

11 Stack Pointer & Frame Pointer



Arquitectura de Computadoras y Ensambladores 1
M.Sc. Luis Fernando Espino Barrios
2024



luisespino@raspberrypi: ~/nfs/arm64



```
luisespino@raspberrypi:~/nfs/arm64 $ cat asld
```

```
#!/bin/bash
```

```
as $1.s -o $1.o
```

```
ld $1.o -o $1
```

```
./$1
```

```
luisespino@raspberrypi:~/nfs/arm64 $ cat asldec
```

```
#!/bin/bash
```

```
as $1.s -o $1.o
```

```
ld $1.o -o $1
```

```
./$1 ; echo $?
```

```
luisespino@raspberrypi:~/nfs/arm64 $
```

Stack Pointer

Stack Pointer

- Es el puntero de la pila ubicado en el registro X28 o SP, almacena la cabeza de la pila en el actual modo del procesador.
- Las operaciones de la pila se analizarán posteriormente. Principalmente para el uso de llamadas a funciones y retornos, también en las funciones recursivas.

Stack Pointer

- Por el momento, se pueden manipular la cabeza de la pila para manejar espacio de memoria que se necesite para la ejecución de un programa, utilizando las operaciones conceptuales de push y pop.
- Para este caso, se utilizan las instrucciones SUB y ADD para asignar y desasignar espacio de memoria para variables locales a la función principal del programa.
- En Arm64 la modificación de SP debe estar alineada a 16 bytes.

Address	Data	
BEFFFAE8	AB000001	← SP
BEFFFAE4		
BEFFFAE0		
BEFFFADC		
⋮	⋮	

(a)

Address	Data	
BEFFFAE8	AB000001	
BEFFFAE4	12345678	
BEFFFAE0	FFEEDDCC	← SP
BEFFFADC		
⋮	⋮	

(b)

Memory

Figure 6.11 The stack (a) before expansion and (b) after two-word expansion

Manejo de la pila

- Si se declaran variables en una función se debe reservar espacio en el stack y luego liberarlo.
- Reservar `sub, sp, sp, #bytes`
- Liberar `add, sp, sp, #bytes`

Preserved	Not preserved
Saved registers: X19–X27	Temporary registers: X9–X15
Stack pointer register: X28 (SP)	Argument/Result registers: X0–X7
Frame pointer register: X29 (FP)	
Link Register (return address): X30 (LR)	
Stack above the stack pointer	Stack below the stack pointer

FIGURE 2.12 What is and what is not preserved across a procedure call. If the software relies on the global pointer register, discussed in the following subsections, it is also preserved.

Manejo de la pila

- Si el número de variables llega al límite de los registros no preservados R0-R3 y se desean utilizar otros se deben guardar los registros preservados a utilizar R4-R11 y luego restaurarlos.
- Guardar PUSH{R4,...,R11, etc}
- Restaurar POP{R4,...,R11, etc}