

Primer Parcial Organización del Computador

FIUBA - 2025-1C

- Nombre y apellido:
- Padrón:
- Hojas adicionales entregadas:

Assembly

Implementar una función en assembly que reciba un string y un caracter, y retorne un nuevo string donde todos los caracteres del string original que no sean el caracter recibido se hayan intercambiado por el caracter `_` y el resto se mantenga igual. La función debe retornar un puntero al nuevo string alocado en la memoria dinámica.

La función debe tener la siguiente firma:

```
char* sitting_ducks(char* str, char marker);
```

Ejemplos de uso:

```
sitting_ducks("axbxcxdx", 'x'); // Retorna un puntero a "_x_x__x"  
sitting_ducks("", 'x'); // Retorna un puntero a ""  
sitting_ducks("hello", 'x'); // Retorna un puntero a "_____"
```

Condiciones a cumplir:

- El string pasado como parámetro es un string válido de C.
- Si el string original es vacío, la función debe retornar un string vacío.
- El string original NO debe ser modificado.
- Si `marker` no se encuentra en el string, el nuevo string debe tener la misma longitud que el original y contener todos los caracteres reemplazados por `_`.
- La función debe ser implementada en assembly.
- No se permite el uso de funciones de la biblioteca estándar de C para manipular strings (como `strlen`, `strcpy`, etc.). La implementación debe ser completamente en assembly.
- Se evaluará el uso correcto de la pila, el cumplimiento de la ABI y el manejo de memoria dinámica.

Microarquitectura

Queremos expandir la arquitectura `OrgaSmall` para soportar el llamado a funciones.

Se pide:

- Describir clara y concisamente los cambios arquitecturales a implementar para agregar las instrucciones `CALL` y `RET`.
- Dar la secuencia de microinstrucciones para las dos instrucciones. Considerar que uno de los parámetros de ambas es un índice del registro que se utilizará como Stack Pointer. Además, la instrucción `CALL` toma como parámetro un inmediato de 8 bits que indica la dirección de memoria a donde saltar.

Representación de la información

En un sistema de software utilizado por el CERN para detección de partículas elementales, se tienen dos magnitudes obtenidas de dos sensores, `A` y `B`, como números en representación IEEE-754 de 32 bits (i.e. punto flotante de precisión simple). A partir de dichos valores se calcula un valor `C` como $C = A + B$. Se sabe que la relación entre el número más grande y el más chico que se puede obtener de las mediciones es de 2^{23} , es decir, A/B como máximo puede dar 2^{23} .

Se pide:

- Determinar si el formato de punto flotante de precisión simple es adecuado para los valores `A`, `B` y `C`. Tener en cuenta que por un lado se requiere muy alta performance computacional, pero por otro no se debe incurrir en errores numéricos groseros (i.e. comparables con las magnitudes de entrada) para el rango de valores mencionado. En caso de que el formato sea adecuado, determinar hasta que valor se podría incrementar

la relación que actualmente da 2^{23} . En caso de que no sea adecuado, proponer un formato que si lo sea.
Justificar claramente la respuesta.

Memoria Cache

Hacer el seguimiento de escrituras y lecturas de la siguiente secuencia de direcciones, en una memoria cache con las siguientes características:

- dirección de 7 bits y memoria direccionable a byte.
- 4 sets
- 2 líneas por set
- 4 bytes por bloque
- 1 byte por lectura/escritura

Suponga que la memoria cache está vacía al inicio y utiliza como política write-back / write-allocate. La política de desalojo es LRU.

op	address	t	s	b	hit/miss/dirty-miss
R	0x58				
R	0x2E				
R	0x10				
R	0x49				
W	0x63				
R	0x15				
R	0x7F				
W	0x1B				
W	0x15				
R	0x55				
W	0x7A				
R	0x29				
R	0x7F				
W	0x2A				
W	0x5F				
R	0x6F				
R	0x1B				

- Mostrar el estado de la memoria caché al final del ejercicio, indicando qué líneas están ocupadas y cuáles están vacías, así como el estado de cada línea (dirty o clean).
- ¿Cambiaría algo si la política de desalojo fuera FIFO?

Paginación

Estamos diseñando un sistema y necesitamos realizar las siguientes traducciones de direcciones virtuales a físicas:

Direcciones:

Virtual	Física
0x00000001	0x00000001
0x0400F007	0x00002007
0x04207042	0x00003042

Para ello, nuestro sistema operativo completó las tablas de paginación como se muestran más abajo.

Se pide:

- Indicar para los siguientes punteros, si la memoria apuntada por ellos está mapeada: si lo está, definir el tamaño máximo que se puede extender el contenido en bytes, y de caso contrario completar las tablas para que lo estén.

```
int32_t *array = 0x04207FF0;  
int32_t *data  = 0x0000A000;  
int32_t *unInt = 0x0000AFFE;
```

- Dar una dirección válida para el Page Directory.

Estructuras de paginación:

Page Directory:

Entry	Page Table
0	0x00010
16	0x00020

Page Tables:

0x00010:

Entry	Page
0	0x00000

0x00020:

Entry	Page
15	0x00002
519	0x00003