



Tarea 4

Introducción a C - Estructuras

1. Objetivo

Al finalizar este trabajo el estudiante podrá conocer y aplicar arreglos de estructuras, estructuras autoreferenciadas y búsqueda en tablas.

2. Problema 1

2.1. Ejercicio 1 (30 %)

Registrador de dato en base a una estructura.

Para comprender la función de una estructura en un programa de lenguaje c, desarrolle un código que permita de manera simple, ingresar los datos técnicos de un sensor en una estructura y luego los imprima a través de una función. Los datos a ingresar desde la función main, son los siguientes:

- ID Sensor: 1
- Sensor: Termopar
- Tipo: K
- Max temperatura: 400°C
- diámetro: 5mm

Finalmente, desarrolle una función que permita visualizar las variables de la estructura y los respectivos datos almacenados en cada una de ellas.

2.2. Ejercicio 2 (30 %)

Para las mediciones se integran dos modelos de sensores nuevos. En este caso, considere el mismo código del ejercicio 1 y busque la forma de almacenar los tres sensores, utilizando estructuras y arreglos.

- Utilice el mismo programa del ejercicio 1, en donde la estructura debe contener las variables y la función creada debe presentar la información de los 3 sensores.
- A dicho programa, se debe integrar una array de estructura, que permita ingresar las especificaciones de los 3 sensores.
- Por consiguiente, tenga en consideración que en el programa se aplica una estructura como parámetros de función y un array de estructuras. NOTA: Utilice un bucle para ingresar los datos de sensores.

Las características de los sensores extras son las siguientes:

Sensor 2:

- ID Sensor: 2
- Sensor: Termopar
- Tipo: J
- Max temperatura: 760°C
- diámetro: 8mm

Sensor 3:

- ID Sensor: 3
- Sensor: Termistor
- Tipo: NTC
- Max temperatura: 400°C
- diámetro: 2mm

El objetivo de este programa, es presentar los datos de los sensores ingresados por teclado (ID Sensor 1, 2 y 3), utilizando estructuras y arreglos para ordenar la información. Cabe destacar, que las estructuras permiten buscar un conjunto de variables bajo el mismo nombre.

3. Problema 2 (40 %)

Como introducción a la búsqueda de tablas y el uso del comando hash, desarrolle un programa que permita escribir, eliminar, buscar y presentar datos en una tabla definida. Asimismo, lleve a cabo las siguientes instrucciones para desarrollar el programa:

- Defina una tabla de 10 posiciones (`#define tam 10`)
- El programa deberá ejecutar las acciones en el siguiente orden: función inicial, función de escritura y función de impresión, función eliminar y función impresión; y finalmente función de búsqueda.
- Función inicial: Al iniciar el programa, escriba un valor -1 en cada una de las posiciones de la tabla a través de una función destinada para ello.
- Función de escritura: Debe crear una función hash que recibe la clave y devuelve un índice para acceder a una posición de la tabla y escribir un valor sobre ella. Asimismo, para simplificar el ejercicio utilizaremos el valor de la función (ej: `escriurahash(6)`), para indicar el valor 6 como clave de la función y escribir el mismo valor en la posición de la tabla. En el caso de no haber colisión, se debe presentar un mensaje de que se escribió el valor en la tabla. Por consiguiente, en el caso de que se intente escribir en la tabla dos valores iguales, esta debe indicar que existe colisión. Para este efecto, no es necesario solucionar la colisión.
- Función impresión: Esta función solo imprime los valores de la tabla de manera ordenada.
- Función eliminar: La función eliminar al igual que el de escribir, consultará si el valor de la clave ingresada es el mismo que el del valor de la tabla. En el caso de que sea el mismo, este escribirá un valor -1; en el caso contrario, debe imprimir que el valor no esta presente en la tabla hash. Al finalizar esta función se debe imprimir el estado actual de la tabla.
- función búsqueda: La función búsqueda (ej: `busqueda(6)`), posicionará el valor 6 y lo igualará a la clave, en el caso de que ya exista un valor 6 en la posición `arr[6]`, este debe presentar un mensaje con búsqueda encontrada; en el caso contrario, búsqueda no encontrada.

Finalmente, el estudiante aprenderá de manera básica a escribir, eliminar, buscar e imprimir un valor en una tabla hash. Para finalizar, en el `main()` del programa, se debe cumplir la siguiente secuencia:

- Al inicio, la tabla debe partir con un valor -1 en todos los bloques.
- Luego debe iniciar la función de escritura, asignando al menos 4 valores distintos a la tabla de 10 posiciones. Al menos uno de los valores, debe presentar una colisión en la tabla.

- Imprima los valores de la tabla, los valores no escritos serán -1 y los escritos deberán indicar su respectivo valor.
- Elimine al menos dos elementos de la tabla, utilizando la respectiva función para ello.
- Genere al menos dos búsquedas de valores en la tabla, utilizando la respectiva función.

```

int main ()
{
    init();
    escriturahash (2);
    escriturahash (3);
    escriturahash (6);
    escriturahash (6);

    printf ("Tabla hash \n");
    impresion(); //funcion muestra
    printf ("\n");

    printf("Eliminando el valor 10 .. \n");
    eliminar (10); // funcion para borrar
    printf ("Despues de la tabla hash de eliminacion \n");
    impresion(); //funcion muestra
    printf ("\n");

    printf("Buscando valor 2.. \n");
    busqueda (2); //funcion para buscar

    return 0;
}

```

```

2 insertado em arr[2]
3 insertado em arr[3]
6 insertado em arr[6]
Colision: arr[6] ya tiene el elemento6!
No se puede insertar 6
Tabla hash
arr[0]=-1
arr[1]=-1
arr[2]= 2
arr[3]= 3
arr[4]=-1
arr[5]=-1
arr[6]= 6
arr[7]=-1
arr[8]=-1
arr[9]=-1

Eliminando el valor 10 ..
10 no esta presente en la tabla hash
Despues de la tabla hash de eliminacion
arr[0]=-1
arr[1]=-1
arr[2]= 2
arr[3]= 3
arr[4]=-1
arr[5]=-1
arr[6]= 6
arr[7]=-1
arr[8]=-1
arr[9]=-1

Buscando valor 2..
Busqueda encontrada

```

Figura 1: Ejemplo de la respuesta del programa (tablas hash)