Arquitectura de computadoras

Explicación Práctica 1

Subrutinas

- **Uso de subrutinas**
  - **CALL** **nombreSubrutina** → se llama a la subrutina desde el programa principal o desde otra subrutina.
    - Para poder retornar al punto de llamada, se guarda automáticamente en la pila el IP de la siguiente instrucción a la llamada.
    - El IP actual pasa a ser el de la primera instrucción de la subrutina llamada.
  - **RET** → se retorna al punto de llamado (en el programa principal o subrutina desde donde se haya realizado el llamado).
    - Para esto, automáticamente se desapila un elemento y se pone como IP actual. Si la pila no se uso (o se uso de forma correcta) el elemento desapilado será el IP que se guardo al realizar el CALL.

Subrutinas

- **Pasaje de parámetros a subrutinas**

  Se puede pasar datos a una subrutina → **parámetros de entrada**

  Se puede recibir datos de una subrutina → **parámetros de salida**

  - **¿Cómo se pasa un parámetro?**
    - **Por valor**: se pasa una copia del valor original.
    - **Por referencia**: se pasa la dirección de memoria en donde esta almacenado el parámetro, de forma que puedan usar/modificar el valor original.
  - **¿Dónde se pasa un parámetro?**
    - **Por registro**: se guarda el valor/dirección en un registro antes de llamar a la subrutina.
    - **Por pila**: se guarda en la pila el valor/dirección antes de llamar a la subrutina (tener en cuenta que en la pila solo se pueden guardar elementos de 16 bits).

Interrupciones

Son un mecanismo mediante el cual se puede interrumpir el procesamiento normal de la CPU (ejecución secuencial de instrucciones de un programa).

- Según su **origen** se clasifican en:
  - **Internas** → Interrupciones por Software
  - **Externas** → Interrupciones por Hardware

En esta práctica solo se usan las interrupciones por software, que son similares a las llamadas a subrutinas, y se usan por ejemplo para realizar **operaciones de E/S**.

Interrupciones

- **Interrupciones por software**: utilizan posiciones del vector preasignadas. Por eso dichas posiciones no pueden ser utilizadas para definir otras rutinas de interrupción
  - **INT 0** → similar al **HLT** (permite que se terminen las interrupciones)
  - **INT 3** → **DEBUG** (punto de parada para depuración)
  - **INT 6** → **LEER** UN CARACTER DESDE TECLADO
    - **BX** debe tener la dirección donde se almacena el carácter
  - **INT 7** → **ESCRIBIR** EN PANTALLA UNA CADENA DE CARACTERES
    - **BX** debe tener la dirección de inicio de la cadena de caracteres
    - **AL** debe tener la cantidad de caracteres a escribir

Ej. 1: E/S y llamado a subrutina

El siguiente ejemplo suma un vector de 5 números (de 1 digito) mediante una rutina de suma, y luego guarda el resultado en memoria.

```
            ORG 1000H
MSJ         db “Presione 0 para comenzar.”
NROS        db 1, 2, 3, 4, 5
CANT        db 5
TOTAL       db ?
CAR         db ?

            ORG 3000H
sumar:      MOV AL, 0
loop:       ADD AL, [BX]
            INC BX
            DEC CL
            JNZ loop
            RET
```

```
            ORG 2000H
volver:     MOV BX, OFFSET MSJ
            MOV AL, OFFSET NROS - OFFSET MSJ
            INT 7
             MOV BX, OFFSET CAR
            INT 6
            CMP CAR, ‘0’
            JNZ volver
            MOV BX, OFFSET NROS
            MOV CL, CANT
            CALL sumar
            MOV TOTAL, AL
            INT 0

            END
```

Ej. 2: preservación de valores en subrutina

El siguiente ejemplo cuenta la cantidad de letras ‘e’ que tiene una cadena de caracteres mediante una subrutina, y luego guarda el resultado en memoria.

```
            ORG 1000H
MSJ         db “¿Cuantas letras e tengo?”
CANT        db ?

            ORG 3000H
contar:     PUSH AX
            MOV CH, 0
loop:       MOV AL, [BX]
            CMP AL, ‘e’
            JNZ saltear
            INC CH
saltear:    INC BX
            DEC CL
            JNZ loop
            POP AX
            RET
```

```
            ORG 2000H
            MOV BX, OFFSET MSJ
            MOV CL, OFFSET CANT - OFFSET MSJ
            CALL contar
            MOV CANT, CH
            HLT

            END
```