

Organización de Computadoras

SEMANA 5

UNIVERSIDAD NACIONAL DE QUILMES

Que vimos?

- Program Counter
- Modularización
- Reuso
- Especificación por contratos
- Llamadas a subrutinas
- Q3

Hoy!

- Pila
- Funciones Push Pop
- SP – Stack Pointer
- CALL y RET – Como trabajan con la pila?

Pila



Pila



ESTRUCTURA PARA ALMACENAR DATOS

Pila

- Push: **Agrega** un elemento en el tope de la pila.
Ponemos un plato en la pila (**APILAMOS**)
Agregamos un elemento sobre el último agregado

Pila

- Push: **Agrega** un elemento en el tope de la pila.
Ponemos un plato en la pila (**APILAMOS**)
Agregamos un elemento sobre el último agregado
- Pop: **Saca** el elemento del tope.
Retiramos un plato de la pila (**DESAPILAMOS**)
Sacamos del tope o primer plato de la pila

SP - Stack Pointer

- Registro interno, igual que PC e IR.

SP - Stack Pointer

- Registro interno, igual que PC e IR.
- Apunta al tope de la pila (Dirección). Espacio disponible de la pila.

SP - Stack Pointer

- Registro interno, igual que PC e IR.
- Apunta al tope de la pila (Dirección). Espacio disponible de la pila.
- Es el registro que nos permite interactuar con la Pila.

SP - Stack Pointer

Entonces:

Si hago PUSH:

Agrega el dato en la dirección que apunta SP.

Luego decrementa en 1 la dirección de SP para poder seguir apilando datos.

SP - Stack Pointer

Entonces:

Si hago POP:

Incrementa en 1 la dirección de SP para ir a buscar el último dato agregado.

Saca el dato que está en el tope de la pila.

Stack Pointer - Pop

| | Dirección | Contenido |
|------|-----------|-----------|
| | 0xFFFF0 | 0x0000 |
| | 0xFFFF1 | 0x0000 |
| SP → | 0xFFFF2 | 0x0000 |
| | 0xFFFF3 | 0x0A53 |
| | 0xFFFF4 | 0x1111 |

Stack Pointer - Pop

| Dirección | Contenido |
|-----------|-----------|
| 0xFFFF0 | 0x0000 |
| 0xFFFF1 | 0x0000 |
| 0xFFFF2 | 0x0000 |
| 0xFFFF3 | 0x0A53 |
| 0xFFFF4 | 0x1111 |

SP →

Pop

Stack Pointer - Pop

| Dirección | Contenido |
|-----------|-----------|
| 0xFFFF0 | 0x0000 |
| 0xFFFF1 | 0x0000 |
| 0xFFFF2 | 0x0000 |
| 0xFFFF3 | 0x0A53 |
| 0xFFFF4 | 0x1111 |

SP →

Pop
0x0A53

Stack Pointer - Push

| Dirección | Contenido |
|-----------|-----------|
| 0xFFFF0 | 0x0000 |
| 0xFFFF1 | 0x0000 |
| 0xFFFF2 | 0x0000 |
| 0xFFFF3 | 0x0A53 |
| 0xFFFF4 | 0x1111 |

SP →

Push
0xFEDE

Stack Pointer - Push

| | Dirección | Contenido | |
|------|-----------|-----------|----------------|
| | 0xFFFF0 | 0x0000 | |
| | 0xFFFF1 | 0x0000 | |
| SP → | 0xFFFF2 | 0x0000 | |
| | 0xFFFF3 | 0xFEDE | Push 0xFEDE |
| | 0xFFFF4 | 0x1111 | |

Stack Pointer - Push

| | Dirección | Contenido | |
|------|-----------|-----------|--------|
| | 0xFFFF0 | 0x0000 | |
| | 0xFFFF1 | 0x0000 | |
| SP → | 0xFFFF2 | 0x0000 | |
| | 0xFFFF3 | 0xFEDE | Push |
| | 0xFFFF4 | 0x1111 | 0x78D3 |

Stack Pointer - Push

SP →

| Dirección | Contenido |
|-----------|-----------|
| 0xFFFF0 | 0x0000 |
| 0xFFFF1 | 0x0000 |
| 0xFFFF2 | 0x78D3 |
| 0xFFFF3 | 0xFEDE |
| 0xFFFF4 | 0x1111 |

Push
0x78D3

Pila

- La arquitectura decide el valor inicial del SP.
- La memoria que ocupa la pila es memoria común y corriente

CALL y RET

- Ejemplo: Ejercicio 23 de la práctica
- 0x00F0 rutina1: MOV R0, R1
CALL rutina2
RET
- 0x00A1 rutina2: MOV R2, R3
RET
- 0x0E00 Principal: CALL rutina2
CALL rutina1
- Como varia la pila?

CALL y RET

- Qué pasaría si la subrutina sobrescribiera el valor de retorno de la pila?

CALL y RET

- Que pasaría si la subrutina sobrescribiera el valor de retorno de la pila?

BOOOM!!!!

¿Que paso hoy?

- Pila
- SP - Stack Pointer
- Como trabajan CALL y RET con la pila.

¿Que Vendrá?

- Sistema de numeración: Enteros
- Signo Magnitud – SM
- Complemento a 1 - CA1
- Complemento a 2 – CA2
- Exceso

Quiero saber más?

- Organización y Arquitectura de computadoras, Stanllings
Apendice 10A – pág 396 – 397.

Gracias ! :)

