DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

ISIS1304

FUNDAMENTOS DE INFRAESTRUCTURA TECNOLÓGICA

Proyecto 2 - 2015-2

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 15 de noviembre

A. OBJETIVOS

- Practicar con el lenguaje ensamblador.
- Practicar con programación mixta C-ensamblador.
- Aplicar lo anterior en un programa que permite codificación y decodificación de información.

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.

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; también deben reservar explícitamente el espacio para las variables locales en la pila.

Procedimientos que no deben ser modificados:

- Todos los que no se mencionen a continuación que ya tenga en su proyecto anterior. Puesto que el proyecto se basa es 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.

Procedimientos para escribir en ensamblador usando nombres simbólicos:

- `void unirArchivosWAVE(int numMuestreos, unsigned short *parte1, unsigned short *parte2, unsigned short *salida, int bitsPorMuestreo)`

***ATENCIÓN**: note que se adicionó el parámetro `numMuestreos`, el cual indica el número de muestreos que se deben procesar. Esto para evitar el uso de variables globales dentro del procedimiento.*

- `void escribirMuestreo(unsigned short *pista, int bitPos, unsigned short muestreo, int bitsPorMuestreo)`

Procedimientos para escribir en ensamblador sin usar nombres simbólicos:

- `unsigned short leerMuestreo(unsigned short *pista, int bitPos, int bitsPorMuestreo)`

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: indentar 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 15 de Noviembre.

E. CASOS DE PRUEBA

Para hacer pruebas use los 3 casos de prueba del proyecto 1.

Las pruebas de funcionamiento están basadas en la comparación binaria de los archivos.

F. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Para las funciones propuestas se harán 5 pruebas: tres de los casos de prueba entregados 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.