

Este proyecto vale 15% de la nota del curso.
Debe ser elaborado en grupo (mínimo 2, máximo 3 integrantes).
No se permite ningún tipo de consulta entre grupos.
Se debe entregar por Sicua a más tardar el 20 de noviembre a las 23:50

A. OBJETIVOS

- Practicar con el lenguaje ensamblador.
- Practicar con programación mixta C-ensamblador.
- Aplicar lo anterior en un programa que permite la codificación de un archivo binario a uno de texto

El proyecto consiste en completar un programa en C mediante ensamblador embebido siguiendo las instrucciones que se dan a continuación.

B. DESCRIPCIÓN DEL PROBLEMA

Utilizando como base el proyecto anterior se reescribirá el programa usando el lenguaje ensamblador. Sin embargo, existe una diferencia respecto a la entrada que recibe el proyecto: será necesario indicar si se está haciendo un cambio de archivo binario a archivo ASCII o viceversa, para lo cual usaremos un tercer parámetro en el método main. Para indicar que se convertirá a ASCII, se utilizará el argumento '-a', y para el caso contrario se utilizará el argumento '-b'.

PROCEDIMIENTOS

Complete los procedimientos según lo indicado en el esqueleto del programa adjunto. En particular, note que hay tres tipos de restricciones sobre los procedimientos:

- Los que no deben ser modificados (se deben dejar tal como están).
- Los que deben ser escritos en ensamblador pero pueden utilizar nombres simbólicos de variables. Esto quiere decir que pueden declarar variables locales en C y usarlas en el código ensamblador, igual que los parámetros de las funciones.
- Los que deben ser escritos sin usar nombres simbólicos, es decir, que acceden a los parámetros y variables locales a través de desplazamientos en la pila.

Procedimientos que no deben ser modificados:

- `main`, `cargarArchivo` y `guardarArchivo`.

Procedimientos para escribir en ensamblador usando nombres simbólicos:

- `int convertir(char * p):`

Este procedimiento recibe el nuevo parámetro y retorna

- 1 si hay que convertir a ASCII
- 2 si hay que convertir a binario
- 0 si no se reconoce el parámetro

El parámetro no se reconoce si no empieza por '-', si no es ni 'a' ni 'b' o si tiene más de dos caracteres; es decir, si no es exactamente "-a" ni "-b".

- `void conversionTexto(ARCHIVO *arch, ARCHIVO *resultado):`

Convierte la información binaria del archivo `arch` al archivo `resultado` en texto (ASCII).

- `unsigned char sacar5bits(ARCHIVO *arch, int n):`

Extrae el n-ésimo grupo de 5 bits del archivo de entrada y retorna un `char` que contiene en sus 5 bits menos significativos.

- `void codificar(unsigned char cinco):`

Recibe un byte con información en los 5 bits menos significativos, y realiza la codificación correspondiente en ASCII.

- `void conversionBinario(ARCHIVO *data, ARCHIVO *resultado):`

Convierte la información en ASCII de la estructura `ARCHIVO` apuntada por '`data`' (archivo de texto) * y la guarda en la estructura `ARCHIVO` apuntada por '`resultado`' como una serie de bytes en la que a cada caracter se le extraen sus últimos 5 bits y se guardan en la salida unos a continuación de los otros

Procedimientos para escribir en ensamblador sin usar nombres simbólicos:

- `void meter5bits(ARCHIVO *arch, int n, char bits):`

Introduce los 5 bits menos significativos del byte '`bits`' en el n-ésimo grupo de 5 bits de la estructura `ARCHIVO` apuntada por '`arch`'.

Nota: Puesto que el proyecto se basa en su proyecto anterior, para que funcione debe funcionar también la parte anterior; en caso de que no sea así, debe arreglar el proyecto anterior para esta entrega.

C. ESPECIFICACIONES

- Los programas se deben escribir en C (en Visual Studio). Nota importante: los programas se calificarán únicamente usando el ambiente de visual; si el programa

no compila en este ambiente, se considerará que no corre (así compile en otros ambientes).

- Legibilidad del programa: sangrar el programa; escribir comentarios explicando el código; nombres dicientes de variables.
- Debe respetar la estructura y requerimientos del código entregado. En particular, debe usar los procedimientos y variables del esqueleto, y no pueden crear procedimientos adicionales.

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 .doc explicando por qué cree que no funciona o qué fue lo que hizo.
- El trabajo se realiza en grupos de máximo 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 del grupo.
- **Se debe entregar por Sicua a más tardar el 20 de noviembre a las 23:50.**

E. CASOS DE PRUEBA

Para hacer pruebas se adjuntan 3 casos de prueba: dos archivo de texto (caso1.txt y caso2.txt) y un archivo de mapa de bits (casoImagen.bmp). Además, se adjuntan los archivos de texto resultantes de la codificación de cada uno de los dos casos de prueba con el objetivo de poder verificar el funcionamiento de la codificación o decodificación por separado.

F. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Para las funciones propuestas se harán cuatro pruebas: dos de los casos de prueba entregados (seleccionados al azar) y otros dos nuevos.
- Inspección del código (50%). Se consideran tres aspectos:
 - ✓ 10% - legibilidad (nombres dicientes para variables, comentarios e indentación)
 - ✓ 20% - direccionamiento en ensamblador
 - ✓ 20% - uso del lenguaje ensamblador (evaluación de expresiones y control).

G. RECOMENDACIONES

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