

Programación en lenguaje C en QT

Alvaro Martinez – Diego Quiroga

ABSTRACT: El presente informe se refiere a la práctica de laboratorio de la clase de Sistemas embebidos para reforzar conocimientos en lenguaje C.

I. INTRODUCCION

La práctica de laboratorio se realiza en base al estudio del software de programación C con la intención de que los estudiantes del curso sistemas embebidos comprendan el uso de apuntadores, estructuras, entre otros, para el desarrollo de los ejercicios planteados en la clase.

Por medio de tales ejercicios cada estudiante en grupos de 2 personas es introducido a los comandos básicos con los que luego deberá realizar las practicas propuestas por el profesor. El lenguaje de programación c es de uso generalizado y conduce a promover y desarrollar software de sistemas orientados a objetos y aplicaciones.

OBJETIVOS

Objetivo general:

- Comprender el uso del lenguaje de programación C.

Objetivos específicos:

- Reforzar la teoría vista en clase.
- Identificar como se puede utilizar el lenguaje C en arreglos, matrices, funciones, etc.
- Resolver los problemas planteados en el laboratorio propuesto por el profesor.

II. DESARROLLO

1. Diccionario:

Se hizo un programa el cual empezaba llamando el archivo de valores separados por comas (CSV), para esto se creó un apuntador de tipo FILE y se hizo el llamado de la función fopen para que abriera el archivo en modo de lectura en el programa. Luego se hizo uso de la función fgets para leer el contenido del archivo CSV la cual nos pedía la variable en donde se iba a guardar cada dato, el tamaño de dicha variable y el puntero el cual apunta al archivo CSV; ya que el diccionario podía contener hasta 100 palabras y significados se hizo uso de la función strstr para comparar cada que dato contenga su palabra y su significado, y se hizo un condicional para que solo leyera el número de palabras que se encuentran en el archivo y de esta manera no arroje basura en los demás espacios a la hora de imprimir el resultado.

Posteriormente se utilizó la función strtok para que separara lo que había antes (las palabras) y después (los significados) del token (en este caso la coma). Las palabras se iban almacenando en un arreglo llamado add y los significados en un arreglo llamado des.

Luego se procedió a hacer el menú en donde se tenían tres opciones:

1. Agregar Palabra
2. Buscar
3. Listar

El usuario puede ingresar cualquiera de los tres números sea el caso que desea este.

Si el usuario ingresaba uno desde el menú de inicio, el programa inmediatamente le solicitaba la palabra que quería agregar; esta palabra ingresada se comparaba con todas las palabras ingresadas anteriormente y con las palabras del archivo CSV mediante la función strcmp, si se daba el caso que la palabra ya estaba en el diccionario, se iba a mostrar en pantalla que la palabra ya existe y se regresaba al menú de inicio mediante la tecla enter; si por el contrario la palabra aún no estaba ingresada en el diccionario, el programa le iba a solicitar al usuario que digitara la descripción de la palabra e iba a mostrar un mensaje de que la palabra fue agregada correctamente, al finalizar este proceso se retornaba al menú de inicio mediante enter.

Si el usuario ingresaba dos desde el menú de inicio, se le solicitaba al usuario que digite la palabra o parte de la palabra para que el programa busque todas las coincidencias y las muestre en una lista, seguido de esto se hace uso de la función strstr para comparar cada palabra del diccionario con la palabra o parte de la palabra que el usuario ingreso y estas se guardaban en nuevas variables junto con sus respectivos significados para que las palabras no quedaran con significados diferentes. Si el programa no encontraba ninguna coincidencia, este arrojaba un mensaje diciendo que no se encontró ninguna coincidencia, si por el contrario encontró coincidencias, el programa listaba todas las coincidencias y las enumeraba para que el usuario escoja el numero de la palabra a la cual quiere ver su significado. Al finalizar, se retornaba al menú mediante la tecla enter.

Si el usuario ingresaba tres desde el menú de inicio, el programa simplemente tomaba todas las palabras y las mostraba listadas en orden alfabético, para esto se hizo uso de la función strcpy para guardar las palabras a ordenar en nuevas variables y de esta manera evitar los conflictos entre las palabras y sus significados, después, mediante la función strcmp se comparaba la palabra presente de acuerdo al iterador con todas las demás para determinar si era mayor o menor, después se procedía a realizar el método de la burbuja para ordenarlas y por último se imprimían. Se retornaba al menú de inicio con enter.

Si el usuario ingresaba otro número desde el menú de inicio, el programa iba a mostrar un mensaje diciendo comando no permitido intente de nuevo y lo retornaba al menú de inicio. Por último, se creó una función llamada enter que permitía volver al menú de inicio si se oprimía la tecla enter cuando se

mostrara el mensaje solicitando presionar dicha tecla para volver al menú de inicio.

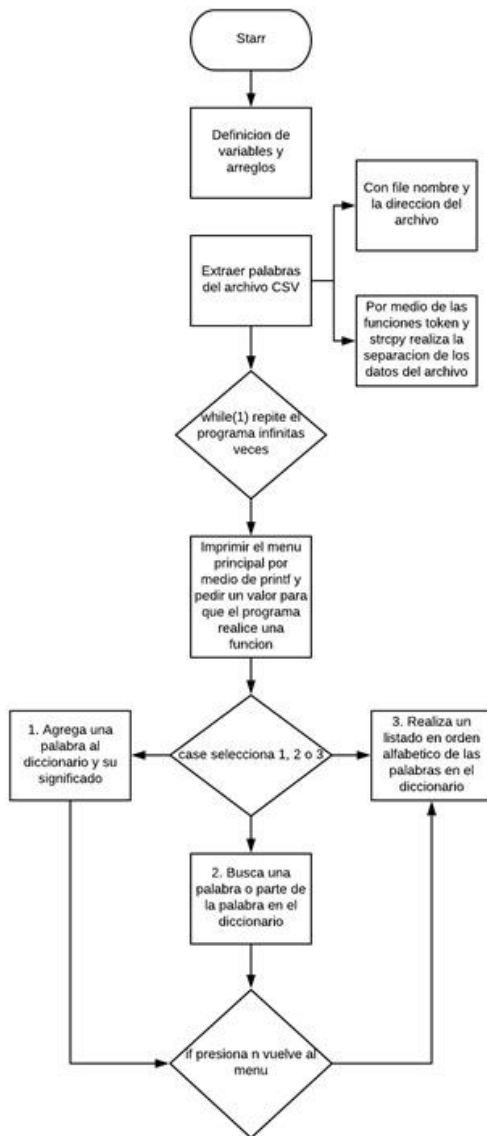


Ilustración 1 (Diagrama de flujo)



Ilustración 2 (Opciones en pantalla)

2. Cine

Se diseño y se puso en marcha un programa que maneje la venta y asignación de salas de cine y puestos

- El programa tiene un menú para ingresar películas en cartelera, listarlas y ubicarlas en 5 salas de cine. Cuando una persona quiere comprar una o varias boletas, el sistema debe preguntar ¿cuál película quiere ver?, según la sala debe mostrar en pantalla la gráfica del teatro junto con sus sillas (120 sillas ordenadas en 10 filas de 12 sillas con 2 pasillos cada 6 sillas)

- Muestra cuáles sillas están ocupadas y cuales libres, permitir escoger las sillas correspondientes, cambiar su estado a ocupadas y generar el ticket en pantalla con números de silla y precio. (cada boleto vale \$10.000, y se suman 2000 por cada fila que avance hacia la parte de atrás del teatro).

- El sistema guarda en un archivo el estado actual de la venta de las boletas, para que así se cierre el programa se mantengan todas las reservas.

- El sistema tiene una opción para realizar la contabilidad de las ventas, mostrando un listado con todas las ventas y la suma total de las mismas

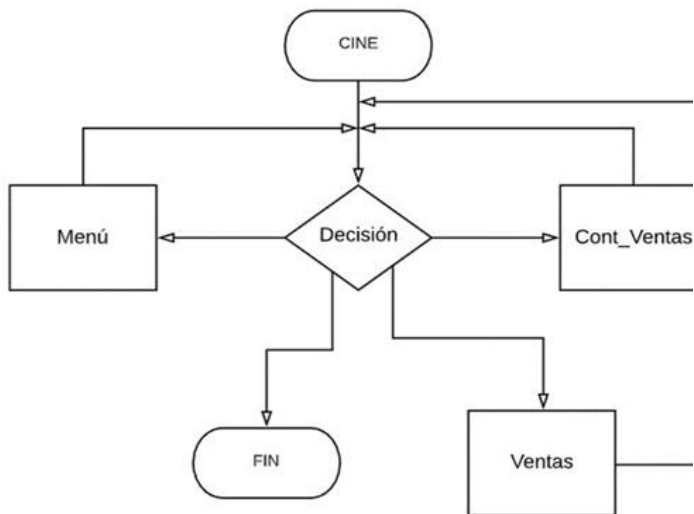


Ilustración 3 (Diagrama de flujo)

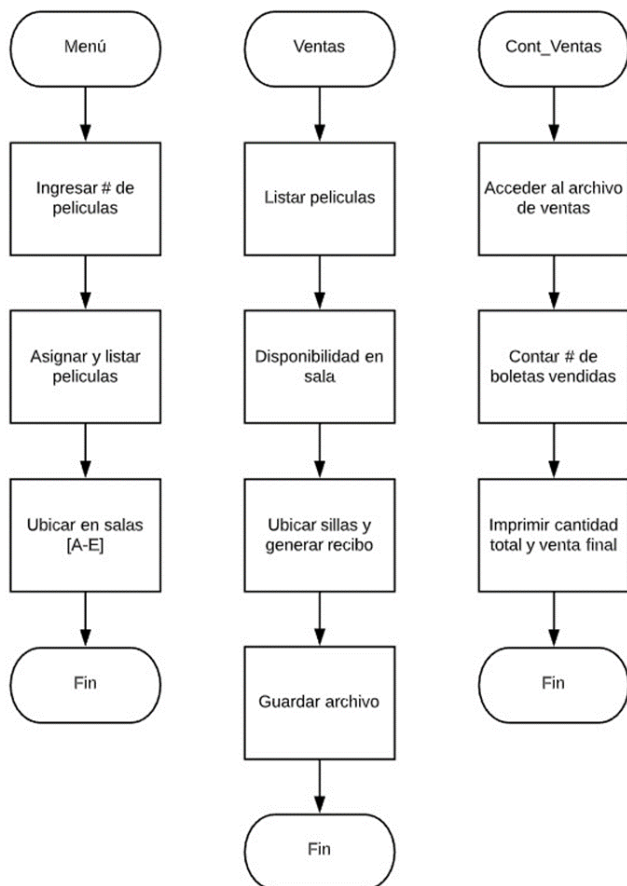


Ilustración 4 (Diagrama de flujo)

de código y optimiza el uso de memoria y el tiempo de ejecución del código.

- Los punteros nos permiten acceder a la dirección de la variable lo cual nos optimiza la lógica de programación.
- Se utiliza la función `strcmp` que realiza la comparación de variables de tipo string ya que compara una cadena con otra para saber si son iguales.
- Para separar los datos que se encuentran en el archivo CSV se utilizó la función `strtok` la cual pide una coma para detectar la separación entre palabra y definición. Almacenando las cadenas de caracteres en diferentes arreglos.

III. CONCLUSIONES

- El uso de funciones en un programa de lenguaje c facilita al programador la comprensión y realización