Asignatura: ESTRUCTURAS DE DATOS Y ALGORITMOS I

Grupo: 1

No de Práctica(s): 6

Integrante(s): GÓMEZ LUNA ALEJANDRO

*No. de Equipo de
cómputo empleado:* 34

No. de Lista o Brigada: 14

Semestre: 2019-2

Fecha de entrega: 25/MARZO/2019

Observaciones:

CALIFICACIÓN: _____

- **Objetivo de la práctica**

Revisarás las definiciones, características, procedimientos y ejemplos de las estructuras lineales cola circular y cola doble, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

- **Desarrollo**

Guía: En la guía se empieza abordando el concepto de cola, así como las operaciones que se pueden realizar dentro de esta. Una vez que se ha realizado esta breve introducción, se comienza a definir como funciona la cola circular. Encontramos que lo que cambia principalmente es la manera en como se van almacenando y eliminando los datos en la cola circular, ya que ahora los elementos se pueden almacenar en memoria antes del primer elemento de la cola. En clase se explicó de mejor manera cómo es que la cola circular hace un mejor uso de la memoria, ya que en la guía no se realiza la representación gráfica de que sucede cuando se eliminan objetos de una cola simple, por lo que la explicación en clase fue más detallada. De igual forma, tanto en clase como en la guía, se ofrecen algunos ejemplos del empleo de la cola circular. Posteriormente, se define el funcionamiento de la cola doble, en dónde ahora se pueden insertar y retirar elementos de ambos lados de la cola, por lo que las funciones ya definidas no cambian, lo único que se tiene que especificar es si se quiere realizar la función al inicio o fin de la cola. La explicación tanto en la guía como en clase es prácticamente igual.

Para la cola circular y la cola doble, en la guía se plantean y manejan como si los elementos estuvieran relacionados, es decir, se tratan como si fueran una especie de lista ligada. Por último, en clase también se habló de que se pueden realizar colas circulares dobles, las cuales no son ni siquiera mencionadas en la guía, siendo así que en clase se profundizó más en el tema de colas.

Ejercicios propuestos

1) Actividad 1: Para el programa proporcionado, se solicita el tamaño que tendrá la cola doble circular. Para este caso, se ingresa un tamaño de 10. Una vez ingresado el tamaño, se procede a crear la cola doble circular, donde se le pasa como parámetro el tamaño que tendrá.

Ya que se ha creado la lista, se realizan diversas operaciones sobre ella, en el siguiente orden:

1. Se encolan dos elementos al final de la cola doble y se muestran las posiciones de cada localidad de memoria dentro de la memoria

dinámica (la cual es la lista de elementos), así como el valor que hay dentro de cada una. Como se está utilizando la función `calloc`, se observa que todas las posiciones que no tengan un valor almacenado se inicializan con cero. Asimismo, se muestra la posición en la que se encuentra el índice del primer elemento, el índice del último elemento y la cantidad de espacios disponibles para poder almacenar elementos. Para este caso, el índice del primero es menor que el índice del último.

2. Para la segunda parte, se desencolan los elementos que se almacenaron en el paso anterior. Posteriormente, se muestran las posiciones con sus elementos de la cola doble. Puesto que se desencolaron los elementos que ya estaban almacenados, se observa que todos los elementos de la cola doble son ceros. También se muestra que el índice del último elemento es menor que el índice del primero, debido a que no existen elementos dentro de la cola doble.
3. Para la última parte, se encolan dos elementos al inicio y dos elementos al final, sin embargo, casi instantáneamente se desencola un elemento al final, por lo que, cuando se muestran los valores que están dentro de la cola doble, solo se observa un elemento al final y dos elementos al inicio. De igual forma, se aprecia que el índice del primer elemento es mayor al índice del último elemento.

Para la biblioteca proporcionada, se pueden observar las siguientes funciones:

- `crearCola()`: Esta función toma como parámetro el tamaño que tendrá la cola. El índice de primero es 1 y el índice de último es 0. La cantidad de espacios disponibles será igual al tamaño proporcionado. Por último, haciendo uso de la función `calloc`, se reserva la memoria necesaria para la lista interna de la cola doble circular.
- `IsEmpty()`: Verifica que la cola tenga elementos. Si no tiene retorna valor de 1, en caso contrario retorna valor 0.
- `encolarFinal()`: Toma como parámetros la cola en donde insertará el valor, y el valor a insertar. Primero se verifica que la cola tenga espacios disponibles para posteriormente poder encolar el elemento al final. En segundo lugar, se almacena el elemento calculando el módulo del tamaño con el índice del último, para que cuando el índice de último sea igual al tamaño, el índice de último vuelva a ser cero, debido a que la cola es circular. Por último se decrementa el valor de disponibles, pues ya se ha ocupado un nuevo espacio.

- `encolarInicio()`: Toma como parámetros la cola en donde insertará el valor, y el valor a insertar. Se realiza una comparación con el índice de último, para poder insertar el elemento al inicio de la cola, ya que también es circular. Después se realiza otra comparación para determinar si la cola está llena. Luego, se realiza una última comparación para hacer que el índice de primero marque siempre el inicio de todos los elementos en la cola. Por último, se decrementa el valor de disponibles, pues ya se ha ocupado un nuevo espacio.
- `desencolarInicio ()`: Toma como parámetro la cola en donde se realizará la operación. Se verifica que la cola no esté vacía, en caso de estarlo se muestra un mensaje en pantalla. Una vez hecho esto, se incrementa el valor de disponibles, debido a que un elemento ha sido desencolado. Luego se guarda el valor en una variable auxiliar y el valor original se hace cero. Ya que se ha hecho esto, se hacen otras comparaciones para determinar la posición exacta en donde hay que desencolar el elemento. En caso de que el índice de primero y último coincidan, entonces se crea una nueva cola, debido a que no existen más elementos dentro de la cola. En último lugar, se retorna -1 en caso de que la cola esté llena o el elemento que se desencoló.
- `desencolarFinal()`: Toma como parámetro la cola en donde se realizará la operación. Se verifica que la cola no esté vacía, en caso de estarlo se muestra un mensaje en pantalla. Una vez hecho esto, se incrementa el valor de disponibles, debido a que un elemento ha sido desencolado. Luego se guarda el valor en una variable auxiliar y el valor original se hace cero. Después, se decrementa el valor de último, ya que se ha desencolado un elemento al final. Ya que se ha hecho esto, se hace una comparación para saber si el índice de último es cero. Si el índice de primero coincide con el de último, se crea una nueva cola, en caso contrario el índice de último pasa a ser el tamaño. En último lugar, se retorna -1 en caso de que la cola esté llena o el elemento que se desencoló.
- `mostrarValores()`: Muestra cada posición de la cola doble circular, junto con el elemento en cada posición.
- `mostrarIndices()`: Muestra el valor del primero y último índice, así como la cantidad de espacios disponibles.

En la biblioteca proporcionada solamente se encontraron dos errores, los cuales consistían en que no se retornaba un valor cuando se entraba en una condición, es decir, como se cumplía la condición de que la cola estaba llena, solamente se mostraba un mensaje en pantalla, pero se necesitaba también retornar un valor.

```
Ingrese el tamaño
10
Posición 1    valor 11
Posición 2    valor 12
Posición 3    valor 0
Posición 4    valor 0
Posición 5    valor 0
Posición 6    valor 0
Posición 7    valor 0
Posición 8    valor 0
Posición 9    valor 0
Posición 10   valor 0
Primero = 1
Ultimo = 2
Disponibles = 8
Primera Ejecucion*****
Posición 1    valor 0
Posición 2    valor 0
Posición 3    valor 0
Posición 4    valor 0
Posición 5    valor 0
Posición 6    valor 0
Posición 7    valor 0
Posición 8    valor 0
Posición 9    valor 0
Posición 10   valor 0
Primero = 1
Ultimo = 0
Disponibles = 10
Segunda Ejecucion*****
Posición 1    valor 13
Posición 2    valor 0
Posición 3    valor 0
Posición 4    valor 0
Posición 5    valor 0
Posición 6    valor 0
Posición 7    valor 0
Posición 8    valor 0
Posición 9    valor 9
Posición 10   valor 10
Primero = 9
Ultimo = 1
Disponibles = 7
Tercera Ejecucion*****
sh: PAUSE: command not found
Program ended with exit code: 0
```

All Output ↕

Filter



2)Actividad 2:

- a) En la estructura final de la cola, solamente quedan los valores de 45, 55, 65, 75 y 85. Los demás elementos, que son los tres primeros, se encuentran con ceros, ya que se desencolaron dichos elementos. La cantidad de espacios vacíos o disponibles es tres. El valor de primero es cuatro y el valor de último es 8.

```
Ingresa el tamaño
8
Posición 1    valor 0
Posición 2    valor 0
Posición 3    valor 0
Posición 4    valor 45
Posición 5    valor 55
Posición 6    valor 65
Posición 7    valor 75
Posición 8    valor 85
Primero = 4
Ultimo = 8
Disponibles = 3
Program ended with exit code: 175
```

- b) En la estructura de la final de la cola, solamente quedan los valores de 45, 55, 65 y 85. El número 75 no se pudo insertar, ya que la cola se encontraba llena, como muestra el mensaje. Los demás elementos se encuentran con ceros, ya que se desencolaron con dichos elementos, sin embargo, en la posición número 1 debería de haber un cero, pero como la lista es circular, entonces el último valor encolado, que es 85, se encuentra al «inicio» de la cola. El valor de primero es 4 y el valor de último es 1.

```
Ingrese el tamaño
6
La cola está llena
Posición 1    valor 85
Posición 2    valor 0
Posición 3    valor 0
Posición 4    valor 45
Posición 5    valor 55
Posición 6    valor 65
Primero = 4
Ultimo = 1
Disponibles = 2
Program ended with exit code: 175
```

All Output ↕

Filter



3) Actividad 3: Para esta actividad, se utilizó la biblioteca colaDoble.h (ya corregida), la cual fue proporcionada desde un inicio. Dentro de la función principal, es decir, la función main, se creó una cola con un tamaño de 8, ya que es la máxima cantidad de elementos en cada estado. A continuación, se da una breve descripción de cada estado, así como el código para cada parte.

- Estado 1: Se tomó el valor de 103 como el primer elemento. Dado que es el primer elemento, es indistinto si se usa la función encolarInicio o encolarFinal.

```
printf("*****ESTADO 1*****\n");
encolarInicio(&superCola, 103);
encolarInicio(&superCola, 104);
encolarInicio(&superCola, 105);
encolarFinal(&superCola, 102);
mostrarValores(superCola);
```

- Estado 2: Se tomó como pivote a los valores de 103 y 104, ya que cada uno se encontraba en cada extremo de la cola circular, por lo que resultaba más fácil poder realizar las operaciones a partir de éstos.

```

printf("*****ESTADO 2*****\n");
desencolarFinal(&superCola);
desencolarInicio(&superCola);
encolarInicio(&superCola,120);
encolarInicio(&superCola, 121);
encolarFinal(&superCola, 99);
encolarFinal(&superCola, 98);
encolarFinal(&superCola, 97);
mostrarValores(superCola);

```

- Estado 3: Para este último estado, se desencolaron los elementos antes del valor 103 y se encoló el valor 122 al inicio.

```

printf("*****ESTADO 3*****\n");
desencolarFinal(&superCola);
desencolarFinal(&superCola);
desencolarFinal(&superCola);
encolarInicio(&superCola, 122);
mostrarValores(superCola);

```

Para cada estado, se imprimían los valores de la lista circular doble en pantalla, para corroborar que se estuviera realizando de manera adecuada el llenado. Con la finalidad de lograr lo anterior, se utilice la función `mostrarValores`.

```

*****ESTADO 1*****
Posicion 1   valor 103
Posicion 2   valor 102
Posicion 3   valor 0
Posicion 4   valor 0
Posicion 5   valor 0
Posicion 6   valor 0
Posicion 7   valor 105
Posicion 8   valor 104
*****ESTADO 2*****
Posicion 1   valor 103
Posicion 2   valor 99
Posicion 3   valor 98
Posicion 4   valor 97
Posicion 5   valor 0 |
Posicion 6   valor 121
Posicion 7   valor 120
Posicion 8   valor 104

```



```
*****ESTADO 3*****  
Posicion 1    valor 103  
Posicion 2    valor 0  
Posicion 3    valor 0  
Posicion 4    valor 0  
Posicion 5    valor 122  
Posicion 6    valor 121  
Posicion 7    valor 120  
Posicion 8    valor 104  
Program ended with exit code: 0
```

All Output ↕

Filter



- Conclusiones

Para esta práctica se pudieron cumplir los objetivos propuestos. Se pudo observar como poder emplear lo que se vió en clase en el lenguaje C, así como el funcionamiento de las funciones que se emplean para una cola circular y una cola circular doble.

Los conceptos son bastante útiles para poder resolver problemas de mayor complejidad, además de que son bastante utilizadas en los videojuegos o en aplicaciones, como es el caso de las redes sociales, en donde se almacena contenido que nosotros compartimos, etc. La única posible desventaja es que el manejo de estas estructuras es un tanto complicado, de manera que se debe de comprender muy bien el funcionamiento, lo que realiza cada función para poder emplearlo de una manera correcta.

Por otro lado, se comprendió de mejor manera las características con las que cuentan estas estructuras, debido a que al momento de implementar la estructura es cuando te das una mejor idea de las características y diferencias que existen entre estructuras, para este caso fueron las diferencias entre una cola circular y una cola circular doble.

La práctica permitió visualizar de una muy buena manera el como se emplean estas estructuras en el lenguaje C, porque ya se nos había proporcionado la biblioteca donde venían las funciones que permiten manejar una cola doble simple. Cabe resaltar que, aunque ya se nos hubieran proporcionado las funciones, se necesitaba prestar bastante atención para determinar el funcionamiento de cada función, el porqué se realizaban ciertas acciones, etc.