

**Universidad Rey Juan Carlos**  
**Arquitectura de Computadores/Arquitectura de Sistemas**  
**Audiovisuales II**

**Práctica 2: Utilizando `syscall`**

Katia Leal Algara

**INTRODUCCIÓN:**

Mars proporciona una serie de servicios del sistema por medio de la instrucción `syscall`. Para ejecutar una llamada al sistema, el programa debe

- A. Cargar el código de la llamada al sistema en el registro `$v0`
- B. Cargar el/los argumentos necesarios en los registros `$a0-$a3`
- C. Las llamadas al sistema que devuelven valores los almacenan en el registro `$v0`

**A.** Escribir un programa que calcula e imprima la suma de dos números introducidos por el usuario en tiempo de ejecución. El programa deberá:

- 1)** Leer dos números del usuario
- 2)** Realizar la suma
- 3)** Imprimir el resultado
- 4)** Salir del programa

Además de la instrucción `syscall`, es posible que necesites utilizar instrucciones para el movimiento de datos, como la pseudoinstrucción `move`.

**B.** Escribe un programa que imprima un mensaje por pantalla, por ejemplo, la cadena `"\tHello Word!\n"`. Analiza los datos almacenados en el segmento de datos, ¿cómo se han almacenado los caracteres? Es decir, ¿cómo se representan los caracteres en memoria? Consulta una tabla ASCII para entender la conversión de caracteres en números. ¿Cuántas *palabras* son necesarias para almacenar la cadena? Prueba a modificar la longitud de la cadena añadiendo o eliminando caracteres.

**C.** Modifica el primer programa para que después de calcular la suma de los dos números imprima el siguiente mensaje: "El resultado es: ". A continuación de la cadena se debe imprimir el resultado de la suma.

- D.** Prueba a modificar el programa anterior para que en lugar de leer los números del terminal se correspondan con dos variables de nuestro programa. De nuevo, se debe calcular la suma de las variables, para lo cual primero las tenemos que cargar en registros. Prueba a crear datos de tamaño 8, 16 y 32 bits. En cada caso, comprueba en el segmento de datos el valor almacenado, las palabras necesarias y las direcciones de memoria correspondientes a cada variable.
- E.** Modifica el programa anterior y reserva espacio en el segmento de datos para leer dos números por el terminal, almacenarlos en memoria y después sumar y mostrar su resultado. Prueba primero a reservar 2 y 4 bytes mediante `.space`. ¿Qué sucede en tiempo de ejecución cuando tratas de almacenar el segundo número? ¿puedes explicar por qué se produce este error?. Modifica el programa, pero esta vez declara una variable de tipo `.half` y otra de tipo `.word`. ¿Salta ahora el error? ¿por qué?