



Laboratorio 2. Acercándose al Hardware: Programación en Lenguaje Ensamblador

Objetivos de aprendizaje

- Usar MARS (un simulador para MIPS) para escribir, ensamblar y depurar programas MIPS
- Escribir programas MIPS incluyendo instrucciones aritméticas, de salto y memoria
- Comprender el uso de subrutinas en MIPS, incluyendo el manejo del *stack*
- Realizar llamadas de sistema en MIPS mediante “*syscall*”

Entrega

Sube los archivos creados, junto con tu informe, a través de Google Classroom. **Todos los archivos con código MIPS deben poder ensamblarse y ejecutarse en el simulador MARS, y deben estar debidamente comentados.** Guarda cada programa en un archivo distinto con el nombre de la parte que corresponde. Por ejemplo, tu trabajo de la Parte 1, pregunta A, guárdalo en el archivo “parte1a.asm”.

Parte 1: Escribir programas MIPS

1. Considera el siguiente código. Traduce este código en instrucciones MIPS, y guárdalas en un archivo llamado “program1.asm”. Suponga que las variables *a* y *b* se asignan a los registros *\$s0* y *\$s1* respectivamente. Suponga que la dirección base del arreglo *D* está en el registro *\$s2*. (1.0 punto)

```
while (a < 10) {  
    D[a] = b+a;  
    a += 1;  
};
```

2. Considera el siguiente código. Traduce este código en instrucciones MIPS, y guárdalas en un archivo llamado “program2.asm”. (2.0 puntos)

```
int[] arr = {11, 22, 33, 44};  
arrlen = arr.length; // traduccion de lo de arriba esta dada  
// complete la traduccion de lo siguiente...  
int evensum = 0; // usar $t0 para valores de evensum  
for (int i = 0; i < arrlen; i++) {  
    if (arr[i] & 1 == 0) { // ¿Qué significa esta condicion?  
        evensum += arr[i];  
    }  
}
```



Tu código MIPS debería comenzar con algo así:

```
.data
arr: .word 10 22 15 40
end:
.text
la $s0, arr          # esta instrucción pone la dirección base de arr en $s0
la $s1, end
subu $s1, $s1, $s0
srl $s1, $s1, 2       # ahora $s1 = num elementos en arreglo. ¿Cómo?
```

Todo bajo `.data` hasta `.text` es el segmento de datos, donde podemos declarar datos estáticos como arreglos. Todo debajo `.text` es el segmento de texto o código, donde escribimos las instrucciones del programa. Cuando ni `.data` ni `.text` están presentes en el archivo, se asume que el archivo completo contiene un segmento de texto. Asegúrate de probar tu programa con arreglos de distinto contenido y largos y reporta los resultados.

Parte 2: Uso de `syscall`

- Escribe un programa que lea dos enteros ingresados por el usuario, determine si su diferencia es par o impar y luego imprima este resultado. El cálculo del máximo debe ser realizado en una subrutina, utilizando los registros apropiados para argumentos y salida de un procedimiento. (1.5 puntos)
- A partir del código anterior: en otra subrutina, si el resultado de la diferencia obtenida entre el primer y segundo entero es par, sume el valor de la diferencia al primer entero y luego imprima el resultado obtenido por pantalla. Si la diferencia es impar, sume el valor al segundo entero y muestre el resultado por pantalla. (1.5 puntos)

Salida para diferencia par:

```
Por favor ingrese el primer entero: 31
Por favor ingrese el segundo entero: 11
La diferencia es: 20 (Par)
Nuevo primer entero: 51
-- program is finished running--
```

Salida para diferencia impar:

```
Por favor ingrese el primer entero: 31
Por favor ingrese el segundo entero: 10
La diferencia es: 21 (Impar)
Nuevo segundo entero: 31
-- program is finished running--
```

Considere para su solución también el caso con números negativos.



Para ello, utiliza llamadas de sistema, "syscall", para 1) imprimir en la consola de MARS los mensajes de interacción con el usuario, 2) permitir que el usuario ingrese los números en tiempo de ejecución, 3) imprimir el resultado en la consola, y 4) terminar el programa (exit). (Nota que en la ayuda de MARS puedes encontrar la documentación sobre el uso de syscall).

Informe

El informe para entregar debe contar con lo siguiente:

- Introducción que incluya el problema, solución y objetivos de esta experiencia
- Marco teórico que explique los conceptos necesarios para entender el trabajo desarrollado
- Explicación breve del desarrollo de la solución y cómo se llegó a esta
- Resultados de cada parte del laboratorio
- Conclusiones
- Referencias bibliográficas. Debe citar todas las fuentes externas usadas. Formato APA.

Exigencias

- El informe escrito debe ser entregado en formato PDF. Se sugiere una extensión no mayor a 10 páginas. Tanto el código fuente como el informe deben ser enviados por medio de la plataforma Google Classroom **en un archivo comprimido .zip**, cuyo nombre debe incluir el RUT de el/la alumno/a (ej: lab2_12345678-9.zip).

Descuentos

- Entregas que no cumplan con el formato indicado se descontaran 2.0 puntos.
- Por cada tres faltas ortográficas o gramaticales en el informe, se descontará una décima a la nota del informe.
- En caso de que no se entregue el informe o el código fuente, se evaluará con la nota mínima.
- **Esta permitido realizar consultas a chatgpt sobre funciones, instrucciones, ... ,sin embargo, si el código es similar a la respuesta que da chatgpt al colocar la pregunta en el prompt, su respuesta quedará invalidada y con 0 puntos.**
- La nota del laboratorio será calculada como: $0.7 * \text{código} + 0.3 * \text{informe}$.
- **Fecha entrega diurno:** Lunes 05 de mayo de 2025 a las 23:59 hrs.
- **Fecha entrega vespertino:** Viernes 09 de mayo de 2025 a las 23:59 hrs.