

WUOLAH



josemhv

www.wuolah.com/student/josemhv



4270

Tema1_FS_Teoria.pdf

Tema 1 Teoría y Relación Ejercicios Tema 1 Resueltos



1º Fundamentos del Software



Grado en Ingeniería Informática



UGR - Universidad de Granada

TEMA 1 :: FUNDAMENTOS DEL SOFTWARE

- 1º Relación de ejercicios.
- 2º Teoría.

Relación de ejercicios:

1. ¿Qué parte de una computadora realiza operaciones con datos?

- ☒ la CPU.
- (b) la Memoria.
- (c) el hardware de E/S.
- (d) Ninguna de las anteriores.

2. ¿Qué dispositivo de almacenamiento intermedio maneja datos de forma temporal?

- (a) la ALU.
- ☒ un Registro.
- (c) la UC (Unidad de Control).
- (d) un disco duro.

3. ¿Qué unidad puede disponer de dos entradas?

- ☒ una ALU.
- (b) un Registro.
- (c) una CU.
- (d) un disco duro.

4. ¿Qué maneja un registro de la CPU?

- (a) datos.
- (b) instrucciones.
- (c) valores para el CP (Contador de Programa).
- ☒ Cualquiera de los anteriores.

5. Una palabra de memoria es:

- (a) El conjunto de bits que representa una dirección de una posición de memoria.
- (b) La información que se da en el conjunto de las líneas de entrada a la memoria.
- (c) El conjunto de bits que representa la capacidad máxima de la memoria en un instante dado.
- ☒ El contenido de una posición de memoria.

6. Una memoria de 8Mpalabras, con palabras de 32 bits tiene:

- ☒ 32 MB.
- (b) 8 MB.
- (c) 16 MB.
- (d) Ninguna de las anteriores.

FÓRMULA 1: $T(\text{tamaño de memoria}) = P(\text{nº de palabras}) * L(\text{longitud de palabra})$

$$T = 2^3 * 2^{20} * 2^5 = 2^{28} \text{ bits} = 2^{28}/2^{23} \text{ MB} = 2^5 \text{ MB} = 32 \text{ MB}$$

7. Una palabra tiene una longitud de

- (a) 8 bits.
- (b) 16 bits.
- (c) 32 bits.
- ☒ (d) Cualquiera de las anteriores.

8. Si el espacio de direcciones de memoria es de 16MB y la longitud de palabra es 8 bits, ¿cuántos bits se necesitan para acceder a cada palabra?

- (a) 8.
- (b) 16.
- ☒ (c) 24.
- (d) 32.

Usando la Fórmula 1: $T = P * L \rightarrow P = T / L = 16MB / 8bits = 2^4 * 2^{20} / 2^0 = 2^{24}$ palabras.

FÓRMULA 2: $N(n^\circ \text{ de bits}) = \log_2(\text{logaritmo en base dos}) P(\text{palabras de memoria})$

$N = \log_2 2^{24} = 24 \text{ bits.}$

9. Si la memoria tiene una capacidad de 2^{32} palabras, el bus de direcciones necesitará disponer un canal de:

- (a) 8 líneas.
- (b) 16 líneas.
- ☒ (c) 32 líneas.
- (d) 64 líneas.

Usando la Fórmula 2: $N = \log_2 2^{32} = 32bits = 32 \text{ líneas}$

10. 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:

- ☒ (a) E/S Programada.
- (b) E/S Dirigida por Interrupciones.
- (c) DMA.
- (d) E/S a Distancia.

11. 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:

- (a) E/S Programada.
- ☒ (b) E/S Dirigida por Interrupciones.
- (c) DMA.
- (d) E/S a Distancia.

12. 64MB es equivalente a:

- (a) 512 Kbits.
- (b) $256 \cdot 10^{10}$ bits.
- ☒ (c) 512 Mbits.
- (d) Ninguna de las anteriores.

$64MB = 2^6 * 2^3 * 2^{20} = 2^{29} \text{ bits} = 2^9 \text{ Mbits} = 512 \text{ Mbits}$

13. Cuál de las siguientes afirmaciones es correcta:

- (a) En algunas computadoras un programa puede ejecutarse sin necesidad de cargarlo en la memoria principal.
- ☒ (b) Un programa, para que se ejecute, debe estar cargado en la memoria principal.
- (c) Un programa, para que se ejecute, basta con que esté en el disco duro.
- (d) Un programa, para que se ejecute, si está en lenguaje máquina, puede estar en cualquier unidad.

14. El ancho de banda de un bus:

- (a) Es la longitud (medida en pulgadas o centímetros) transversal de la banda donde van embebidos los hilos conductores del bus.
- (b) Representa la cantidad de información que se transfiere a través de él, dada usualmente en Bytes/segundo.
- ☒ (c) Es el número de bits que transmite simultáneamente, en paralelo.
- (d) Es la longitud (medida en pulgadas o centímetros) total de la banda donde van embebidos los hilos conductores del bus, medida entre las unidades más lejanas que interconecta.

15. ¿Cuántos bits hay en 32KB? ¿Y en 64MB? ¿Y en 4GB?

$$32KB = 2^5 * 2^{10} * 2^3 = 2^{18} \text{ bits.}$$

$$64MB = 2^6 * 2^{20} * 2^3 = 2^{29} \text{ bits.}$$

$$4GB = 2^2 * 2^{30} * 2^3 = 2^{35} \text{ bits.}$$

16. Un computador tiene 36 Kpalabras de memoria principal estructurada en palabras de 32 bits ¿Cuántos bytes caben en dicha memoria?

Usando la fórmula 1: $T = 36K * 32 \text{ bits} = 9 * 2^{14} \text{ B}$

17. Dado un computador cuya capacidad máxima de memoria es de 32 MB organizado en palabras de 32bits .

a) ¿Cuántos bits tiene en total?

$$T = 32MB = 2^{28} \text{ bits}$$

b) ¿Cuál es el ancho (número de hilos) de los buses de datos y direcciones?

Usando la fórmula 1: $P = 32MB/32bits = 2^{23} \text{ palabras}$

Usando la fórmula 2: $N = \log_2 2^{23} = 23 \text{ bits}$

Ancho del bus de datos = $L = 32bits = 32 \text{ hilos}$

Ancho del bus de direcciones = $N = 23 \text{ bits} = 23 \text{ hilos}$

18. Suponiendo que una memoria está direccionada por palabras de 32 bits, si su bus de direcciones dispone de 20 hilos, puede direccionar una memoria de hasta:

(a) 1 MB.

☒ (b) 4 MB.

(c) 5 MB.

(d) 2 MB.

Usando la fórmula 2: $20 = \log_2 P \rightarrow P = 2 \text{ elevado } N = 2^{20} \text{ palabras}$

Usando la fórmula 1: $T = 2^{20} * 2^5 = 2^{25} = 32 \text{ Mbits} = 4MB$

19. Suponiendo un computador con una memoria de 128 MB y que direcciona palabras de memoria de 32 bits, ¿cuál sería el tamaño en bits del contador de programa?

Usando la fórmula 1: $P = 128\text{MB}/32\text{bits} = 2^{25}$ palabras

Usando la fórmula 2: $N = \log_2 2^{25} = 25$ bits

Tamaño Contador de Programa(PC) = $N = 25$ bits

20. ¿Cuántos bits serían necesarios para codificar un conjunto formado por 108 caracteres?

Cada carácter se representa con $1\text{B} = 8$ bits , por lo tanto, $108 * 8 = 864$ bits

21. Dado el esquema de un computador elemental según se ha descrito en el tema (para mayor detalle, ver figura del ejercicio 30), el puntero de pila (SP) indica:

(a) La dirección de memoria donde debe saltar el programa después de ejecutarse la instrucción de retorno correspondiente.

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

(c) La dirección de memoria a donde se ha producido el último salto.

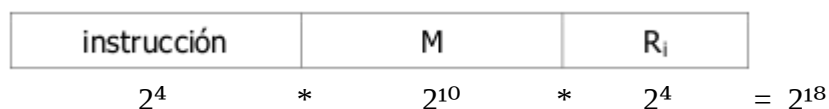
(d) La dirección de memoria donde se encuentra la dirección a donde se ha producido la última llamada a una subrutina.

22. Dado un computador que dispone de 64 MB de memoria principal y una longitud de palabra de 4 bytes. ¿Cuántos bits son necesarios para direccionar cada palabra en memoria?

Usando la fórmula 1: $P = 64\text{MB}/4\text{B} = 2^4$ Mpalabras

Usando la fórmula 2: $N = \log_2 2^{23} = 23$ bits

23. Dado un computador imaginario que dispone de una arquitectura formada por 16 registros de propósito general (R 0 , R 1 , ..., R 15), 1024 palabras de memoria principal y 16 instrucciones diferentes (suma, resta, etc.). ¿Cuál sería la longitud de una instrucción en bits si tiene el siguiente formato:



Donde M es una dirección de memoria y R_i es uno de los registros.

Registros = $R_i = 16 = 2^4$ registros

Palabras = direcciones = $M = 2^{10}$ palabras

Instrucciones = $16 = 2^4$ instrucciones

Longitud de una instrucción con el formato anterior = 18 bits

24. Dado un computador imaginario que dispone de una arquitectura formada por 16 registros de propósito general (R 0 , R 1 , ..., R 15), 1024 palabras de memoria principal y 16 instrucciones diferentes (suma, resta, etc.). ¿Cuántos bits serían necesarios para direccionar a los registros?

Registros = $R_i = 16 = 2^4$ registros los cuales se direccionan con 4 bits.

25. Dado un computador imaginario que dispone de una arquitectura formada por 16 registros de propósito general (R 0 , R 1 ,..., R 15), 1024 palabras de memoria principal y 16 instrucciones diferentes (suma, resta, etc.) ¿Cuál sería el tamaño de las siguientes instrucciones?

Tomando el tipo de instrucción del ejercicio 23: sería 4 bits debido a la instrucción + 10 bits por cada contenido en memoria implicado + 4 bits por registro.

Ejemplo:

ADD Ri,Rj,Rk $\rightarrow 4 + 4 + 4 + 4 = 16$ bits.

(a) Transferencia entre una dirección de memoria y un registro (M[d] \leftarrow Ri).

$4 + 4 + 10 = 18$ bits = tamaño IR

(b) Operación aritmética donde intervienen 3 registros (Ri \leftarrow Rj op Rk).

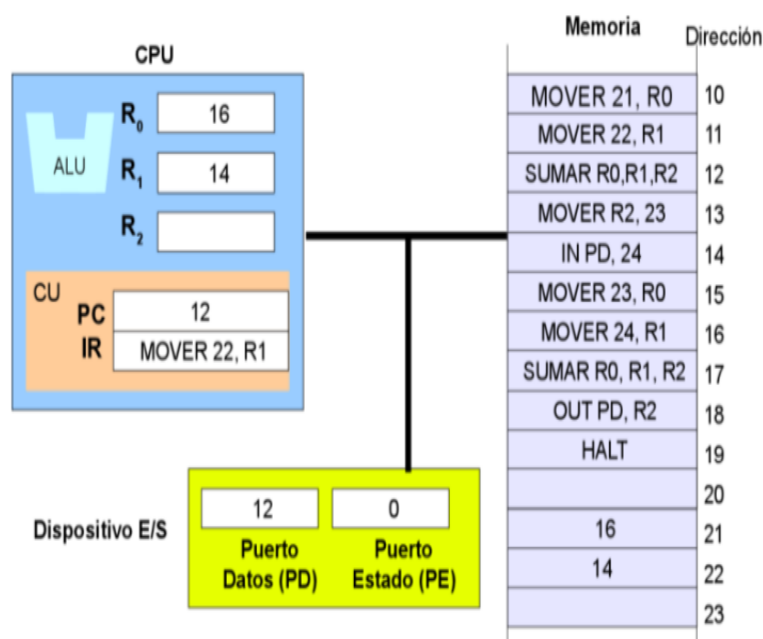
$4 + 4 + 4 + 4 + 4 = 20$ bits

26. Dado un computador imaginario que dispone de una arquitectura formada por 16 registros de propósito general (R 0 , R 1 ,..., R 15), 1024 palabras de memoria principal y 16 instrucciones diferentes (suma, resta, etc.) ¿Cuál sería el tamaño del PC?

$1024 = 2^{10}$ palabras.

Tamaño del PC = $\log_2 2^{10} = 10$ bits

27. 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). 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.



Instrucción	Descripción
MOVER Orig, Dest	Copia el valor del origen (Orig) al destino (Dest).
SUMAR Ri, Rj, Rk	Suma el valor de Ri y Rj, depositando el resultado en Rk.
IN Pi, Dest	Lee del Puerto (Pi) y lo deposita en el destino (Dest).
OUT Pi, Orig	Escribe el contenido del origen (Orig) en el puerto Pi.
HALT	Detiene al procesador.

PC	PC+1	IR	R0	R1	R2	PD	PE	Comentarios
11	12	MOVER 22,R1	16	14		12	0	Dirección 22 = 14 → R1
12	13	SUMAR R0,R1,R2	16	14	30	12	0	
13	14	MOVER R2,23	16	14	30	12	0	R2 = 30 → Dirección 23
14	15	IN PD,24	16	14	30	12	0	PD = 12 → Dirección 24
15	16	MOVER 23,R0	30	14	30	12	0	Dirección 23 = 30 → R0
16	17	MOVER 24,R1	30	12	30	12	0	Dirección 24 = 12 → R1
17	18	SUMAR R0,R1,R2	30	12	42	12	0	
18	19	OUT PD,R2	30	12	42	42	0	
19	20	HALT	30	12	42	42	0	FIN

28. 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.

Instrucciones = 14

Palabras en Memoria = $256 = 2^8$

Registros = 3 --> (tres de propósito general (R0, R1, R2))

Instrucciones:

$$2^4 = 16 < 23 < 32 = 2^5$$

Registros:

$$2^1 = 2 < 3 < 4 = 2^2$$

$$\text{SUMAR} \mid \begin{array}{c} \text{R0,} \\ 2^4 \end{array} \mid \begin{array}{c} \text{R1,} \\ 2^2 \end{array} \mid \begin{array}{c} \text{R2} \\ 2^2 \end{array} \mid = 2^{10}$$

$2^{10} \rightarrow 10$ bits se necesitan para codificar dicha instrucción.

$$\text{MOVER} \mid \begin{array}{c} 20 \\ 2^4 \end{array} \mid \begin{array}{c} \text{R0} \\ 2^2 \end{array} \mid = 2^{14}$$

$2^{14} \rightarrow 14$ bits se necesitan para codificar dicha instrucción.

29. Suponiendo que el lenguaje máquina de la arquitectura descrita en el ejercicio 27 dispone de 23 instrucciones distintas y posee 4096 palabras de memoria, muestre cuántos bits serían necesarios para codificar las instrucciones IN Pi , DD (suponiendo que hay 64 puertos) y MOVER 23,R0 .

Instrucciones = 23

Palabras en Memoria = $4096 = 2^{12}$

Registros = 3 --> (R0, R1, R2,)

Puertos = $64 = 2^6$

Instrucciones:

$$2^4 = 16 < 23 < 32 = 2^5$$

Registros:

$$2^1 = 2 < 3 < 4 = 2^2$$

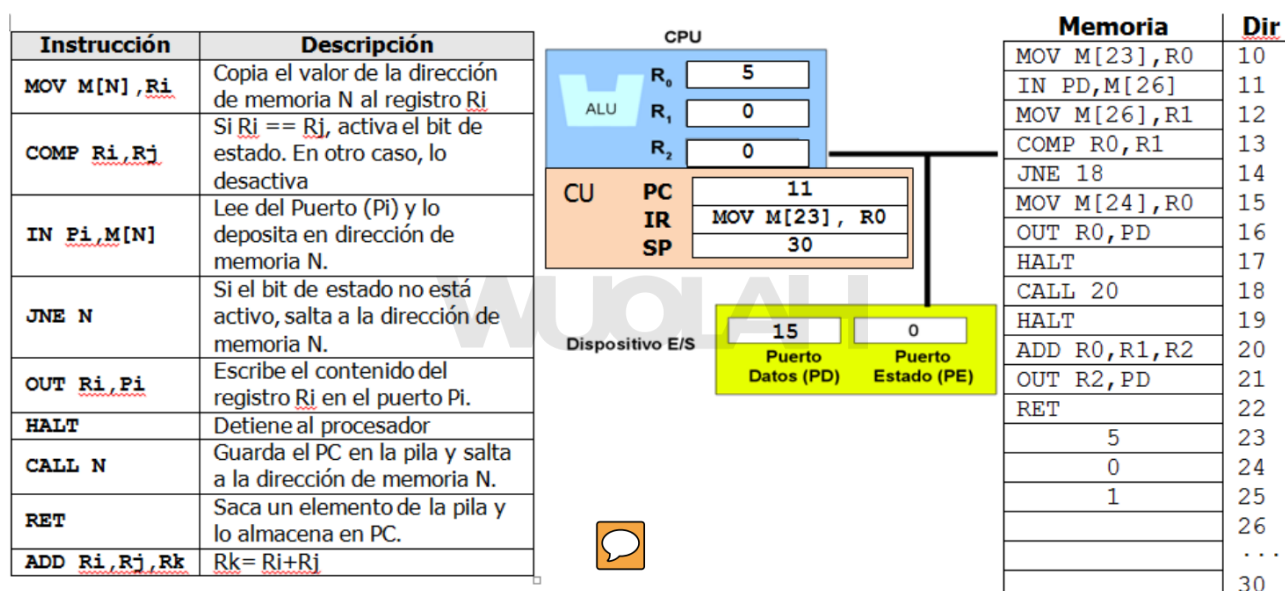
$$\begin{array}{c|c|c} \text{IN} & \text{Pi} & \text{DD} \\ 2^5 * & 2^6 * & 2^{12} = 2^{23} \end{array}$$

$2^{23} \rightarrow 23$ bits se necesitan para codificar dicha instrucción.

$$\begin{array}{c|c|c} \text{MOVER} & 23 & \text{R0} \\ 2^5 * & 2^{12} * & 2^2 = 2^{19} \end{array}$$

$2^{19} \rightarrow 19$ bits se necesitan para codificar dicha instrucción.

30. 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).

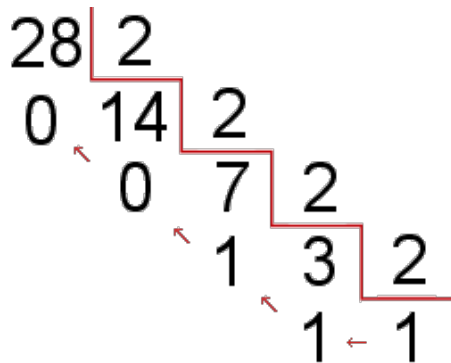


PC	PC+1	SP	IR	R0	R1	R2	PD	PE	M[d]
10	11	30	MOV M[23],R0	5	0	0	15	0	5 = M[23] → R0
11	12	30	IN PD,M[26]	5	0	0	15	0	15 = PD → M[26]
12	13	30	MOV M[26],R1	5	15	0	15	0	15 = M[26] → R1
13	14	30	COMP R0,R1	5	15	0	15	0	
14	15	30	JNE 18	5	15	0	15	0	
18	19	19	CALL 20	5	15	0	15	0	
20	21	19	ADD R0,R1,R2	5	15	20	15	0	
21	22	19	OUT R2,PD	5	15	20	20	0	
22	23	30	RET	5	15	20	20	0	
19	20	30	HALT	5	15	20	20	0	

Teoría:

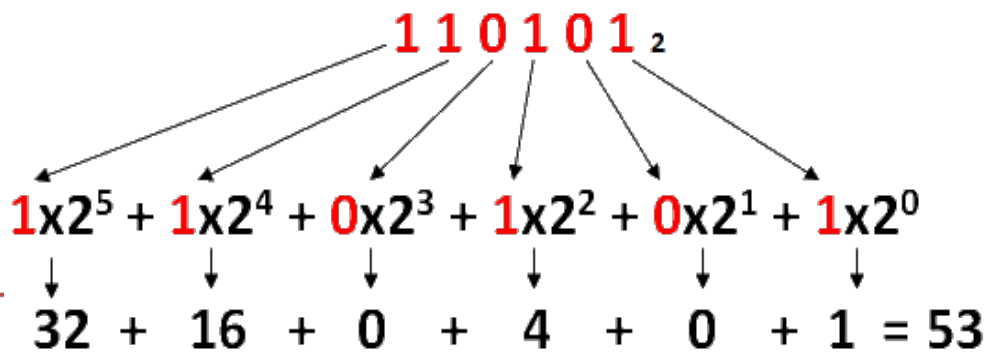
- Bit : unidad mínima de información.
- Byte : unidad mínima para direccionar.

Pasar de Decimal a Binario:



$$28 = 11100_2$$

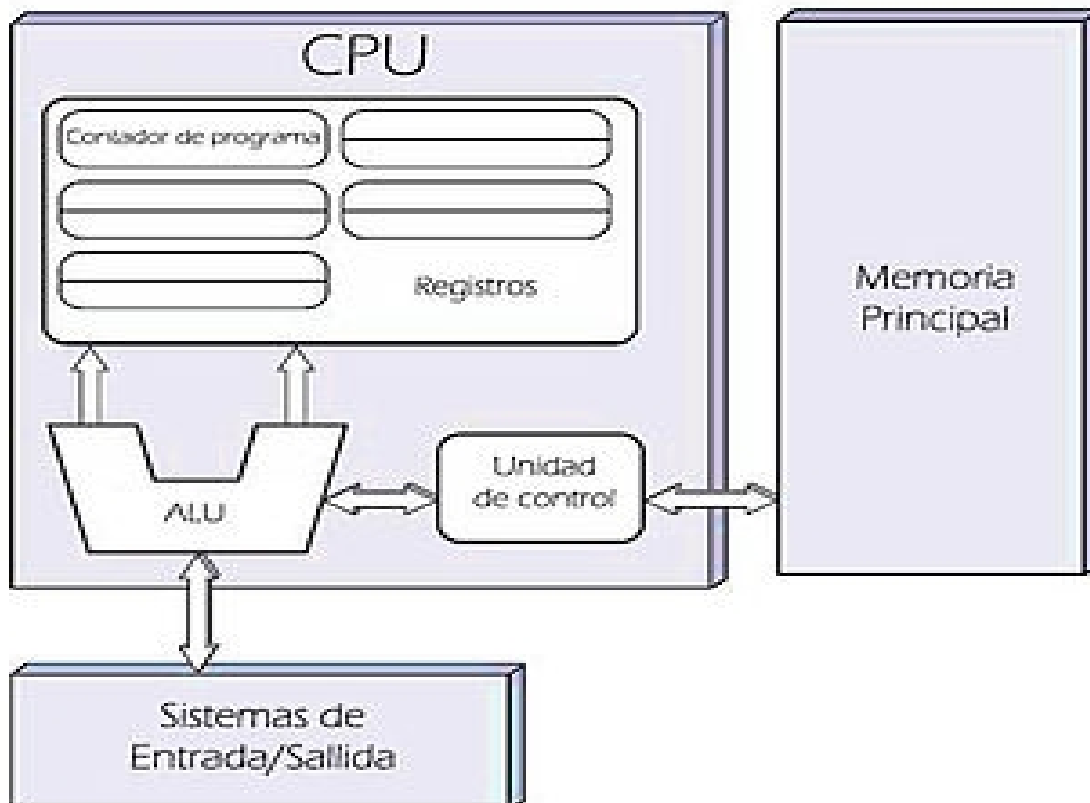
Pasar de Binario a Decimal:



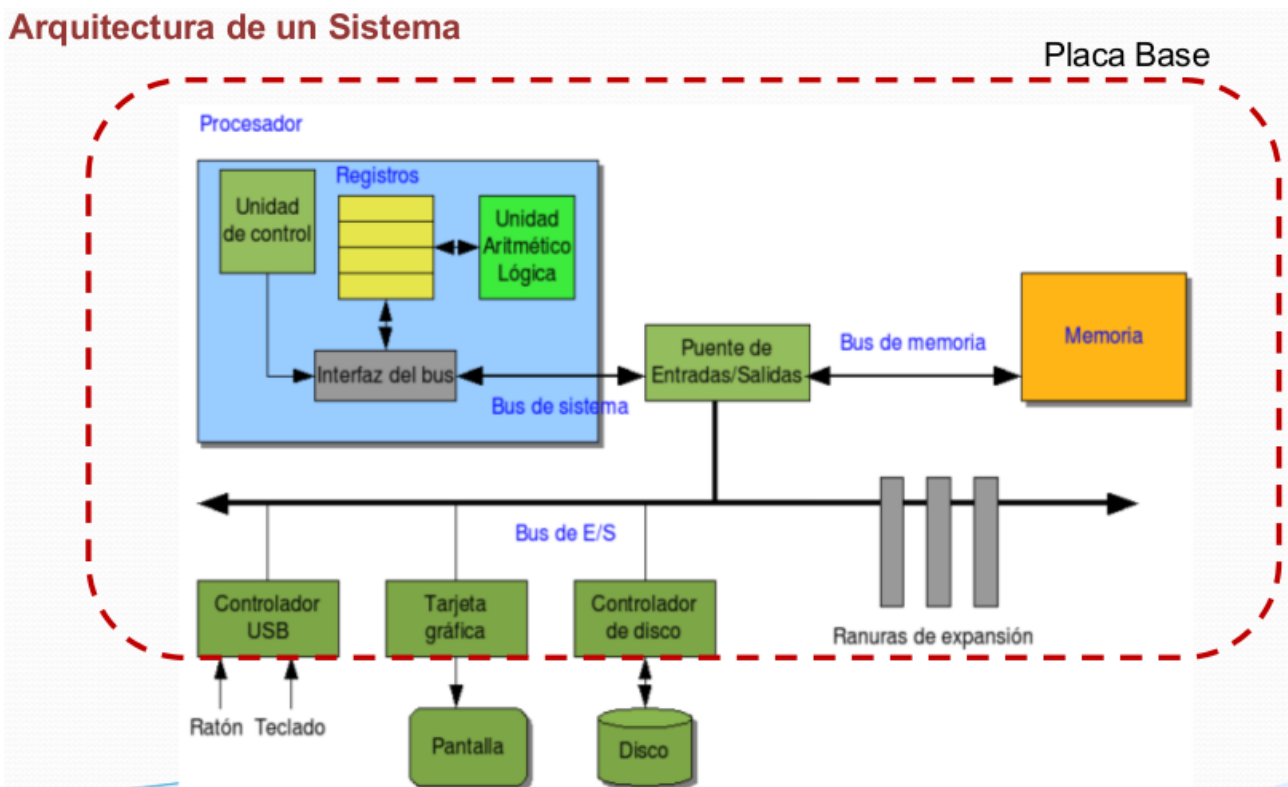
$$110101_2 = 53_{10}$$

- Instrucción: conjunto de símbolos insertados en una secuencia estructurada o específica que el procesador interpreta y ejecuta.
- Datos: Símbolos que representan hechos, condiciones, situaciones o valores. Elementos de información.

ARQUITECTURA DE VON NEUMANN:



Arquitectura de un Sistema



COMPONENTES:

1. **Procesador(CPU):**

-**Unidad Aritmético Lógica(ALU)** : Es la encargada de realizar operaciones tanto aritméticas como lógicas. Posee dos entradas de datos conectadas a los registros y a la UC y una salida con el resultado de la operación conectada a la UC y a los Sistemas de E/S.

-**Unidad de Control(UC)** : dirige y coordina la mayoría de las operaciones en la computadora estando conectada tanto a la memoria principal como a la ALU.

-**Registros del Procesador [Stal05] (pp.11-13)(resumido):**

- **Registros de básicos de control y estado (propósito específico):**

-**Contador de programa(PC)**: Contiene la dirección de la próxima instrucción que se leerá de la memoria.

-**Puntero de pila(SP)**: Apunta al último dato introducido en pila.

-**Registro de instrucción(IR)**: Contiene la última instrucción leída.

-**Registro de estado (bits informativos) o palabra de estado (PSW)**: Almacenan códigos de condición, vienen a ser unos indicadores que responden a cierta condición almacenada en el código.

- **Registros de propósito general (R0,R1,...):**

-Son visibles para el usuario/programa y se suelen usar para almacenar

temporalmente datos de forma mas rápida que el almacenamiento en memoria. Conectados con la ALU.

2. **Memoria Principal:** es el dispositivo donde se almacenan temporalmente tanto los datos como los programas que la CPU está procesando o va a procesar en un determinado momento. Por su función, es una amiga inseparable del microprocesador, con el cual se encuentra conectada.

3. **Dispositivos de Entrada y Salida (E/S):** Son aquellos que permiten la comunicación entre la computadora y el usuario. Conectados tanto a la CPU como a la Memoria Principal.

*Todas las conexiones entre estos componentes se realizan a través de Buses.

EJECUCIÓN DE INSTRUCCIONES:

Procesar una instrucción consta de dos pasos:

1. El Procesador lee (busca) instrucciones de la memoria, una cada vez.
2. El Procesador ejecuta cada instrucción.

Se denomina ciclo de instrucción al procesamiento requerido por una única instrucción.

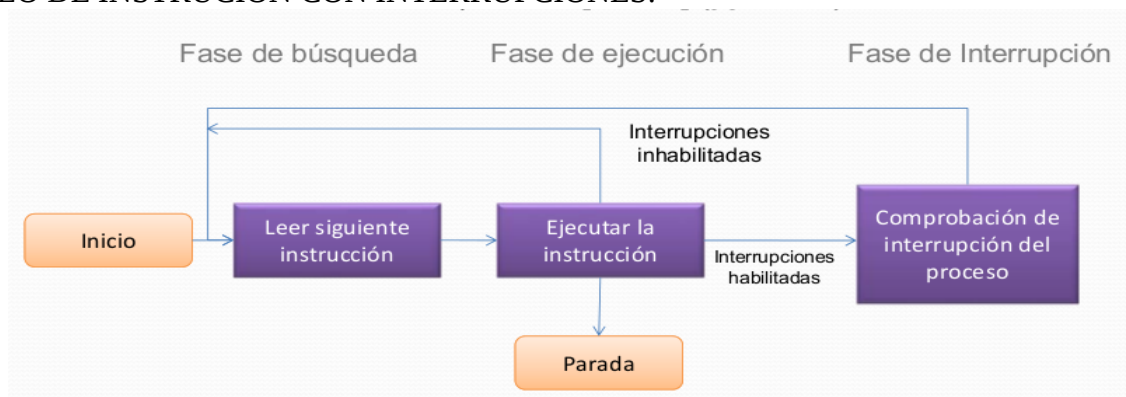
La ejecución de un programa consiste en repetir el proceso de búsqueda y ejecución de instrucciones.



Proceso a seguir:

1. Se lee la instrucción cuya dirección está en el PC.
2. Se incrementa el PC en una unidad.
3. Se ejecuta la instrucción.

***CICLO DE INSTRUCCIÓN CON INTERRUPCIONES:**



TIPOS DE INSTRUCCIONES:

- **Transferencia de datos:** Se transfieren datos desde el procesador a la memoria o viceversa.
Ejemplo: MOV origen, destino
...
- **Aritmético-lógicas:** El procesador realiza algunas operaciones aritméticas o lógicas sobre los datos.
Ejemplo: ADD op1, op2, resultado
COMP Ri, Rj
MULT op1, op2, resultado
...
- **Entradas/salidas:** Envían datos a un dispositivo periférico o los reciben desde el mismo, transfiriéndolos entre el procesador y un módulo de E/S.
Ejemplo: IN puerto, destino
OUT origen, puerto
...
- **Control:** Una instrucción puede especificar que se va a alterar la secuencia de ejecución.
Ejemplo: JUMP/JNE/JE etiqueta
CALL función
RETURN e IRETURN
HALT
...

TÉCNICAS DE COMUNICACIÓN DE E/S :

- **E/S Programada:** El procesador encuentra una instrucción con la E/S. Se genera un mandato al módulo de E/S apropiado.

El procesador adopta un papel activo mientras se atiende la instrucción de E/S y comprueba periódicamente el estado de la ejecución del módulo de E/S hasta que ha finalizado la operación.

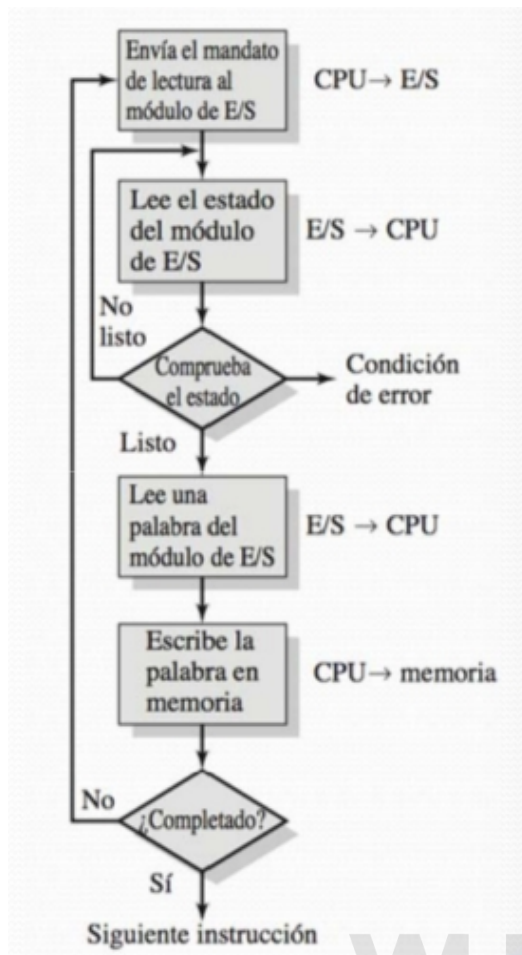
Problema: El procesador pasa mucho tiempo esperando la finalización del módulo de E/S y la respuesta del sistema se degrada gravemente.

- **E/S Dirigida por Interrupciones:** El procesador encuentra una instrucción con la E/S. Se genera un mandato al módulo de E/S apropiado.

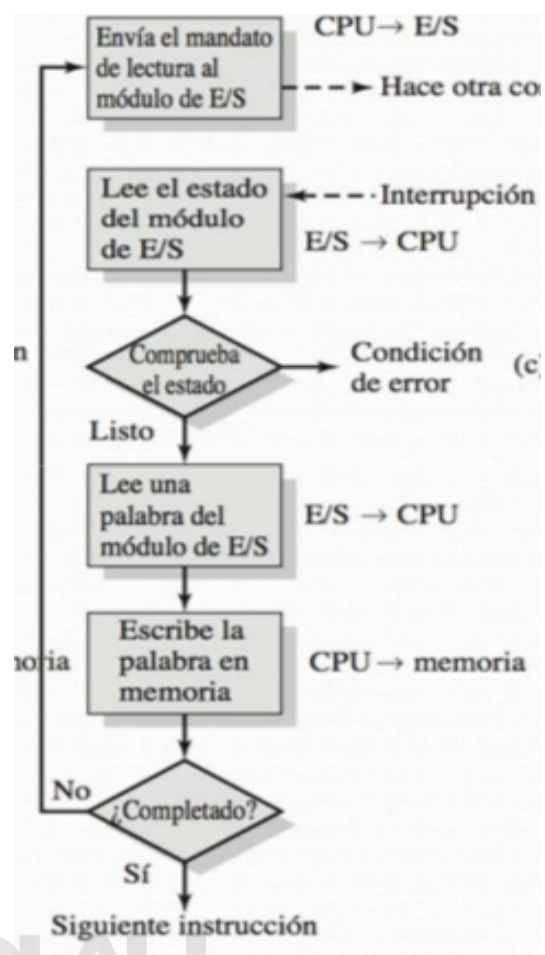
Mientras se atiende al módulo de E/S, el procesador continua realizando otro trabajo útil. Una vez que es atendido el módulo de E/S produce una interrupción al procesador para continuar con la instrucción que genero el mandato al módulo de E/S de la cual anteriormente se salvo su contexto(PC,etc).

Interrupción: Evento que interrumpe el flujo normal de ejecución y que está producido por un elemento externo al procesador. Es un evento asíncrono.

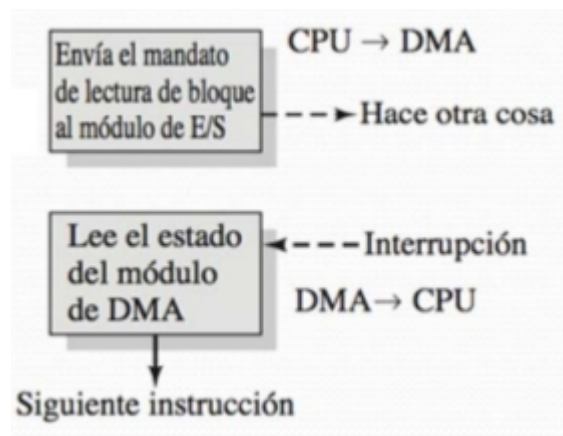
Problema: En transferencias considerables de memoria a dispositivo o viceversa conlleva un uso excesivo del procesador.



E/S PROGRAMADA



E/S DIRIGIDA POR INTERRUPCIONES



ACCESO DIRECTO A MEMORIA(DMA)

- **Acceso Directo a Memoria (DMA, Direct Memory Access):** Realizada por un módulo separado conectado en el bus del sistema o incluida en un módulo de E/S. Útil cuando el procesador desea leer o escribir un bloque de datos.

//DE AQUI PARA ABAJO ESTA COPIADO TAL CUAL DE LAS DIAPOSITIVAS...

EXCEPCIONES:

- **Definición de excepción:** Evento inesperado generado por alguna condición que ocurre durante la ejecución de una instrucción máquina (ejemplo, desbordamiento aritmético, dirección inválida, instrucción privilegiada, etc.). Es un evento síncrono.

Un conjunto predefinido de excepciones las maneja o resuelve el Sistema Operativo (por ejemplo, fallos de memoria).

No todas las excepciones están relacionadas con errores desde el punto de vista del software.

Un programa durante su ejecución también puede tratar algunas excepciones (por ejemplo, en Java se pueden controlar algunos casos).

PROTECCIÓN DEL PROCESADOR:

- Funcionamiento en Modo Dual (es un mecanismo hardware). ¿Qué ocurre si un programa accede a la memoria donde se alojan los vectores de interrupciones? ¿Qué pasa si las modifica?
- Solución: El procesador dispone de diferentes modos de ejecución de instrucciones:
 - Instrucciones privilegiadas (modo supervisor/kernel): Aquellas cuya ejecución puede interferir en la ejecución de un programa cualquiera o programa del SO (ejemplo, escribir en el puerto de un dispositivo).
 - Instrucciones no privilegiadas (modo usuario): Aquellas cuya ejecución no presenta ningún problema de seguridad para el resto de programas (ejemplo, incrementar un contador).

PROTECCIÓN DE LOS DISPOSITIVOS DE E/S:

- Los dispositivos de E/S son recursos que han de estar protegidos (ejemplo, los archivos, las impresoras, ...)
- ¿Cómo se consigue? → Las instrucciones máquina para acceso a los dispositivos de E/S no pueden ejecutarse en modo usuario: son privilegiadas.
- Cualquier acceso a los dispositivos desde un programa de usuario se hará mediante peticiones al SO.

PROTECCIÓN DE MEMORIA:

- Cada programa en ejecución requiere de un espacio de memoria.
- Objetivo: Hay que proteger la zona de memoria asignada para cada programa y la memoria en la que está el código del sistema operativo y sus datos (tabla de vectores de interrupción, rutinas de tratamiento de cada interrupción).
- Unidad de Gestión de Memoria (MMU – Memory Management Unit): Hardware especial para controlar las regiones de memoria asignadas a los programas y velar por su protección.

EL SISTEMA OPERATIVO:

- Un SO es un programa o conjunto de programas que controla la ejecución de los programas de aplicación y que actúa como interfaz entre el usuario de una computadora y el hardware de la misma.



- **El SO como interfaz Usuario/Computadora:**

-Presenta al usuario una máquina abstracta más fácil de programar que el hardware subyacente:

-Oculta la complejidad del hardware.

-Da tratamiento homogéneo a diferentes objetos de bajo nivel (archivos, procesos, dispositivos, etc.).

-Una aplicación se puede expresar en un lenguaje de programación y la desarrolla un programador de aplicaciones.

-Es más fácil programar las aplicaciones en lenguajes de alto nivel que en el lenguaje máquina que entiende el hardware.

-Un SO proporciona normalmente utilidades en las siguientes áreas:

-Desarrollo de programas (editores de texto, compiladores, depuradores de programas).

-Ejecución de programas (cargador de programas y ejecución de éstos).

-Acceso a dispositivos de E/S (cada dispositivo requiere su propio conjunto de instrucciones).

-Acceso al sistema (En sistemas compartidos o públicos, el SO controla el acceso y uso de los recursos del sistema: Shell, Interfaz gráfico).

-Detección y respuesta a errores (tratamiento de errores a nivel software y hardware).

-Contabilidad (estadísticas de uso de los recursos y medida del rendimiento del sistema).

- **El SO como Administrador de Recursos:**

-Un computador es un conjunto de recursos y el SO debe gestionarlos y para ello posee un mecanismo de control que cubre dos aspectos:

-Las funciones del SO actúan de la misma forma que el resto del software, es decir, son programas ejecutados por el procesador.

-El SO frecuentemente cede el control y depende del procesador para volver a retomarlo.

-Por lo tanto:

-El SO dirige al procesador en el uso de los recursos del sistema y en la temporización de la ejecución de otros programas.

-Una parte del código del SO se encuentra cargado en la memoria principal (kernel y, en ciertos momentos, otras partes del SO que se estén usando).

-El resto de la memoria está ocupada por programas y datos de usuario.

-La asignación de la memoria principal la realizan conjuntamente el SO y el hardware de gestión de memoria del procesador.

-El SO decide cuándo un programa en ejecución puede usar un dispositivo de E/S y también el acceso y uso de los ficheros. El procesador es también un recurso.

- **Características deseables en un Sistema Operativo:**

-Comodidad en el uso del computador.

-Eficiencia: Existen más programas que recursos. Hay que repartir los recursos entre los programas.

-Facilidad de Evolución: Un SO importante debe evolucionar en el tiempo por las siguientes razones:

-Actualizaciones del hardware y nuevos tipos de hardware.

-Mejorar y/o aportar nuevos servicios.

-Resolución de fallos.

- **Programas de Servicio del SO:**

-Se trata de un conjunto de programas de servicio que, en cierta medida, pueden considerarse como una ampliación del SO:

- Compactación de discos.
- Compresión de datos.
- Gestión de comunicaciones.
- Navegadores de internet.
- Respaldo de seguridad.
- Recuperación de archivos eliminados.
- Antivirus.
- Salva-pantallas.
- Interfaz gráfica.

- **Herramientas Generales:**

-Su misión es facilitar la construcción de las aplicaciones de los usuarios, sea cual sea la naturaleza de éstas, tales como:

- Editores de texto.
- Compiladores.
- Intérpretes.
- Enlazadores.
- Cargadores/Montadores.
- ...