

FUNDAMENTOS DEL SOFTWARE – Relación de problemas 1

1. El método de comunicación de E/S en el que la CPU está esperando hasta que la operación de E/S ha finalizado se conoce como:

Respuesta correcta: a) E/S Programada

Justificación: En la E/S programada, la CPU o procesador encuentra una instrucción con la E/S, por lo que pasa el mandato al módulo de E/S que corresponde. Mientras se atiende a la instrucción de E/S, la CPU comprueba periódicamente el estado de la instrucción del módulo de E/S hasta que dicha instrucción finaliza y la CPU recupera el mandato, es decir, ha estado esperando la finalización de la operación de E/S.

2. El método de comunicación de E/S en el que el dispositivo de E/S informa a la CPU en qué momento está preparado el dispositivo para la transferencia de datos se conoce como:

b) E/S Dirigida por Interrupciones

Justificación: Cuando la CPU lee una instrucción de E/S, pasa el mandato al módulo de E/S y la CPU guarda el contador de programa y registros para ocuparse de otras tareas. Mientras tanto, el módulo de E/S se dedica a leer los datos de un periférico asociado. La CPU, tras cada ciclo de instrucción, comprueba si hay interrupciones de E/S, es decir, si el módulo de E/S ha acabado de realizar la lectura. Cuando se realiza la lectura, el módulo de E/S genera una interrupción (**informa a la CPU**), entonces la CPU guarda los datos de lo que estaba haciendo y ya pasa a leer la palabra del módulo de E/S y a escribirla en memoria.

3.Cuál de las siguientes afirmaciones es correcta:

b) Un programa, para que se ejecute, debe estar cargado en la memoria principal.

Justificación: Como podemos ver en los ejercicios 5 y 9, un programa consta de una serie de instrucciones que deben estar almacenadas en direcciones de memoria, por lo cual es necesario que cada una de sus instrucciones estén cargadas en cada una de las direcciones de memoria.

4. Dado el esquema de un computador elemental según se ha descrito en el tema, el puntero de pila (SP) indica:

b) La dirección de memoria donde se encuentra la dirección donde debe saltar el programa después de ejecutarse la instrucción de retorno correspondiente.

Esto se debe a que el puntero de pila (SP) es un registro de dirección que contiene direcciones de memoria principal de datos e instrucciones y cuya función principal es apuntar a la cima de la pila, donde se van apilando o extrayendo información. Por ejemplo, en las interrupciones la CPU necesita guardar la palabra de estado (PSW) y el contador de programa (PC) en la pila, además de los contenidos de los registros, y todo ello lo almacena guardándolo en la cima de la pila, por lo que el SP se debe actualizar para apuntar a la nueva cima. Una forma más fácil de ver esto es el ejercicio 9 en la sexta línea, pues el puntero pasa a apuntar a la dirección 29 que contiene a su vez la dirección 19 que es justo por donde luego se retomará el programa.

5. Sea un ordenador elemental con una arquitectura tal y como se muestra en la figura, es decir, tres registros de propósito general, registro contador de programa (PC) y registro de instrucción (IR). El registro SP (Puntero de pila) contiene la dirección 35 y la pila crece hacia posiciones menores de memoria. La memoria principal dispone de 256 palabras donde cada palabra tiene la longitud necesaria para albergar la instrucción de mayor tamaño. Describa el estado final de ejecución del procesador a partir del estado actual de la CPU mostrado en la figura. Ponga todos los valores de los registros de cada ciclo de instrucción realizado por el procesador hasta llegar a dicho estado final.

Las instrucciones de las direcciones 10 y 11 ya se han ejecutado.

- 1) PC = 13, IR = SUMAR R0,R1,R2, R0 = 16 R1 = 14 R2 = 30 PD = 12 PE = 0, M[21]=16, M[22]=14
- 2) PC = 14, IR = MOVER R2, 23, R0 = 16 R1 = 14 R2 = 30 PD = 12 PE = 0, M[21]=16, M[22]=14, M[23] = 30
- 3) PC = 15, IR = IN PD,24, R0 = 16 R1 = 14 R2 = 30 PD = 12 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12
- 4) PC = 16, IR = MOVER 23,R0, R0 = 30 R1 = 14 R2 = 30 PD = 12 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12
- 5) PC = 17, IR = MOVER 24, R1, R0 = 30 R1 = 12 R2 = 30 PD = 12 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12
- 6) PC = 18, IR = SUMAR R0,R1,R2, R0 = 30 R1 = 12 R2 = 42 PD = 12 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12
- 7) PC = 19, IR = OUT PD,R2, R0 = 30 R1 = 12 R2 = 42 PD = 42 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12
- 8) PC = 20, IR = HALT, R0 = 30 R1 = 12 R2 = 42 PD = 42 PE = 0, M[21]=16, M[22]=14, M[23] = 30, M[24] = 12

Finaliza la ejecución del programa

6. Suponiendo que el lenguaje máquina de la arquitectura anterior dispone de 14 instrucciones distintas, muestre cuántos bits serían necesarios para codificar las instrucciones SUMAR R₀,R₁,R₂ y MOVER 20,R₀ respectivamente.

Como se dispone de 14 instrucciones distintas, es necesario 4 bits:

$$2^4 = 16 > 14 \text{ instrucciones}$$

Estos 4 bits son comunes a todas las instrucciones.

SUMAR R₀,R₁,R₂: Son necesarios los 4 bits de la instrucción más los bits necesarios para codificar los 3 registros. Como se tienen 3 registros, cada registro requerirá 2 bits para ser codificado pues $2^2 = 4 > 3$. Por lo tanto, en total se requerirán $(4 \text{ bits inst.} + (3 \text{ registros}) \cdot 2 \text{ bits}) = 10 \text{ bits}$.

MOVER 20,R₀: Son necesarios los 4 bits de la instrucción más 2 bits para codificar el registro R₀ y los bits necesarios para codificar una dirección. Estos últimos bits se deducen del número de palabras de la memoria principal, $256 = 2^8 \rightarrow 8 \text{ bits por dirección}$. Por lo tanto, en total se requerirán $(4 \text{ bits inst.} + (1 \text{ registro}) \cdot 2 \text{ bits} + (1 \text{ dirección}) \cdot 8 \text{ bits}) = 14 \text{ bits}$.

7. Imagina que el procesador está ejecutando el programa de usuario del ejercicio 5 y en este momento al terminar de ejecutar la instrucción actual, el procesador se da cuenta de que hay una interrupción pendiente. Escribe los pasos que se dan en el sistema y por quién (software o hardware) hasta que se resuelve el tratamiento de la interrupción y el programa finaliza, sabiendo que la rutina de tratamiento de la interrupción comienza en la dirección de memoria principal 56 y termina en la dirección de memoria principal 70.

La respuesta la hallamos en la diapositiva 22. A nivel hardware:

- 1) El controlador de dispositivos u otro sistema hardware genera una interrupción.
- 2) El procesador termina la ejecución de la instrucción actual.
- 3) El procesador indica el reconocimiento de la interrupción.
- 4) El procesador apila PSW y el PC en la pila de control.
- 5) El procesador carga un nuevo valor en el PC basado en la interrupción. Esto se debe a que cada interrupción tiene su propio programa de tratamiento.

Y luego a nivel software:

- 6) Salva el resto de la información de estado del proceso.
- 7) Procesa la interrupción (ejecutando todas las instrucciones de las direcciones 56 a la 70).
- 8) Restaura la información de estado del proceso.
- 9) Restaura los valores PSW y PC.

8. Basándonos en el ejercicio 7, ¿hay diferencias si en vez de producirse una interrupción se ha producido una excepción? Indique cuales.

Sí hay diferencias, pues una instrucción se produce con un propósito en concreto, como por ejemplo gestionar instrucciones de E/S como ya hemos visto, es decir, se trata de un proceso deseado. Sin embargo, una excepción se trata de un evento inesperado provocado por un error en la ejecución del programa, como por ejemplo un desbordamiento aritmético.

9. Sea un ordenador elemental con una arquitectura tal y como se muestra en la figura, es decir, tres registros de propósito general, registro contador de programa (PC), registro de instrucción (IR) y registro de pila (SP). La memoria principal dispone de 512 palabras donde cada palabra tiene la longitud necesaria para albergar la instrucción de mayor tamaño. Describa el estado final de ejecución del procesador a partir del estado actual de la CPU mostrado en la figura y tras la ejecución del programa(nótese que la instrucción de la dirección 10 ya se ha ejecutado).

- 1) PC = 12, IR = IN PD, M[26] (M[26] = 15), SP = 30, R0 = 5, R1 = 0, R2 = 0, PD = 15, PE = 0, M[26]=15
- 2) PC = 13, IR = MOV M[26], R1, SP = 30, R0 = 5, R1 = 15, R2 = 0, PD = 15, PE = 0, M[26]=15
- 3) PC = 14, IR = COMP R0, R1, SP = 30, R0 = 5, R1 = 15, R2 = 0, PD = 15, PE = 0, M[26]=15
- 4) PC = 15/18, IR = JNE 18, SP = 30, R0 = 5, R1 = 15, R2 = 0, PD = 15, PE = 0, M[26]=15
- 5) PC = 19/20, IR = CALL 20, SP = 29, R0 = 5, R1 = 15, R2 = 0, PD = 15, PE = 0, M[26]=15, M[29]=19
- 6) PC = 21, IR = ADD R0, R1, R2, SP = 29, R0 = 5, R1 = 15, R2 = 20, PD = 15, PE = 0, M[26]=15, M[29]=19
- 7) PC = 22, IR = OUT R2, PD, SP = 29, R0 = 5, R1 = 15, R2 = 20, PD = 20, PE = 0, M[26]=15, M[29]=19
- 8) PC = 23/19, IR = RET, SP = 30, R0 = 5, R1 = 15, R2 = 20, PD = 20, PE = 0, M[26]=15

9) PC = 20, IR = HALT, SP = 30, R0 = 5, R1 = 15, R2 = 20, PD = 20, PE = 0

José Alberto Hoces Castro 1ºDGIIM