EXAMEN DE ARQUITECTURA DE COMPUTADORES

- Convocatoria: EXTRAORDINARIA
- Grado: Ingeniería de robótica Software. URJC
- Fecha: Lunes, 24-Junio-2024

Un DNI español consta de dos partes:

- 8 dígitos, que, para el caso de extranjeros residentes, sustituye el primer dígito por una X, Y o Z
- Letra de control

La letra de control es derivada de la siguiente función aplicada a los 8 dígitos de entrada: El resto de dividir el número formado por los 8 dígitos entre 23 (para el caso de DNI de extranjeros, se sustituye X por 0, Y por 1, Z por 2), es la entrada en una tabla (o array) que indica la letra de control.

Resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Letra	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Ejercicio 1 (6 puntos):

Se debe implementar un programa en lenguaje ensamblador RISC-V que pida el identificador de un DNI español y que compruebe si el número es correcto.

El programa ha de pedir que el usuario introduzca el DNI con el mensaje “DNI:”

El cliente introduce un identificador que cumpla las normas de formato de un DNI (8 dígitos pudiendo ser el primero una X, Y o Z) más la letra de control.

Tras introducir el identificador, se ha de indicar en la siguiente línea lo siguiente:

- CORRECTO.
  En el caso de introducir un DNI correctamente formateado y con la letra de control correctamente calculada
- LETRA DE CONTROL INCORRECTA
  Cuando la letra de control no corresponda a los 8 primeros dígitos.

Ejemplos de ejecución:

08080808S
LETRA DE CONTROL INCORRECTA

#08080808B
CORRECTO

Normas de implementación a cumplir:

- Se ha de definir una subrutina que reciba un identificador de DNI con formato correcto y retorne si la letra de control corresponde con los 8 dígitos de inicio o no
    - La subrutina ha de hacer uso de la subrutina en el fichero string8ToInt.asm que implementa la transformación de string (8 caracteres dígitos (0-9)^8) a entero: string8toInt
    - Las subrutinas implementadas no pueden realizar operaciones de entrada/salida de datos con los dispositivos externos
- Por simplicidad, todas las letras de control serán mayúsculas.
- Como ayuda, todas las letras de control en función del resto están colocadas en el siguiente string: "TRWAGMYFPDXBNJZSQVHLCKE"
- Se recomienda utilizar la plantilla dni.asm, que contiene los mensajes de salida y una recomendación de estructura de programa siguiendo instrucciones anteriores.

Ejercicio 2 (2,5 puntos):

Sobre la base del ejercicio 1, añadir la posibilidad de retornar un error de formato en el DNI introducido.

Se ha de comprobar lo siguiente:

- que la longitud de la entrada es de 9 caracteres.
- que el primer carácter es un dígito o una de las tres letras permitidas: X,Y y Z.
- que los siguientes 7 caracteres son dígitos.
- que el último carácter es una letra

En el caso de error de formato, la aplicación ha de imprimir el siguiente ERROR: “ERROR de FORMATO”

Normas de implementación a cumplir:

- Se ha de definir una subrutina llamada *chequeaFormatoDNI* para la realización de esta tarea, que retorna en registro a0 un valor 0 si el formato es correcto, y otro valor si no lo es.
    - La plantilla provista en ejercicio 1 contiene comentada una propuesta de definición

- o

Ejercicio 3 (1,5 puntos):

Sobre la base del ejercicio 1 (o ejercicio 1 y 2), realizar una modificación para pedir continuamente identificadores de DNI para realizar la comprobación de si son correctos.

Para terminar la ejecución del programa, se ha de introducir el comando ‘q’ (quit) que indica el final.

Ejemplo de ejecución (incluye ejercicio 1, 2 y 3):

```
Bienvenidos a comprobador de DNIs

DNI: **** user input : 08080808B
CORRECTO

DNI: **** user input : 08080808S
LETRA DE CONTROL INCORRECTA

DNI: **** user input : A1234567R
ERROR DE FORMATO

DNI: **** user input : X0000000T
CORRECTO

DNI: **** user input : Y0000000T
LETRA DE CONTROL INCORRECTA

DNI: **** user input : q
```

Entrega

Todos los ficheros que se entreguen han de ensamblar en RARS, de lo contrario se consideran como no válidos.

Se recomienda guardar el fichero fuente que funcione antes de comenzar el siguiente ejercicio para evitar que los siguientes ejercicios penalicen a los realizados anteriormente.

En el caso de tener un ejercicio a medias, se pueden entregar un máximo de dos ficheros indicando en el nombre lo incluido, por ejemplo, si el ejercicio 2 no está completo, pero si lo está el 1 y el 3, se pueden entregar dos ficheros con los siguientes nombres:

- dni_1_3.asm
- dni_1_3_2_incompleto.asm