



Estructura y Tecnología de Computadores

Examen final de prácticas (22 de enero de 2020)

En el archivo `examen-2020-enero-prácticas.tar.gz` se incluyen dos versiones incompletas de un programa «progexamen»: una en C y otra en ensamblador MIPS. El programa, una vez completo, mostrará un menú donde se nos ofrecerán varias opciones para probar los ejercicios del examen que se describen a continuación.

Los códigos C y ensamblador que se proporcionan incluyen el menú y algún código auxiliar y de prueba, además de la cabecera de los procedimientos que se deberán implementar en ensamblador durante el examen. Sin embargo, no se incluye en la versión en C el código de algunos procedimientos que deberán ser diseñados e implementados durante el examen, por lo que no se puede compilar el programa dado. Por ello, también **se incluye una versión precompilada del programa completo para poder comprobar cuál debe ser su funcionamiento exacto una vez terminado.**

Teniendo en cuenta lo anterior, haga lo que piden los siguientes apartados, implementando los procedimientos que se piden en el fichero `progexamen.s`:

1. (1,5 puntos) **Implemente** el procedimiento `preguntar_entero`, que recibe tres parámetros: la dirección de comienzo de una cadena de caracteres que contiene el mensaje que se le debe mostrar al usuario para pedir el entero, el valor mínimo aceptable y el valor máximo aceptable. El procedimiento debe imprimir el mensaje y a continuación leer un entero por teclado. Si el entero es mayor o igual al mínimo y menor o igual al máximo indicados, lo devuelve. En caso contrario, si está fuera de rango, debe mostrar el mensaje «*El entero no es válido.*» y volver a pedirlo. Se aconseja utilizar la función `print_string` que se proporciona ya implementada.
2. (1,75 puntos) **Traduzca** el procedimiento `contar_apariciones`, que recibe un puntero a una cadena y un carácter, y devuelve cuántas veces aparece dicho carácter en el texto sin diferenciar las letras mayúsculas de las minúsculas.
3. (2,25 puntos) **Implemente** el procedimiento `imprimir_rectangulo`, que recibe como argumentos dos enteros (ancho y alto) y un carácter (relleno) y debe imprimir un rectángulo del tamaño indicado y relleno con el carácter indicado. El rectángulo debe estar delimitado horizontalmente por caracteres «-», verticalmente por caracteres «|» y caracteres «+» en las esquinas. Si el ancho o el alto del rectángulo es menor que 2, debe de mostrar un mensaje de error «Rectángulo demasiado pequeño». Por ejemplo, la llamada `imprimir_rectangulo(5, 4, 'x')` debe imprimir lo siguiente:

```
+---+
|xxx|
|xxx|
+---+
```

4. (2,25 puntos) **Traduzca** el procedimiento `sumar_vectores`, el cual recibe tres parámetros: la dirección del vector en el que almacenar el resultado (`resultado`), la dirección de un array de vectores (`array`) y la longitud del array. El procedimiento almacena en el vector resultado la suma de los elementos del array.
5. (2,25 puntos) **Implemente** el procedimiento `concatenar_cadenas`, el cual recibe como parámetros la dirección de un buffer de destino (`dst`) y de dos cadenas (`src_a` y `src_b`). Debe almacenar en el buffer `dst` la concatenación de las cadenas `src_a` y `src_b`. Puede suponer que el buffer siempre tendrá espacio suficiente para almacenar el resultado.

Notas importantes a tener en cuenta para la realización de los ejercicios:

- Los procedimientos auxiliares y el menú están bien implementados y no es necesario ni recomendable gastar tiempo en entender cómo están implementados. Sólo hay que usarlos para comprobar el funcionamiento de los procedimientos implementados durante el examen.

- En el caso de los ejercicios en los que se pide que se traduzca una función, la traducción debe de ser lo más literal posible y se debe evitar cualquier tipo de optimización (en particular, debe respetar fielmente la estructura de los bucles originales).
- En el caso de los ejercicios en los que se pide que se implemente alguna función, se pueden utilizar funciones auxiliares siempre que se considere oportuno.
- A la hora de evaluar, sólo se tendrá en cuenta el código ensamblador (se ignorará cualquier modificación de la versión en C).
- Puede comprobar si su solución se comporta como la solución correcta comparando la salida de su programa con la versión precompilada, pero tenga en cuenta que el código de la solución puede ser incorrecto aún cuando se comporte correctamente.