

EXAMEN DE PRÁCTICAS
Convocatoria ordinaria
ARQUITECTURA DE COMPUTADORES

1º Robótica Software, URJC

20 de Mayo de 2021

AVISO: Asegúrate que tus programas **cumplen** con los siguientes criterios. Si no se cumple alguno de ellos la **nota será de 0** en ese apartado

- **Cumplimiento de especificaciones.** Se deben cumplir las especificaciones indicadas en el enunciado: nombres de funciones, nombres de archivos, , parámetros de las funciones, funcionalidad, etc. Compruébalo antes de entregar el examen
- **Respetar el convenio.** Resuelve las preguntas **sin violar** el convenio del uso de registros (ABI del RISC-V)
- **Sin errores en tiempo de ejecución** (Runtime errors). Tus programas no deben generar excepciones al ejecutarse
- **Sin errores al ensamblar.** Los ficheros entregados NO deben dar errores al ensamblarlos. Si una función la has dejado a medio hacer, asegúrate que al menos se ensambla sin errores

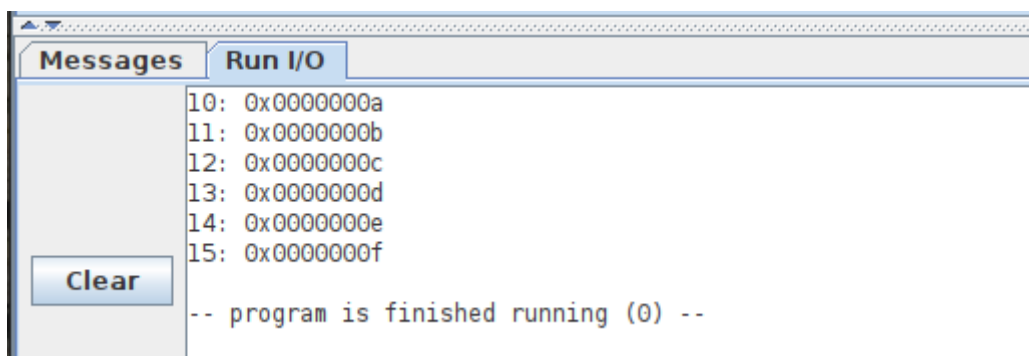
Se quiere implementar en ensamblador del RISC-V la siguiente **función**, cuya **especificación** es la siguiente:

- *void printhex(num):* Imprimir en la consola un número en hexadecimal y un salto de línea. Tiene un único parámetro de entrada: el número a imprimir en hexadecimal. No devuelve ningún valor. Para implementar esta función deberás utilizar una **llamada al sistema** que permita imprimir números en hexadecimal, y a continuación otra para imprimir el salto de línea

Además se necesitan los siguientes **programas principales**:

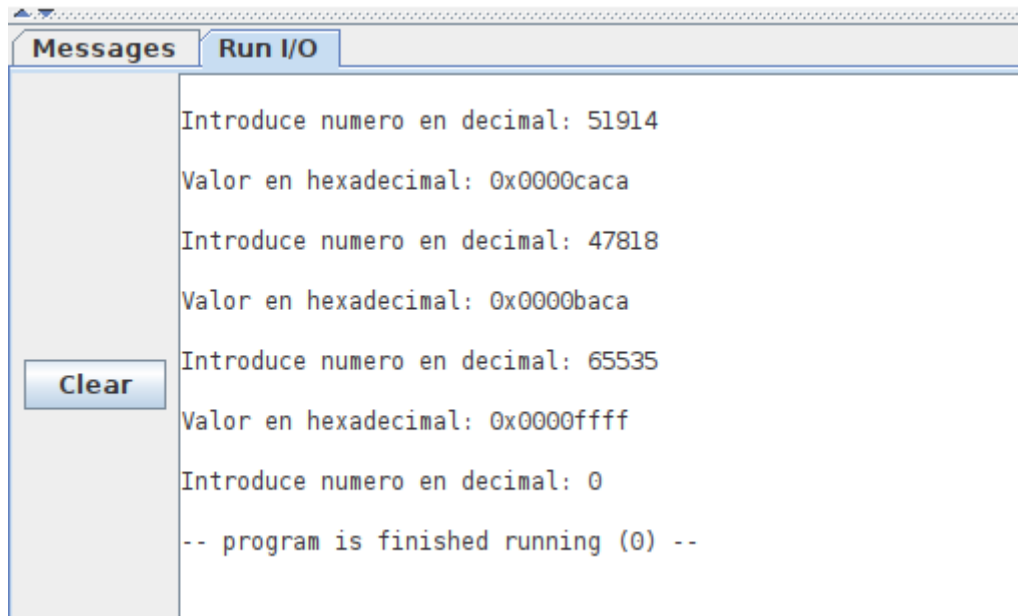
- *test_printhex.s:* Programa para comprobar la función `printhex()`. Se utiliza un bucle para imprimir en hexadecimal los números desde el 10 al 15. En la consola aparecerá en la izquierda el número en decimal, luego la cadena “: “ y después el número en hexadecimal. La impresión del número hexadecimal se debe hacer llamando a la función `printhex()`.

Tras su **ejecución**, este es el resultado que debe aparecer en la **consola**



```
Messages Run I/O
10: 0x0000000a
11: 0x0000000b
12: 0x0000000c
13: 0x0000000d
14: 0x0000000e
15: 0x0000000f
-- program is finished running (0) --
```

- *conversor.c*: Programa principal que pide al usuario que introduzca un número en decimal. Si este número es 0, el programa terminará. De lo contrario mostrará su valor en hexadecimal, llamando a la función `printhex()` y volverá a comenzar. En este pantallazo de ejemplo se muestra su funcionamiento cuando el usuario introduce los tres siguientes números: 51914, 47818, 65535 y finalmente 0 para terminar:



The screenshot shows a window with two tabs: "Messages" and "Run I/O". The "Messages" tab is active, displaying the following text:

```
Introduce numero en decimal: 51914
Valor en hexadecimal: 0x0000caca
Introduce numero en decimal: 47818
Valor en hexadecimal: 0x0000baca
Introduce numero en decimal: 65535
Valor en hexadecimal: 0x0000ffff
Introduce numero en decimal: 0
-- program is finished running (0) --
```

On the left side of the window, there is a "Clear" button.

Se pide:

1. Implementar la función `printhex()` en el fichero **printhex.s** (2 puntos)
2. Implementar el programa de prueba **test_printhex.s** (3 puntos)
3. Implementar el programa principal **conversor.s** (2.5 puntos)
4. Convertir el código del programa `test_printhex.s` en la función `test_printhex()`, que no tiene ningún parámetro de entrada ni ninguno de salida. Cada vez que se llama a esta función se imprime en la consola la conversión de los números del 10 al 15 a hexadecimal (misma salida que el programa `test_printhex.s`). Esta función debe estar en el fichero **test_printhex2.s** (2 puntos)
5. Programa principal: Llama dos veces seguidas a la función `test_printhex()` y luego termina. Impleméntala en el fichero **main.s** (0.5 puntos)

NOTA:

Sólo hay que entregar los 5 ficheros pedidos (y que se han proporcionado en las plantillas). Cualquier constante necesaria se debe definir dentro de cada fichero (y no cargándola de ficheros externos adicionales)

NOTA 2:

¡Y recuerda! Tienes que implementar lo pedido **sin violar** el convenio ABI del RISC-V