

Proyecto 1 - 2018-2

Este proyecto vale 15% de la nota del curso.

Debe ser elaborado en grupo (3 integrantes).

No se permite ningún tipo de consulta entre grupos.

Se debe entregar por Sicua+ a más tardar el 21 de octubre de 2018

A. OBJETIVOS

- Practicar el lenguaje C y desarrollar un programa de complejidad pequeña.
- Conocer las operaciones de C para el manejo de bits.
- Aplicar lo anterior en un traductor de binario a octal.

B. DESCRIPCIÓN DEL PROBLEMA

El objetivo de este proyecto es desarrollar un traductor de binario a octal. Este traductor permitirá leer un archivo cualquiera y guardar su traducción en otro archivo de texto. La forma en que se hará la traducción será explicada más adelante en la sección de codificación.

Codificación

El sistema de numeración octal es un sistema de numeración en base 8. Como es una base que es potencia exacta de 2 ($8=2^3$), nos permite representar todos sus posibles valores en grupos de 3 bits. Se espera de su programa que, a partir del archivo inicial, agrupe los bits en dígitos octales para después reportar su valor en un nuevo archivo. A continuación se muestra un ejemplo. Se tendrá la siguiente cadena de bits representando el mensaje (se separan los bytes con "|"):

Archivo de entrada: **Aa**

01000001 | 01100001 |

Agrupación en octales (se separa cada octal con "-")

010-000-010-110-000-1**00**

Note que, como se explica más adelante, fue necesario completar el último dígito octal agregando dos ceros (**00**)

Esto corresponde con el octal: 202604

El cual se reporta en el archivo de salida representando cada dígito con su ASCII respectivo, es decir: '2', '0', '2', '6', '0', '4'.

Note que el archivo de entrada tiene dos bytes, pero el de salida tiene 6.

Formato del archivo

En este ejercicio se podrá utilizar cualquier archivo de entrada para traducirlo. Sin embargo, el

archivo de salida, solamente contendrá caracteres del '0' al '7' que representan los dígitos octales que codifican los valores numéricos de los bits una vez agrupados en los octales.

Restricciones y consideraciones adicionales

El programa debe tener en cuenta y controlar que si el octal al final del archivo queda incompleto, se deberá completar los dígitos faltantes con 0 en los bits.

El programa

En el archivo adjunto ("main.c"), encuentra el esquema del programa. El programa se invoca por línea de comando.

El programa ya tiene definidos los procedimientos `main ()`, `readFile (Datos *, char *)` y `writeFile (int, Datos * archivoEnOctal, char *)`. El procedimiento `main` se encarga de preguntar qué archivo se desea traducir, cargar sus datos en memoria principal y preguntar el nombre del archivo para guardar la respuesta. Por otro lado, la función `readFile` se encarga de abrir un archivo y leerlo en el vector de datos y `writeFile` se encarga de escribir un archivo a partir del vector de datos. Estas funciones ya se encuentran desarrolladas, por lo cual es muy importante que estas **no sean modificadas**.

- Para traducir el archivo, el procedimiento `main` recibe por parámetro la ruta completa del archivo de origen que se da como argumento de comando y el archivo de destino.
- El procedimiento de lectura recibe el nombre del archivo y un vector de tipo `unsigned char`; el procedimiento lee el archivo y lo carga en el vector. Retorna cuántos bytes leyó.
- El procedimiento de escritura recibe el nombre del archivo donde se va a escribir la traducción y el vector de tipo `unsigned char`, y escribe el vector de datos en el archivo.

El procedimiento que el grupo debe completar es el siguiente:

- **`convertirAOctal(Datos * datosBin, Datos * datosOct)`**: Esta función se encarga de convertir a octal y escribir las representaciones numéricas de cada uno de los bytes que se encuentran en el archivo que se desea traducir. Si queda algún octal incompleto, la función debe completar los bits faltantes con ceros (0). El parámetro **`datosBin`** es un apuntador a una estructura donde se encuentran los datos de entrada y el parámetro **`datosOct`** es un apuntador a una estructura donde se encuentran los datos de salida.

Para desarrollar el programa pueden crear las funciones adicionales que necesiten (recomendado), las cuales deben estar debidamente comentadas y documentadas.

C. ESPECIFICACIONES

- Los programas se deben escribir en C (usando el compilador de Visual C++). Nota importante: los programas se calificarán únicamente usando el compilador de Visual

Studio 2017; si el programa no compila en este ambiente, se considerará que no corre (así compile en otros ambientes).

- Legibilidad del programa: indentar el programa; escribir comentarios explicando el código.
- Debe respetar la estructura del código entregado. En particular, debe usar los procedimientos y variables del esqueleto.

D. CONDICIONES DE ENTREGA

- Entregar el código fuente junto con el ejecutable en un archivo *.zip. **Al comienzo del archivo fuente escriba los nombres de los miembros, sus códigos y correos, de lo contrario no será evaluado (ver esqueleto).** Si su programa no funciona o si su solución tiene particularidades, puede enviar un archivo .docx explicando por qué cree que no funciona o qué fue lo que hizo.
- El trabajo se realiza en grupos de **3** personas. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).
- Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación puede afectar la nota de todos los miembros.
- El proyecto debe ser entregado por Sicua+ por uno solo de los integrantes.
- **Se debe entregar por Sicua+ a más tardar el 21 de octubre de 2018 hasta las 11:50 pm.**

E. CASOS DE PRUEBA

Adjuntos al proyecto encontrarán tres archivos de ejemplo con los valores de entrada y sus respectivos archivos esperados de salida. Pueden utilizar los archivos de entrada para verificar que sus programas estén funcionando correctamente, pero es altamente recomendado que generen sus propias pruebas.

F. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Para las funciones propuestas se harán 5 pruebas: los 3 casos de prueba entregados y otros 2 nuevos. De cada caso, se escribirá s y luego se leerá lo que se escribió. Para cada caso, se revisará si la salida es correcta o no según los requerimientos establecidos en el enunciado. Cada prueba vale 10%.
- Inspección del código (50%). Se consideran tres aspectos:
 - 10% - legibilidad (nombres dicientes para variables, comentarios e indentación)
 - 20% - manejo de bits (uso de los operadores de bits de C: >>, &, etc.)

- o 20% - manejo de la estructura de datos (recorrido, manejo de los elementos).

G. RECOMENDACIONES

- Recuerden que los trabajos hechos en grupo y entregados individualmente (o en grupos diferentes al original) son una forma de fraude académico.