



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Computación II (CI2126)

NOMBRES: _____

APELLIDOS: _____

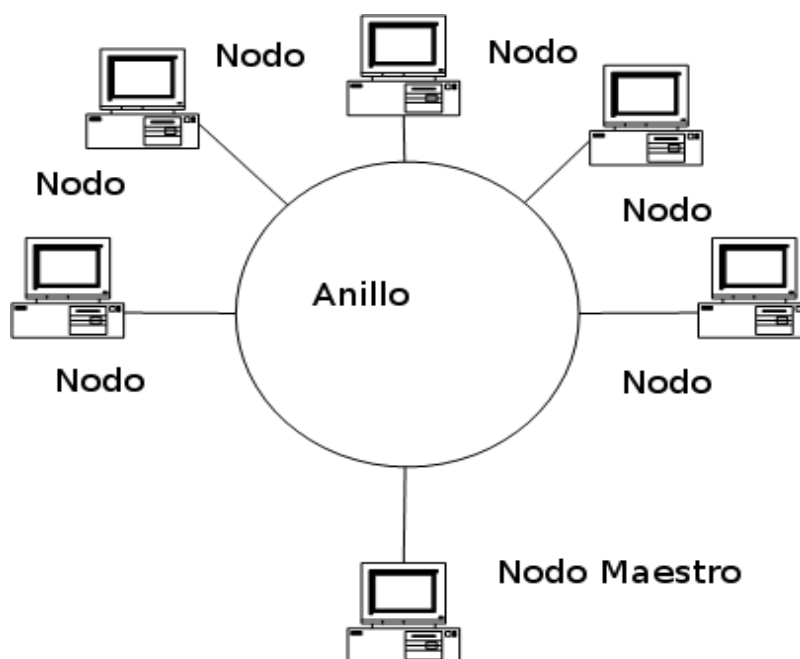
CARNET: _____

SOLUCIONES DEL SEGUNDO EXAMEN PARCIAL

ENUNCIADO:

1.- DESCRIPCIÓN GENERAL:

Suponga una red de computadores del tipo área local (LAN) del tipo anillo. La misma opera sin conexión a ninguna otra red externa a ella misma, es decir, los computadores que la conforman, los cuales reciben la denominación de *nodos*, se comunican únicamente entre sí, están organizados en forma circular y comparten un cable de comunicación común, tal como muestra la siguiente figura:



Por otra parte, para evitar que en el canal circular compartido todos los nodos transmitan información al mismo tiempo y se originen conflictos con los mensajes, se ha acordado que para que cualquier nodo pueda hacerlo, el mismo deberá poseer un elemento llamado “token”. El token es pasado en forma consecutiva y por un lapso de tiempo igual, a cada uno de los nodos que están encendidos y operativos en la red. De modo que si un nodo recibe el token de parte de su predecesor, es decir del nodo que le precede, transmite hasta que expire su lapso de tiempo y le pasa el token al nodo que le sigue a efectos de que este pueda transmitir. Así hasta que el token, regrese a el, de parte de su predecesor. El token se pasa en el sentido de las agujas del reloj.

Para asegurarse que el token no se pierda en caso de que en un instante dado, el nodo que lo posea se apague o quede fuera de la red circular, existe un nodo especial denominado “nodo maestro”. Este nodo especial vigila el funcionamiento de toda la red y en caso de que se pierda, es capaz de regenerar el token.

2.- INICIALIZACIÓN DE LA RED:

Al principio, el primer nodo que se encienda en la red, asume el rol de “nodo maestro” y posteriormente, autorizará cualquier otro ingreso a dicha red. El nodo maestro siempre deberá tener por dirección el valor de “cero” (0). Considere además que cada nodo tendrá una única dirección que no se repetirá. Esta situación es la única en que la dirección del predecesor es igual a la del nodo que lo sucede.

3.- INGRESO DE UN NODO A LA RED:

Después de que el primer nodo haya ingresado y haya tomado el rol de nodo maestro, cualquier otro nodo que desee ingresar a la red, deberá entonces solicitar la autorización del “nodo maestro”. Este último, le suministrará cualquier dato como número de identificación -un número que es su dirección en el anillo-, quien le precede -la dirección- y quien le sigue -otra dirección-. Para hacer eso, el nodo maestro dispondrá de una lista de direcciones disponibles y cada vez que otorgue alguna, la colocará en la lista de direcciones usadas.

4.- RETIRANDO UN NODO DE LA RED:

Si un nodo se apaga apropiadamente, entonces solicitará autorización al “nodo maestro” para retirarse. Éste lo autorizará e informará a su predecesor y sucesor para que se enlacen directamente. Adicionalmente agregará la dirección que el nodo había usado en la lista de direcciones disponibles. En caso de que el nodo haya sufrido alguna situación que lo deje afuera por más de 30 segundos, el “nodo maestro” dará el hecho como un retiro accidentado e iniciará el protocolo para que los nodos que le eran contiguos se interconecten y liberen la dirección. Si el nodo que sufrió la avería se reincorpora, deberá ingresar como si fuera un ingreso por primera vez.

5.- APAGANDO LA RED:

Para desactivar la red por completa, esto se hará como una sucesión continuas de retiros de nodos, dejando al “nodo maestro” como el último. Apagar la red no implica destruirla.

6.- DESACTIVACION ANORMAL DE LA RED:

Si el nodo maestro sufre algún tipo de avería y sale de la red por más de un minuto, los nodos restantes iniciarán el proceso de apagando la red.

7.- TRANSMISIÓN DE DATOS EN LA RED: Cuando un nodo que está en la red tipo anillo, desea transmitir un mensaje a otro nodo que también está activo, debe esperar a que le toque el turno de recibir el token y procederá a enviarle el mensaje con la dirección del nodo destinatario. Al poner el mensaje en el anillo, este automáticamente se propagará como señal electrónica por todo el círculo hasta llegar al mismo nodo que lo emitió, que cortará dicha propagación para que no se repita. Por otra parte, la señal transmitida podrá ser escuchada por todos los nodos del anillo, quienes verificarán si la dirección del destinatario es la suya. En caso de que sea así, tomarán el mensaje y lo procesarán, pero si ese no es el caso, descartarán la lectura del mismo.

8.- MENSAJES EN LA RED: Cuando un mensaje va a ser transmitido este puede ser de dos tipos, o se trata del token, o es un mensaje que va de un nodo a otro. El formato para enviar un token sigue el siguiente formato: *(0, dirección del nodo al cual le es entregado el token)*

Y para el caso en que el mensaje no es un token pero si es una transmisión de datos sigue la siguiente tupla: *(1, dirección del nodo emisor, dirección del nodo receptor, mensaje)*

PREGUNTAS:

Pregunta 1:

Suponga que el **TDA Nodo** existe y su especificación sintáctica es:

Tipo *Nodo*

CREAR_NODO: -> Nodo

DESTRUIR_NODO: Nodo ->

Suponga que además existe el **TDA Anillo**, el cual se soporta sobre el TDA Nodo y cuya especificación sintáctica es:

Tipo *Anillo [Nodo]*

CREAR_ANILLO: -> Anillo

DESTRUIR_ANILLO: Anillo ->

HACER_NODO_MAESTRO: Nodo -> Nodo

INICIALIZAR_ANILLO: Anillo x Nodo -> Anillo

INSERTAR_NODO: Anillo x Nodo x Dirección-> Anillo

RETIRAR_NODO_MAESTRO: Anillo -> Anillo

RETIRAR_NODO: Anillo x NODO -> Anillo

DESACTIVAR_ANILLO: Anillo -> Anillo

IDENTIFICAR_PREDECESOR: Anillo x Nodo -> Dirección

IDENTIFICAR_SUCESOR: Anillo x Nodo -> Dirección

BUSCAR_TOKEN: Anillo -> Dirección

OTORGAR_TOKEN: Anillo x Nodo -> Anillo

TRANSMITIR: Anillo x mensaje x Nodo_Emisor x Nodo_Receptor -> Anillo

BUSCAR_NODO: Anillo x Nodo -> Booleano

BUSCAR_NODO_MAESTRO: Anillo -> Booleano

IDENTIFICAR_NODO: Nodo -> Dirección

RECONOCER_PRIMER_NODO_NO_MAESTRO: Anillo -> Nodo

TAMAÑO_ANILLO: Anillo -> entero

Desarrolle la especificación semántica del TDA Nodo.

(2 puntos)

Pre: Ninguna

Nodo CREAR_NODO ()

Post: Un nodo existe pero no tiene dirección de red en el anillo

Pre: Un nodo existe

*vacío DESTRUIR_NODO (Nodo *N)*

Pre: El nodo ya no existe porque se eliminaron todos los recursos que lo conformaban

Pregunta 2:

Desarrolle la especificación semántica del TDA Anillo.

(18 puntos)

Pre: Ninguna

Anillo CREAR_ANILLO ()

Post: Existe un anillo pero sin nodo maestro ni cualquier otro.

Pre: Un anillo existe pero sin nodo maestro ni cualquier otro.

*vacío DESTRUIR_ANILLO (Anillo *A)*

Post: El anillo ya no existe porque se eliminaron todos los recursos que lo conformaban

Pre: Un nodo existe sin dirección alguna.

*Nodo HACER_NODO_MAESTRO (Nodo *N)*

Post: El nodo ahora contiene una dirección cero

Pre: Un nodo existe con la dirección cero y un anillo existe con tamaño igual a cero.

*Anillo INICIALIZAR_ANILLO (Anillo *A, Nodo *N)*

Post: El anillo ahora contiene un único nodo y con la dirección cero

Pre: Un nodo existe con el nodo maestro.

*Anillo INSERTAR_NODO (Anillo *A, Nodo *N, Entero D)*

Post: El anillo ahora contiene un nodo más.

Pre: Un anillo existe con un único nodo que contiene la dirección cero.

*Anillo RETIRAR_NODO_MAESTRO (Anillo *A)*

Post: Un anillo existe pero sin nodo alguno.

Pre: Un anillo existe con al menos dos nodos, uno del tipo maestro.

*Anillo RETIRAR_NODO (Anillo *A, NODO *N)*

Post: Un anillo existe pero con un nodo menos.

Pre: Un anillo existe con al menos un nodo, pero no contiene el nodo maestro.

*Anillo DESACTIVAR_ANILLO (Anillo *A)*

Post: El anillo existe pero no contiene nodo alguno.

Pre: Un anillo existe con al menos un nodo.

Entero IDENTIFICAR_PREDECESOR (Anillo A, Nodo N)

Post: En caso de que el anillo contenga únicamente el nodo maestro se devuelve cero. Si no es así, se devuelve un valor entero distinto de cero.

Pre: Un anillo existe con al menos un nodo.

Entero IDENTIFICAR_SUCESOR (Anillo A, Nodo N)

Post: En caso de que el anillo contenga únicamente el nodo maestro se devuelve cero. Si no es así, se devuelve un valor entero distinto de cero.

Pre: Un anillo existe con al menos un nodo.

Entero BUSCAR_TOKEN (Anillo A)

Post: En caso de que el anillo contenga únicamente el nodo maestro se devuelve cero. Si no es así, o se devuelve un valor positivo o un valor negativo que indica que nadie tiene el token.

Pre: Un anillo existe con al menos dos nodos.

*Anillo OTORGAR_TOKEN (Anillo A, Nodo *N)*

Post: El token ha pasado de un nodo a su sucesor.

Pre: Un anillo existe con al menos dos nodos.

*Anillo TRANSMITIR (Anillo A, Mensaje M, Nodo NE, Nodo *NR)*

Post: El nodo receptor ha recibido un mensaje y el mensaje ha dado la vuelta al anillo

Pre: Un anillo existe con al menos un nodo.

Booleano BUSCAR_NODO (Anillo A, Nodo N)

Post: En caso de que el anillo contenga al nodo indicado devuelve “verdad”, en caso contrario “falso”.

Pre: Un anillo existe.

Booleano BUSCAR_NODO_MAESTRO (Anillo A)

Post: En caso de que el anillo contenga un nodo con dirección cero devuelve “verdad”, en caso contrario “falso”.

Pre: Un anillo existe con al menos un nodo.

Entero IDENTIFICAR_NODO (Nodo N)

Post: Devuelve la dirección del nodo suministrado.

Pre: Un anillo existe con al menos dos nodos.

Nodo RECONOCER_PRIMER_NODO_NO_MAESTRO (Anillo A)

Post: Devuelve el primer nodo del anillo que no es el maestro.

Pre: Un anillo existe.

Entero TAMAÑO_ANILLO (Anillo A)

Post: Devuelve la cantidad de nodos que contiene el anillo.

Pregunta 3:

Con base al lenguaje de programación C, describa una estructura de datos para representar el TDA
Nodo **(1 punto)**

```
typedef struct nodo {
    char* equipo;
    int dirección; /* -1 indica que no tiene, 0 que es nodo maestro y >0 que es un nodo */
    int predecesor; /* -1 indica que no tiene */
    int sucesor; /* -1 indica que no tiene */
    char* mensaje_recibido;
    char* mensaje_a_transmitir;
} NODO;
```

Pregunta 4:

Con base al lenguaje de programación C, describa una estructura de datos para representar el TDA Anillo. **(1 punto)**

```
typedef struct anillo {
    char* nombre del anillo;
    int tamaño;
    int quantum_de_tiempo_del_nodo;
    int reloj_anomalia_nodo_maestro;
    NODO* nodo_maestro; /* NULL indica que no posee nodo maestro */
    LISTA direcciones_disponibles; /* Se asume que existe el TDA lista simple enlazada */
    LISTA direcciones_usadas;
    LISTA_CIRCULAR nodos; /* Se asume que existe el TDA lista simplemente circular*/
} ANILLO;
```

Pregunta 5:

Haga el algoritmo de la operación INICIALIZAR_ANILLO. **(3 puntos)**

COMIENZO

```
A <-- CREAR_ANILLO ()
N <-- CREAR_NODO ()
N <-- HACER_NODO_MAESTRO (N)
A <-- INICIALIZAR_ANILLO (*A, *N)
```

FIN

INICIALIZAR_ANILLO (Anillo *A, Nodo *N)

COMIENZO

```
A <-- INSERTAR_NODO (A, N, 0)
RETORNAR (A)
```

FIN

Pregunta 6:

Haga un código en lenguaje C para la operación TRANSMITIR. Considere además que dispone de las siguientes dos funciones:

/ Genera un mensaje según el formato de la red del tipo anillo y lo deja listo para ser enviado */*
char colocar_mensaje (tipo_de_mensaje, dirección del emisor, puntero_a_la_dirección del receptor, char *datos);*

/ Envía por el puerto de hardware como salida un mensaje que está en el formato de la red. En caso de que la transmisión falle la función retorna el valor -1. De otro modo, devuelve 0 */*

*int poner_en_la_red (Anillo *A, char *cadena_ya_en_formato_de_la_red_anillo)*

(5 puntos)

```
void transmitir (ANILLO *A, char *mensaje, NODO ne, NODO *nr)
```

```
{
```

```
    char *buffer;
```

```
    if (buscar_token (A) == identificar_nodo (ne))
```

```
    {
```

```
        /* Se tiene el token por lo que se puede transmitir */
```

```
        ne->mensaje_a_transmitir=colocar_mensaje (1, identificar_nodo(ne), identificar_nodo (nr),
```

```
mensaje);
```

```
        buffer = ne -> mensaje_a_transmitir;
```

```

        if ((poner_en_la_red (A, buffer)) == -1)
            fprintf (stderr, "\nerror al enviar datos");
    }
}

```

Pregunta 7:

Haga un algoritmo que describa el proceso para desactivar la red por una condición de anomalía al haberse perdido el nodo maestro. **(5 puntos)**

Anillo **DESACTIVAR_ANILLO** (Anillo *A)

COMIENZO

MIENTRAS (NO(BUSCAR_NODO_MAESTRO(A)))

 ESPERAR_MINUTOS (1)

SI (NO(BUSCAR_NODO_MAESTRO(A))) **ENTONCES**

 T <-- TAMAÑO_ANILLO(A) - 1

MIENTRAS (T > 0)

 N <-- RECONOCER_PRIMER_NODO_NO_MAESTRO (A)

 RETIRAR_NODO (N)

 T <-- T -1

FMIENTRAS

FSI

 A <-- RETIRAR_NODO_MAESTRO (A)

FMIENTRAS

RETORNAR (A)

FIN

MT/mt
20-03-2013