

PROCEDIMIENTOS

El procedimiento o subrutina es una parte importante de la arquitectura del sistema de cualquier computadora. Un procedimiento es un grupo de instrucciones que usualmente desarrollan una tarea. Un procedimiento es una porción reusable del programa que se almacena en memoria una vez, pero se usa tan seguido como es necesario. Esto hace ahorrar espacio de memoria y hace más fácil de desarrollar programas. La única desventaja de un procedimiento es que hacen que la computadora tome una pequeña cantidad de tiempo para encadenarse al procedimiento y regresar de él. La instrucción CALL se encadena a las subrutinas, y la instrucción RET hace que se regrese de ella.

La pila almacena la dirección de regreso siempre que una subrutina es llamada durante la ejecución del programa. La instrucción CALL empuja en la pila la dirección de la siguiente instrucción. La instrucción RET remueve una dirección de la pila así que el programa regresa a la siguiente instrucción del CALL.

EJEMPLO 1

0000	SUMS	PROC	NEAR
0000 03 C3	ADD	AX,BX	
0002 03 C1	ADD	AX,CX	
0004 03 C2	ADD	AX,DX	
0006 C3	RET		
0007	SUMS	ENDP	
0007	SUMS1	PROC	FAR
0007 03 C3	ADD	AX,BX	
0009 03 C1	ADD	AX,CX	
000B 03 C2	ADD	AX,DX	
000D CB	RET		
000E	SUMS1	ENDP	

En el ensamblador del microprocesador 8088/86, hay algunas reglas para el almacenamiento de subrutinas. Un procedimiento empieza con la directiva PROC y termina con la directiva ENDP. Cada directiva aparece con el nombre del procedimiento. Esta estructura hace fácil de localizar la subrutina en el listado del programa; por ejemplo, la directiva PROC es seguida por el tipo de procedimiento: NEAR o FAR. El Ejemplo 1 muestra como el ensamblador requiere la definición de un procedimiento cercano (intrasegmento) y lejano (intersegmento). Cuando comparamos esos dos procedimientos, la única diferencia es el código de operación de la instrucción de retorno. La instrucción de retorno cercano usa el código de operación C3H y el retorno cercano usa el código de operación CBH. Un retorno cercano obtiene un número de 16 bits de la pila y lo coloca en el apuntador de instrucción. Un retorno lejano obtiene 32 bits de la pila y los coloca en IP y CS.

La mayoría de los procedimientos que son usados por todos los programas (globales) deben ser usados como procedimientos lejanos. Los procedimientos que son usados para una tarea dada (locales) son normalmente definidos como procedimientos cercanos.

CALL

La instrucción CALL transfiere el flujo del programa al procedimiento. La instrucción CALL difiere de la instrucción de salto porque una CALL guarda en la pila una dirección de retorno. La dirección de retorno regresa el control a la instrucción que sigue del CALL en el programa cuando se ejecuta una instrucción RET.

CALL Cercana. La instrucción CALL cercana es de tres bytes de largo con el primer byte conteniendo el código de operación y el segundo y tercero contienen el desplazamiento o distancia de $\pm 32K$. Esta es idéntica a la forma la instrucción de salto cercano. Cuando la CALL cercana se ejecuta, primero coloca en la pila la dirección de desplazamiento de la siguiente instrucción a ejecutar. La dirección de desplazamiento de la siguiente instrucción aparece en el apuntador de instrucción (IP). Después de guardar esta dirección de regreso, entonces suma el desplazamiento del byte 2 y 3 a el IP para transferir el control al procedimiento. No hay instrucciones CALL cortas.

¿Porque guardar el IP en la pila? El apuntador de instrucción siempre apunta a la siguiente instrucción en el programa. Para la instrucción CALL, el contenido de IP es empujado dentro de la pila así que el control del programa pasa a la instrucción seguida de CALL después de que el procedimiento termina. La Figura 1 la dirección de regreso (IP) almacenada en la pila, y la llamada a la subrutina.

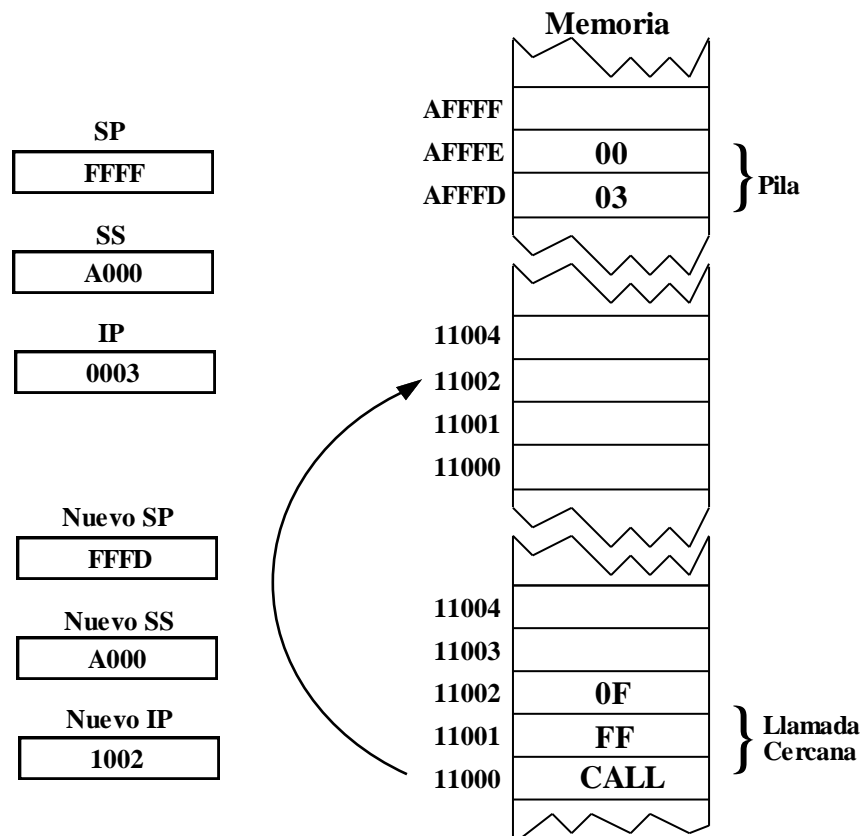


FIGURA 1. Efecto de la instrucción CALL cercana sobre la pila y los registros SP, SS e IP.
(Note que el valor antiguo de IP es almacenado en la pila.)

CALL Lejana. La instrucción CALL lejana es como el salto lejano porque esta puede llamar un procedimiento almacenado en cualquier parte de la memoria en el sistema. La CALL lejana es

una instrucción de 5 bytes que contiene un código de operación seguido por el próximo valor de IP y los bytes 4 y 5 contienen el nuevo valor para CS.

La instrucción CALL lejana coloca el contenido de IP y CS en la pila antes de saltar a la dirección indicada por los bytes 2 al 5 de la instrucción. Esto permite que el CALL lejana llame al procedimiento localizado en cualquier parte de la memoria y regrese del procedimiento.

La Figura 2 muestra como la instrucción CALL lejana llama a un procedimiento lejano. Aquí el contenido de IP y CS son empujados en la pila. Enseguida, el programa salta a el procedimiento.

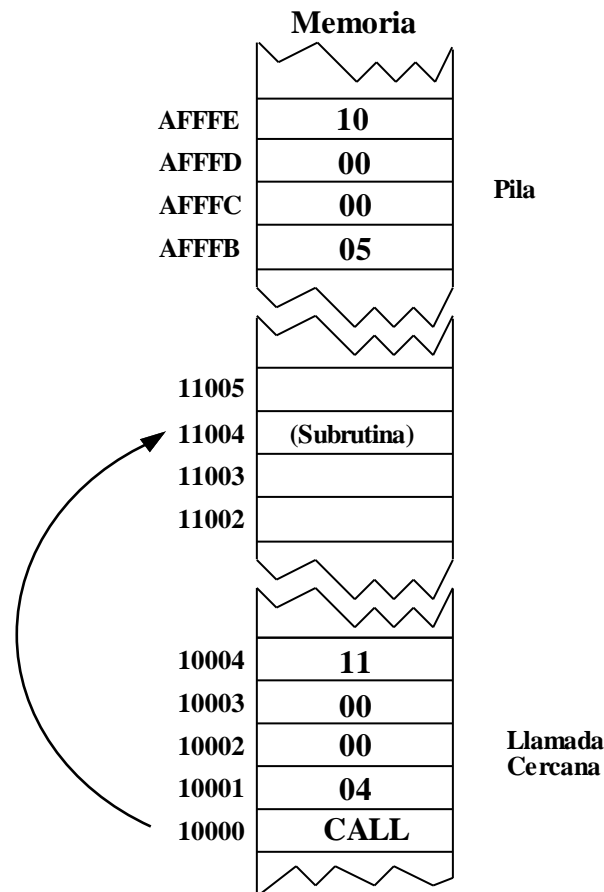


FIGURA 2. Instrucción CALL lejano

CALLs con Operando de Registro. Como los saltos, las CALLs pueden contener también un registro como operando. Un ejemplo es la instrucción CALL BX. Esta instrucción empuja el contenido de IP en la pila. Luego se salta a la dirección de desplazamiento localizado en BX, en el presente segmento de código. Este tipo de CALL siempre usa una dirección de 16 bits almacenado en cualquier registro de 16 bits excepto los registros de segmento.

EJEMPLO 2

```

                                ;llamando una secuencia
0000 BE 0005 R      MOV  SI,OFFSET COMP
0003 FF D6          CALL SI

                                ;procedimiento COMP
0005                COMP PROC NEAR
0006 BA 03F8        MOV  DX,03F8H
0009 EC             IN   AL,DX
000A 42             INC  DX
000B EE             OUT  DX,AL
000C 5A             POP  DX
000D C3             RET
000E                COMP ENDP

```

El ejemplo 2 ilustra el uso de la instrucción CALL con registro para llamar a un procedimiento que empieza en la dirección de compensación COMP. La dirección OFFSET de COMP se coloca en el registro SI, y después la instrucción CALL SI llama a el procedimiento que inicia en la dirección COMP.

CALLs con Direccionamiento Indirecto de Memoria. La CALL con direccionamiento indirecto de memoria es muy útil particularmente siempre que necesitamos escoger una subrutina diferente en un programa. Este proceso de selección se realiza seguido con un número que direcciona una CALL en una tabla de búsqueda.

EJEMPLO 3

```

                                ;Tabla de busqueda
0000 0100 R      TABLA DW  UNO
0002 0200 R      DW      DOS
0004 0300 R      DW      TRES

                                ; secuencia de llamada
0006 BF 0000      MOV  DI,00H                                ; Subrutina UNO
0009 BB 0000 R      MOV  BX,OFFSET TABLA
000C 2E:FF 11      CALL CS:[BX+DI]

                                ;Procedimientos
0100                UNO      PROC  NEAR
                                .
0100                UNO      ENDP
0200                DOS      PROC  NEAR
                                .
0200                DOS      ENDP
0300                TRES      PROC  NEAR
                                .
0300                TRES      ENDP

```

El ejemplo 3 muestra tres subrutinas separadas referenciadas por los números 1, 2 y 3 en AL. La secuencia de llamada ajusta el valor de AL y lo extiende a un número de 16 bits antes de sumarlo a la localidad de la tabla de búsqueda. Esta referencia una de las tres subrutinas usando la instrucción CALL CS:[BX + DI]. El prefijo CS: aparece antes de la instrucción CALL porque TABLA está en el segmento de código en este ejemplo.

La instrucción CALL también puede referenciar apuntadores lejanos si aparece como CALL FAR PTR [SI]. Esta instrucción recupera una dirección de 32 bits de la localidad de memoria del segmento de datos direccionada por SI y la usa como una dirección de un procedimiento lejano.

RET

La instrucción de retorno (RET) obtiene un número de 16 bits de la pila y lo coloca en IP (retorno cercano), o un número de 32 bits y lo coloca en IP y CS (retorno lejano). Las instrucciones de retorno cercano y lejano se definen con la directiva PROC en el procedimiento. Esta selecciona automáticamente la instrucción de retorno apropiada.

Cuando se cambian IP o IP y CS, la dirección de la siguiente instrucción está en una nueva localidad de memoria. Esta nueva localidad es la dirección de la instrucción que sigue inmediatamente después de la llamada más reciente a una subrutina. La Figura 3 muestra como la instrucción CALL se enlaza a un procedimiento y como regresa con la instrucción RET.

Hay otra forma para la instrucción de retorno. Esta forma suma un número a el apuntador de pila (SP) antes del retorno. Si los valores colocados en la pila deben ser borrados antes del retorno, se suma a SP un desplazamiento de 16 bits antes de que el retorno recupere la dirección de retorno de la pila. El efecto de esto es borrar los datos de la pila, o saltar los datos de la pila.

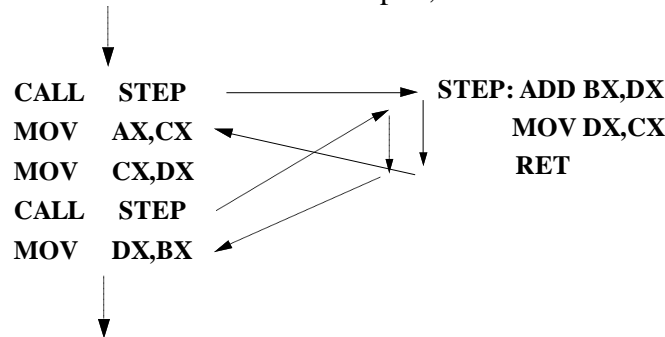


FIGURA 3. Programa y subrutina ilustrando el encadenamiento entre el programa y la subrutina con la instrucción CALL y RET.

EJEMPLO 4

```

0000                TESTS PROC NEAR

0000 50                PUSH AX
0001 53                PUSH BX
                     :
0030 C2 0004          RET 4

0033                TESTS ENDP
  
```

El ejemplo 4 muestra como este tipo de retorno borra los datos colocados en la pila. El RET 4 suma un 4 a el SP antes de remover de la pila la dirección de retorno. Ya que PUSH AX y PUSH BX juntos colocan 4 bytes de datos en la pila, este retorno efectivamente borra AX y BX de la pila. Este tipo de retorno raramente aparece en programas.