

Tarea programada 1

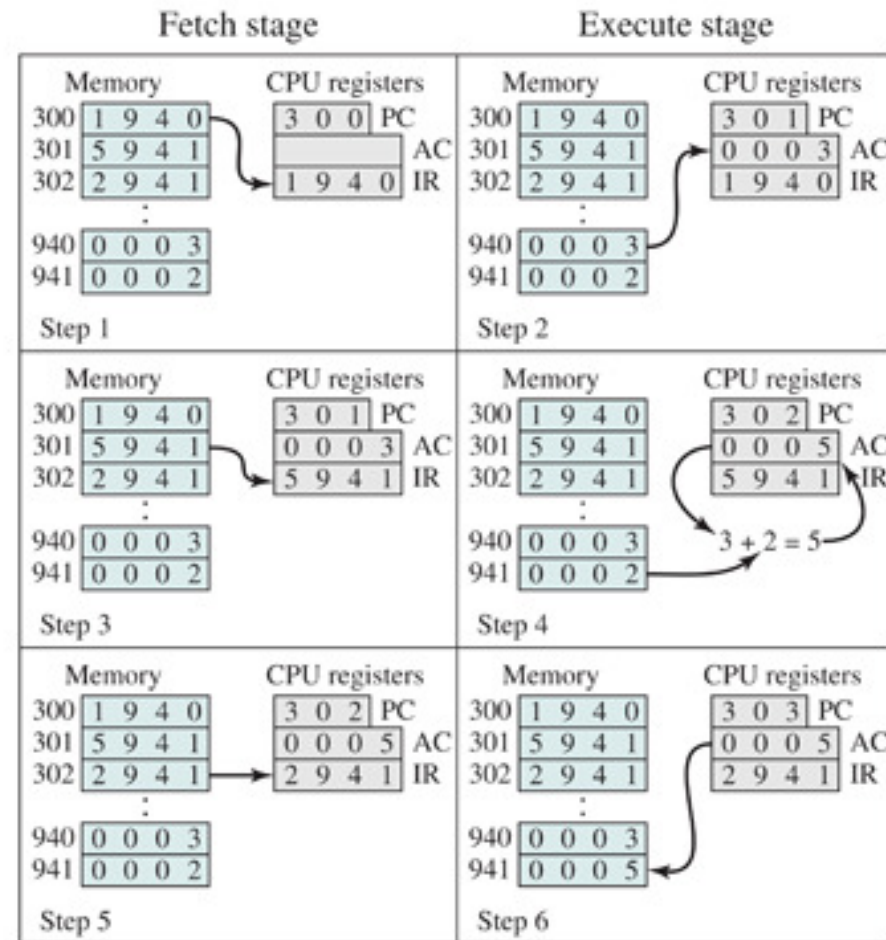
PRINCIPIOS DE
SISTEMAS
OPERATIVOS

Aspectos administrativos

- ❑ Valor de 5%
- ❑ Parte del rubro de Tareas e Investigaciones
- ❑ Fecha de entrega: Domingo 28 de agosto
- ❑ Java, se debe ejecutar paso a paso
- ❑ Debe ser 100% gráfico
- ❑ Carga un archivo [*.asm] con las operaciones a ejecutar
- ❑ Debe de estar guardado en Github

Mini PC

- Memoria
- AC
- IR
- PC
- AX, BX, CX y DX



En qué consiste

- ❑ Construir una aplicación que ejecute instrucciones de un mini lenguaje ensamblador.
- ❑ Este mini computador tendrá un CPU, memoria y registros para poder procesar e interpretar estas instrucciones.

Algunas consideraciones:

- ❑ Cuando carga el programa a memoria, este le debe asignar una posición aleatoria desde la posición 10 hasta el fin de este. Tomar en cuenta el largo el programa, cada línea de programa representará una posición en memoria.
- ❑ El programa debe validar que el archivo esté el formato requerido.

Monoprocesos

Operaciones

001	LOAD
010	STORE
011	MOV
100	SUB
101	ADD



(a) Instruction format



(b) Integer format

Program counter (PC) = Address of instruction

Instruction register (IR) = Instruction being executed

Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory

0010 = Store AC to memory

0101 = Add to AC from memory

(d) Partial list of opcodes

Ejemplo del archivo *.asm

SUPUESTO:

La memoria tiene un largo de 100 entradas [0 al 99]

Formato de la instrucción 8 Bits:

- 0 – 3 Operador
- 4 – 7 Direcccionamiento
 - 4 Registros
 - AX 0001
 - BX 0010
 - CX 0011
 - DX 0100

Formato del número entero, 8 bits

- 0 Signo {0= Positivo | 1= Neg}
- 1 – 7 Valor

CODIGO ASM	COD BINARIO	AC	AX	BX
MOV AX, 5	0011 0001 00000101	0	5	0
MOV BX, 3	0011 0010 00000011	0	5	3
LOAD AX	0001 0001 00000000	5	5	3
ADD BX	0101 0010 00000000	8	5	3
SUB AX	0100 0001 00000000	3	5	3
STORE AX	0010 0001 00000000	3	3	3
MOV BX, -8	0011 0010 10001000	3	3	-8