

EXAMEN PRÁCTICAS
ARQUITECTURA DE COMPUTADORES (RISC-V)
1º Ingeniería de Robótica Software, URJC

Fuenlabrada, 26 de Abril de 2019

AVISO: Asegúrate que tus programas **cumplen** con los siguientes criterios. Si no se cumple alguno de ellos la **nota máxima** de tu examen será de **2 puntos**

- **Cumplimiento de especificaciones.** Se deben cumplir las especificaciones indicadas en el enunciado: nombres de funciones, nombres de archivos, funcionalidad, etc. Compruébalo antes de entregar el examen
- **Respetar el convenio.** Resuelve las preguntas **sin violar** el convenio del uso de registros
- **Sin errores en tiempo de ejecución** (Runtime errors). Tus programas no deben generar excepciones al ejecutarse

Programar una función para el RISC-V para poner del revés una cadena de texto almacenada en memoria: `void reverse(char *pstr)`. Esta función tiene un único parámetro de entrada: la dirección de memoria donde comienza la cadena. Así, si la cadena es "Winter is Coming...", tras invocar a esta función su contenido quedará sustituido por "...gnimoC si retniW".

La función **reverse** debe llamar en algún momento a la función `void swap(char *pa, char *pb)`, que intercambia el contenido de dos posiciones de memoria, cuyas direcciones se pasan como argumentos.

El **programa principal** llama a la función `reverse` pasándole la dirección de la cadena "Winter is Coming..." (declarada en tiempo de compilación), imprime la cadena invertida en la consola y termina.

Se pide:

1. (Obligatorio) Crear el fichero **swap.asm** con la implementación de la función `swap` y un programa principal de prueba, que permita comprobar que está funcionando correctamente (3 puntos)
2. (Obligatorio) Crear el fichero **reverse.asm** con la implementación de la función `reverse`, usando un algoritmo iterativo, y el programa principal de prueba indicado en el enunciado. Recuerda que la función `reverse` debe llamar a `swap` (cópiala del fichero `swap.asm` a `reverse.asm`) (6 puntos)
3. (Opcional) Modificar el programa principal de `reverse.asm` para que llame a la función `reverse` pasándole como argumento la dirección de una cadena que introduce el usuario en tiempo de ejecución. (1 punto)