

Entrega E: Programación y Memoria Caché

Objetivos	<ul style="list-style-type: none"> • Comprensión de enunciados complejos • Entender cómo se programan las iteraciones a bajo nivel . • Explorar conceptos necesarios para definir las estructuras de arreglos y los programas para procesarlos • Comprender la necesidad de una jerarquía de memoria • Comprender la naturaleza de los eventos que justifican la tecnología de caché
Criterios de evaluación	<ul style="list-style-type: none"> • Que las rutinas cumplan con su documentación • Que las rutinas hayan sido validadas. • Que los conceptos se hayan relacionado adecuadamente. • Poder adaptar sus programas a nuevos parámetros

Enunciados

1. Escribir una rutina **xor** que cumpla con la siguiente documentación:

Requiere	Dos cadenas de 16 bits en R6 y R7
Modifica	?
Retorna	En R7 el or exclusivo (bit a bit) entre R6 y R7

2. Implementar la rutina **separarPermisosUNIX** a partir de su documentación:

Requiere	<p>en R4 una cadena donde están codificados los permisos de acceso a un archivo UNIX, con el formato 0000000uuugggooo, donde:</p> <ul style="list-style-type: none"> - uuu son los permisos de lectura, escritura y ejecución del usuario - ggg son los mismos permisos del grupo - ooo son los permisos para otros (ni el usuario dueño ni al grupo que pertenece)
Modifica	?
Retorna	En R5 los permisos del usuario, en R6 los permisos del grupo, y en R7 los permisos de otros. En cada caso, los 13 bits más significativos tienen 0s.

Por ejemplo, la cadena 0000000111111111 le da todos los permisos a todos, y con la cadena 0000000111100000 se indica que el *dueño* tiene los 3 permisos, los miembros del *grupo* solo pueden leer, y *otros* no pueden hacer ninguna de las 3 operaciones.

3. Implementar la rutina **elGrupoEscribe** a partir de su documentación:

Requiere	en R4 un registro con los permisos de acceso a un archivo UNIX , con el formato 0000000uuugggooo
Modifica	?
Retorna	En R3 un 1 si al archivo lo puede escribir algún usuario del grupo, y un 0 en R3 en caso contrario

Nota: debe usar la rutina **separarPermisos**.

- Escribir un programa que controle la calidad de la rutina **elGrupoEscribe** (programa de test). Probar en el simulador, obtener capturas y justificar las conclusiones.
- Se quiere diseñar una arquitectura que en vez de una pila tenga un registro llamado CLD (Called) que cuando ocurra un CALL guardase la dirección a donde se debe volver, en vez de ponerlo en una pila. ¿Qué problemas o limitaciones le ve a este enfoque?
- Suponer que se cuenta con la rutina **sumarSM16** (no se la debe implementar), cuya documentación es como sigue:

Requiere	Dos cadenas en R6 y R7 con valores representados en el sistema SM(16)
Modifica	R4 y R5
Retorna	En R3 el resultado de la suma en SM(16)

Escribir un programa que controle la calidad de la rutina (programa de test).

- Escribir la rutina **invierte** que cumpla con la siguiente documentación

Requiere	<ul style="list-style-type: none"> En R0 la dirección de inicio de un arreglo con valores. Cada valor ocupa una celda En R1 la dirección de fin del arreglo
Modifica	??
Retorna	Un nuevo arreglo B que empieza en la dirección B000, que tenga el mismo contenido que A pero en orden inverso. Por ejemplo, si en A se encuentran los valores 0x0004, 0x0F76 y 0x0099, entonces en el arreglo B deben quedar los valores 0x0099, 0x0F76 y 0x0004.

- Agregue a la red conceptual los siguientes conceptos que caracterizan a las memorias : Metodo de acceso, secuencial, directo, aleatorio, asociativo, volatilidad, capacidad de escritura, ROM
- Considerar una computadora con una memoria de 64 celdas de un byte y una memoria cache de correspondencia directa con 4 líneas y bloques de 8 celdas por línea.

- a. Dada una dirección de memoria calcular la cantidad de bits que se destinan a tag, línea y palabra.
- b. Dar los tags y las líneas de la caché en las que se almacenan las direcciones:

Dirección	Tag	Nro. línea
111000		
011001		
111111		
101000		
101001		

10. Suponer que la siguiente rutina está ensamblada a partir de la posición 0000, PC=0000 y SP=FFEF.

```

main: MOV R7, 0x0005 ; iniciar el contador
      MOV R6, 0x0001 ; iniciar el acumulador
arriba: MUL R6, 0x0002
      SUB R7, 0x0001;
      JNE arriba
      RET

```

- a. Enumerar los accesos a memoria que se producen durante la ejecución de la rutina main
- b. Suponer que la memoria principal tiene un tiempo de acceso de 3s ¿Cuánto tiempo se tarda en ejecutar la rutina?
- c. Suponer que se incorpora una memoria caché de mapeo directo sólo 2 de líneas, con capacidad cada una para almacenar un bloque de 4 celdas. Completar la siguiente tabla para indicar si los accesos enumerados (en el inciso a) son aciertos o fallos. Asumir que la caché está inicialmente vacía.

Dirección	Tag	Línea	Palabra	F/A

- d. ¿A qué principio de localidad se puede atribuir **cada uno de los aciertos**? (Agregar una columna a la tabla anterior)
- e. Calcular la tasa de aciertos obtenida en la ejecución de la rutina main
- f. Suponer que la memoria caché tiene tiempo de acceso es de 0,3s ¿Cuánto tiempo se tarda en ejecutar la rutina main? ¿Se obtuvo un tiempo menor al calculado en el inciso b? ¿Es posible mejorarlo aún más?