

# Tarea Mantequilla

(C – ASM)

PROF. KIRSTEIN GÄTJENS S.

"Existen dos posibilidades: o estamos solos en el Universo o no lo estamos. Ambas son igualmente aterradoras."

Arthur C. Clarke

## Instrucciones generales:

Esta tarea se trata de realizar ligas entre C y Ensamblador. Para lo cual usaremos las versiones de TC y de TASM (el que se usa en FOC). La idea es probar que se pueden ejecutar las ligas sin problemas, por lo que no se espera que sea una tarea complicada, solo la parte de entender las ligas y trabajarlas, de ahí que se considera una tarea mantequilla.

Debe entregarse antes del miércoles 18 de octubre a la medianoche al correo [kirstein.eval@gmail.com](mailto:kirstein.eval@gmail.com)

Es de carácter individual por lo que cualquier intento de fraude se llevará a las últimas consecuencias.

Se debe desarrollar en las herramientas TC 2.0 y TASM 3.2. Allí se probará su tarea. Si la realiza en otra herramienta y no funciona con estas la tarea tendrá una nota de cero en la parte de ejecución.

Entregue un archivo .RAR que contenga los archivos fuentes, el .C, el .H y el .ASM

El nombre de todos los archivos fuentes debe ser los últimos 8 dígitos de su número de carne con la extensión correspondiente. El nombre del .RAR y del .PDF debe ser Tarea-Mantequilla-Apellido-Nombre.XXX donde Nombre y Apellido son los oficiales del curso. El subject del correo es lo usual: TAREA: Mantequilla

El cuerpo del correo debe contener, como es usual, su nombre completo, número de carné, número de grupo y curso.

Recuerde que la claridad y el orden a la hora de escribir el código y la documentación le ayudarán al asistente a calificarle mejor (y con mayor piedad :).

El valor de la tarea es de 4 resúmenes, numerados del 91 al 94.

Algo muy importante es que al ser una tarea tan sencilla no se permitirá que usen ninguna biblioteca estándar de C.

Las dos rutinas que se solicitan deben ser programadas completamente por ustedes.

## Ligas C-ASM

Para probar que se pueden realizar ligas sin problemas se debe realizar un pequeño programa que tenga una declaración de un printf propio en C y cuyo cuerpo se implemente en ASM. Así mismo se debe declarar una variable equivalente a un int en el código de ASM y realizar en C una rutina que la despliegue al revés. El programa en C recibe por la línea de comandos un número y debe convertirlo en un entero que se debe almacenar en la variable declarada en el módulo de ASM. Luego debe acceder de nuevo esa variable entera e imprimirla al revés. La rutina que imprime al revés debe ser hecha en C, pero para imprimir debe invocar el printf propio (que va a estar programado en ASM).

Si no se digitan comandos se debe desplegar una ayuda (por supuesto usando su propio printf).

En la ejecución se debe desplegar un mini acerca de de un par de renglones, dejar una línea en blanco y desplegar al revés el número que se recibió por la línea de comandos.

Se espera que los despliegues los hagan con su propio printf. Tanto el miniacercade como el despliegue del número. La rutina del printf propio debe ser capaz de desplegar enteros y strings (int y char \*) únicamente y debe trabajar con el sistema de %s.

Recuerde que el programa principal debe ser el de C y que el de ASM debe ser un módulo.

**No es permitido usar ninguna biblioteca estándar de C (ni el stdout.h, ni el string.h ni ninguna otra).**

El cuerpo en ASM del printf propio debe imprimir a la salida estándar usando la 21h correspondiente.

Pueden leer el capítulo de ligas de C-ASM para ayudarse con la tarea. Es el que dejamos en el curso de Arquitectura. Es el cap 6 del manual de TASM. Se posteo con la onceava tanda de lecturas.

## Documentación

---

Por ser un proyecto muy pequeño, la documentación debe ser lo mínimo aceptable. Aún así debe ser entregada en un PDF aparte. Debe incluir los siguientes elementos usuales para tareas de este tamaño:

- + Portada de página completa.

- + Análisis de Resultados siguiendo el formato usual de mis cursos.

Se debe dividir en partes todo lo que hay que hacer y asignarle una calificación con letra que va de la A a la E.

Donde la A representa: Concluido con éxito.

Donde la B representa: Concluido con problemas pequeños. Requiere aclaraciones.

Donde la C representa: Concluido con problemas mayores. Requiere aclaraciones.

Donde la D representa: Concluido solo en diseño. Codificación quedó incompleta. Requiere aclaraciones.

Donde la E representa: Tarea no implementada.

- + Un pequeño manual de usuario para cada una de las 18 funciones que le correspondió implementar. Cómo probarlas e interpretar sus resultados.

- + Debe documentarse la forma en la que se hizo las ligas, en especial explicar el acces a la pila para manipular los parámetros en el printf propio.

- + No olviden que el código interno debe estar bien documentado. Tomando en cuenta precondiciones, poscondiciones, parámetros, etc.

## Suerte!!