Organización del Computador
Mini TP3 - ARM Introducción

Propósito y sentido de la actividad

Nuestro primer objetivo es ganar práctica en las siguientes herramientas:

- el ensamblador as, el linker gcc y el debugger **gdb**,
- el editor de texto nano
- la terminal o línea de comandos del sistema donde trabajaremos: GNU/Linux Raspbian.

Nuestro segundo objetivo es escribir, compilar y ejecutar un primer programa en ensamblador de ARM y realizar un seguimiento de la ejecución linea por linea con el debugger gdb.

Producto final de la actividad

Al finalizar este trabajo tendremos un resumen de comandos clave para programar en lenguaje ensamblador en un entorno GNU/Linux. También tendremos una lista de los principales comandos para debuggear con gdb.

Evaluación

Para acreditar y aprobar esta actividad se solicita:

- Un archivo en formato PDF conteniendo un informe del trabajo realizado.
  El nombre del archivo debe ser **MiniTP_03_Apellido_Nombre.pdf**
- No incluir ejecutables, solo el código fuente

Esta actividad es individual, obligatoria y será calificada con:

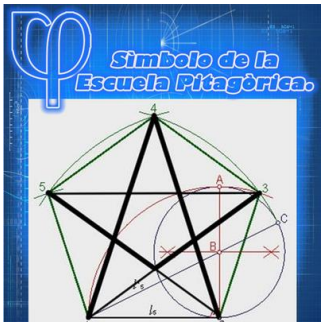
- A (aprobado)
- I (insuficiente)

En caso de no aprobar se solicitará al alumno que re-entregue los ejercicios con las correcciones pertinentes más un ejercicio adicional.

Fecha límite de entrega: 14/5/2021

Espacio de entrega: Moodle

Enunciado del Mini TP:



Para los antiguos griegos pitagóricos el 10 era un número sagrado porque era la suma de los 4 primeros números básicos 1+2+3+4 que representaban los elementos fundamentales aire, tierra, agua y fuego.

En la actualidad el número 10 conserva su importancia, es considerado el número de la perfección y en el ámbito del fútbol el 10 se reserva para el jugador más talentoso y conductor del equipo.

El siguiente programa escrito en lenguaje ensamblador ARM realiza la suma 1+2+3+4 = 10 y deja el resultado en el registro r0:

```
.data
.text
.global main
main:
        mov r0, #0      // inicializar en cero el registro resultado
        mov r1, #1      // R1 <-- 1
        mov r2, #2      // R2 <-- 2
        mov r3, #3      // R3 <-- 3
        mov r4, #4      // R4 <-- 4
        add r5, r1, r2   // R5 <-- 1 + 2 = 3
        add r6, r3, r4   // R5 <-- 3 + 4 = 7
        add r0, r5, r6   //R0 <-- 3 + 7 = 10

fin:
        mov r7, #1      // Salida al sistema
        swi 0
```

Consigna:

- Compilar y ejecutar el programa anterior por la terminal del sistema Raspbian.
- Con la ayuda del programa gdb ejecutar paso a paso el programa anterior y observar cómo se van actualizando los valores de los registros r0 hasta r6.

- Obtener una captura de pantalla donde se pueda observar el contenido inicial del registro r0
- Obtener una captura de pantalla donde se pueda observar el resultado de la suma en el registro r0