



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

COMPUTACIÓN CONCURRENTE

NOTAS

Alumno David Pérez Jacome

Profesor: Jorge Luis Ortega Arjona

Febrero 2023

Notas

TEMA 1: INTRODUCCIÓN.

Factores

1. 1 Procesador

- a)* tienen multithreading.
- b)* memoria compartida.

2. N Procesadores

- a)* **Paralelo**(multiprocesadores o multicore).

1) Memoria compartida

- a'* Semaforos
- b'* Región Critica
- c'* Monitores
- d'* Paso de mensajes
- e'* Llamada a Proc. Remoto (RPC)

2) Memoria Distribuida

- b)* **Distribuido**(red de computadoras)

3. Lenguajes de Programación

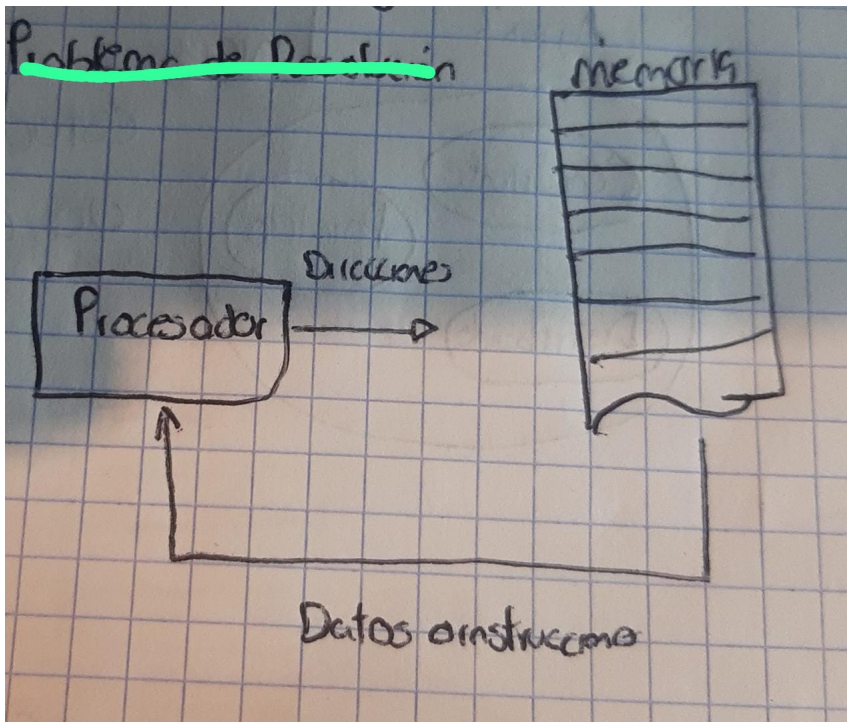
4. Hardware

PROGRAMACIÓN SECUENCIAL VS PROGRAMACIÓN CONCURRENTES

Secuencial	Concurrente
Programa haga lo que deba hacer Programa que se detenga	Controlar el NO-Determinismo Sincronizar procesos.

Factores de desempeño.

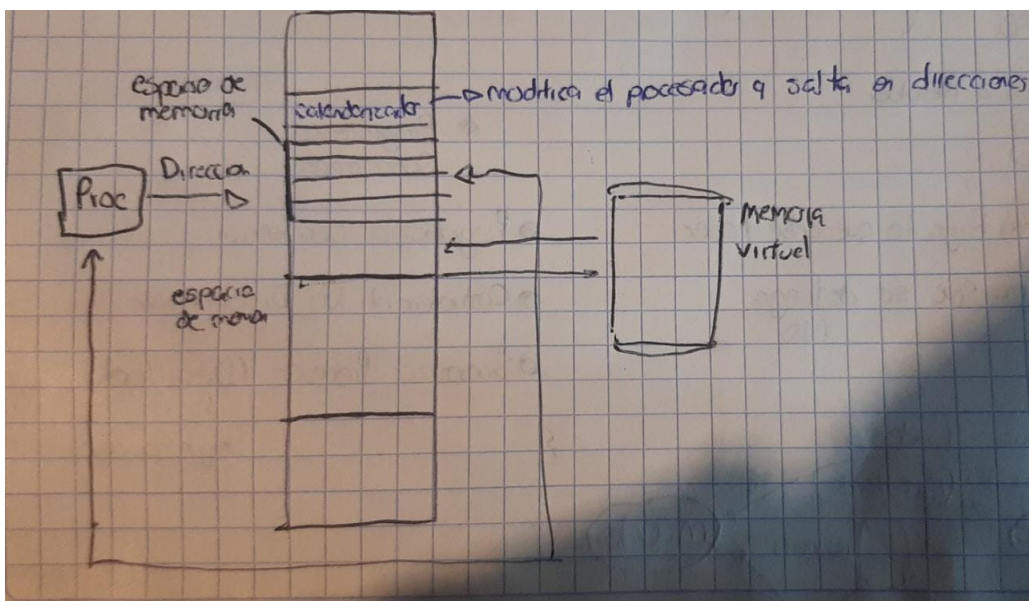
1. Plataforma de Hardware
2. Lenguaje de Programación
3. Problema de Resolución



Instrucciones y datos en memoria codificados.

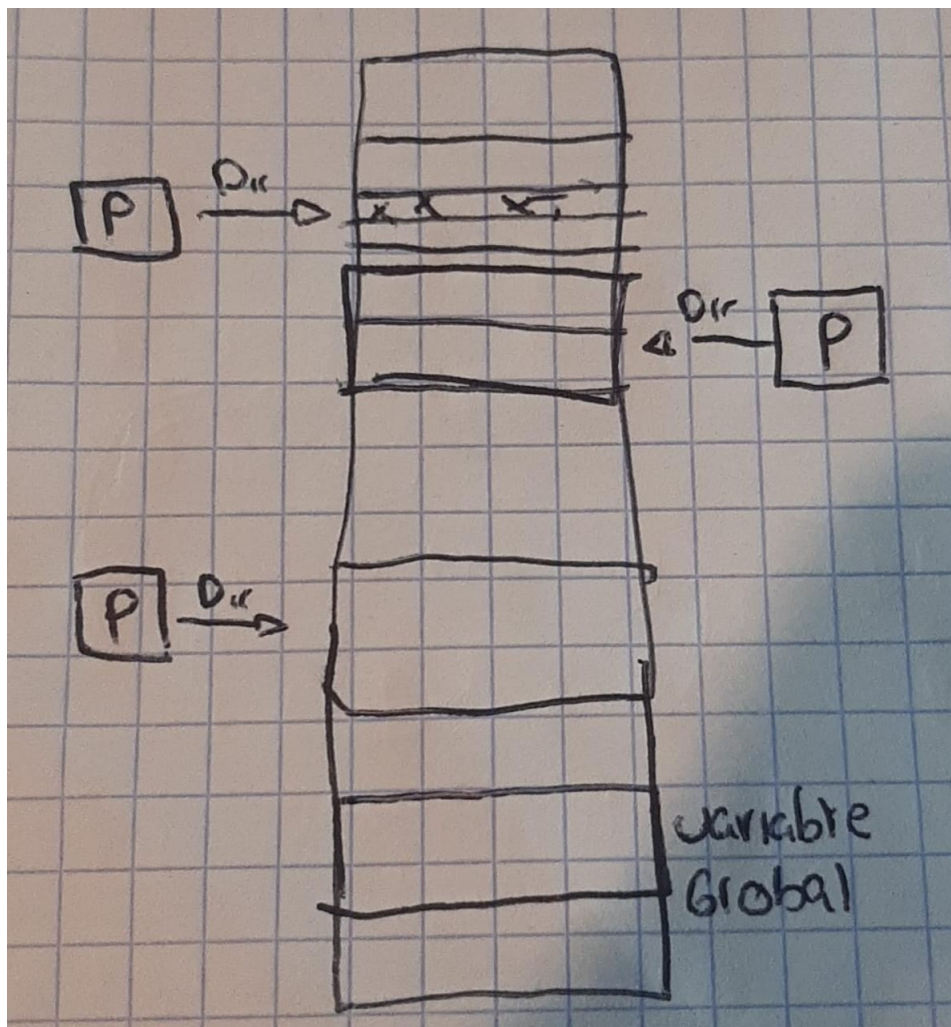
En la ejecución de un programa se siguen varios pasos, como lo son:

1. FETCH: Obtener o buscar las instrucciones.
2. DECODE: De que trata (decodificación).
3. EXECUTE: Ejecución de la instrucción.
4. WRITE: Se escribe en el caso de sincronia para el proceso.



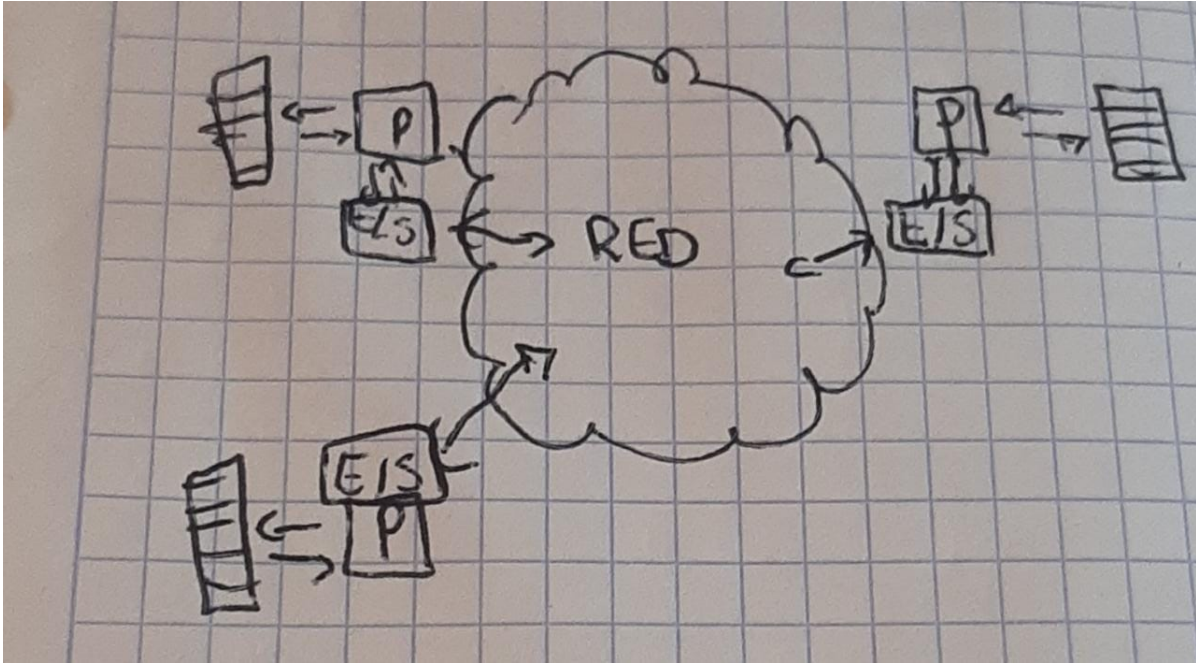
Memoria compartida.

Se encarga de comunicar procesos mediante la variable global (**Variable Compartida**). Todos los procesadores acceden a una memoria global



Memoria Distribuida.

Cada procesador tiene su memoria local intercambiando datos mediante una red de comunicación (E/S)



LENGUAJE DE PROGRAMACIÓN.

1. **Expresar Concurrencia:** OCAN: PAR, — parbegin y parend. *preguntar el lenguaje occan y otro
2. **Expresar Secuencialidad:** $(P_1; P_2;)$ instrucción secuencial, OCCAN: (SEQ)
3. **Expresar Comunicación:** Depende de la organización de la memoria.
 - a) **Compartida:** Variable Compartida.
 - b) **Distribuida:** Llamada a procedimientos Remotos y paso de mensajes `send()`, `recerver()`
4. **Control del NO-Determinismo:** No todos, esto significa que antes de ejecutarlo no sabremos que pasará. Conjunto de estados no se sigue rigurosamente.
Instrucción alternativa de Dijkstra: Instrucción concurrentemente o al mismo tiempo y sin parar.
(threads en java nos dan la concurrencia)

Inclusión de concurrencia en Lenguajes de Programación.

1. Diseño del Lenguaje (Occan).
2. Modificando el lenguaje.(extención en el compilador).
3. Utilizando bibliotecas (libres).

Problema a Resolver.

Programa = Algoritmo + Datos.

(**Capacidad de dividir el algoritmo ó datos para programar en paralelo.**)

Programa Concurrente: Componentes de procesamiento + Componentes de comunicación.

Conceptos y Terminología.

Proceso. Es el cambio en el **estado** de la memoria por acción del procesador. (valor instantaneo de las variables de un sistema.)

Programa. Es la especificación de uno o varios procesos. (ya sea **secuencial** o **concurrente**.)

1. **Programación Secuencial.** Especificación de un proceso.
2. **Programación Concurrente.** Especificación de varios procesos.
Conjunto de de procesos secuenciales que se ejecutan simultaneamente, comunican entre si por un objetivo en común.
 - a) **Programa Multithread.**
 - b) **Programa Paralelo.**
 - 1) Memoria compartida: comunicación por memoria.
 - 2) Memoria Distribuida: Reedes Compartidas.
 - c) **Programas Distribuidos.**

Cnceptos de SW	Cnceptos de HW
Proceso	Procesador
Comunicación(variable compartida (global. Paso de mensajes y llamadas a procedimientos remotos))	Memoria (Distrib y Compartida).

*(procesador accede a la memoria en nano segundos)

Los factores son: plataforma de HW, lenguajes de programación y el problema a resolver.

Coordinación:

Establecer puntos de acciones en tiempo y espacio.

Organizar las tareas de tal forma que esten en tiempo y espacio y no tarden en comunicarse. (termina en tiempo)

Comunicación.

Intercambio de mensajes.

Sincronización.

Protocolo para intercambio de forma ordenada.

Acciones para realizar la comunicación de forma exitosa. Para ello necesitamos la sincronización de las formas de comunicación.

Granularidad..

Que tanto es o se puede dividir algo. se usa la formula: $G = \frac{\text{tiempoProcesamiento}}{\text{tiempoComunicacion}}$

1. **Gruesa:** (acceso controlado a BD, uno y uno). $G = \frac{\text{mas}}{\text{menos}}$
2. **Media:** $G = \frac{\text{normal}}{\text{normal}}$
3. **Fina:** (comunicación entre procesadores). $G = \frac{\text{menos}}{\text{mas}}$

TEMA 2: PROCESO SECUENCIAL.

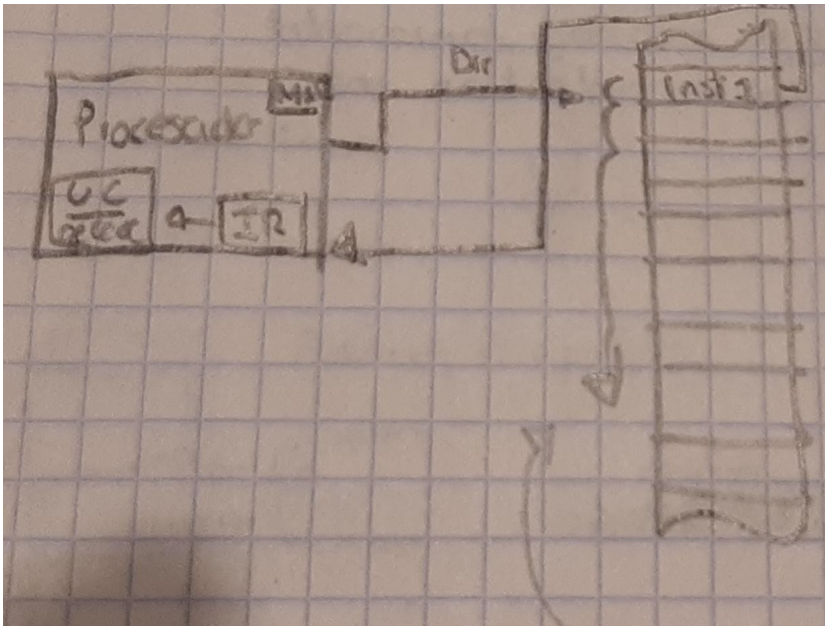
Proceso Secuencial.

Es el cambio de estado en la memoria por acción del procesador. Una instrucción a la vez ya que solo tiene un procesador.

Arquitectura Von Neumman.

1. Instrucciones y datos.
 - a) Almacenados en memoria.
 - b) Codificados (binary).
2. Thread (hilo de control): Como una hebra de control y cada instrucción es un caso al que se le agrega.

Programa con un solo thread es proceso secuencial.

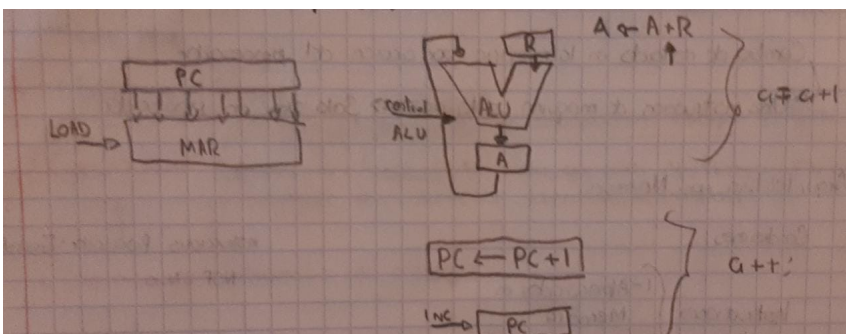


Todo lo siguiente se ejecuta en un solo hilo o thread:

1. FETCH(buscar)
2. DECODE(comunicar)
3. EXECUTE(ejecutar):
 - a) SW a HW.
 - b) De instrucción lógica a señal física.
4. WRITE(resultado): Determina la dirección de la siguiente instrucción.

Lógica de transferencia de registros.

$MAR \leftarrow PC$ (microoperación)



PC → Entre direcciones

1. FETCH

a) $t_0 : MAR \leftarrow PC$

- b) $t_1 : MBR \leftarrow M, PC \leftarrow PC + 1$
- c) $t_2 : IR \leftarrow MBR$
- d) NOP $q_0 t_3 : T \leftarrow \emptyset$
- e) MOVR $q_1 t_3 : A \leftarrow R, T \leftarrow \emptyset$
- f) LDI(Data8) $q_2 t_3 : MAR \leftarrow PC$
- g) $q_0 t_4 : MBR \leftarrow M, PC \leftarrow PC + 1$

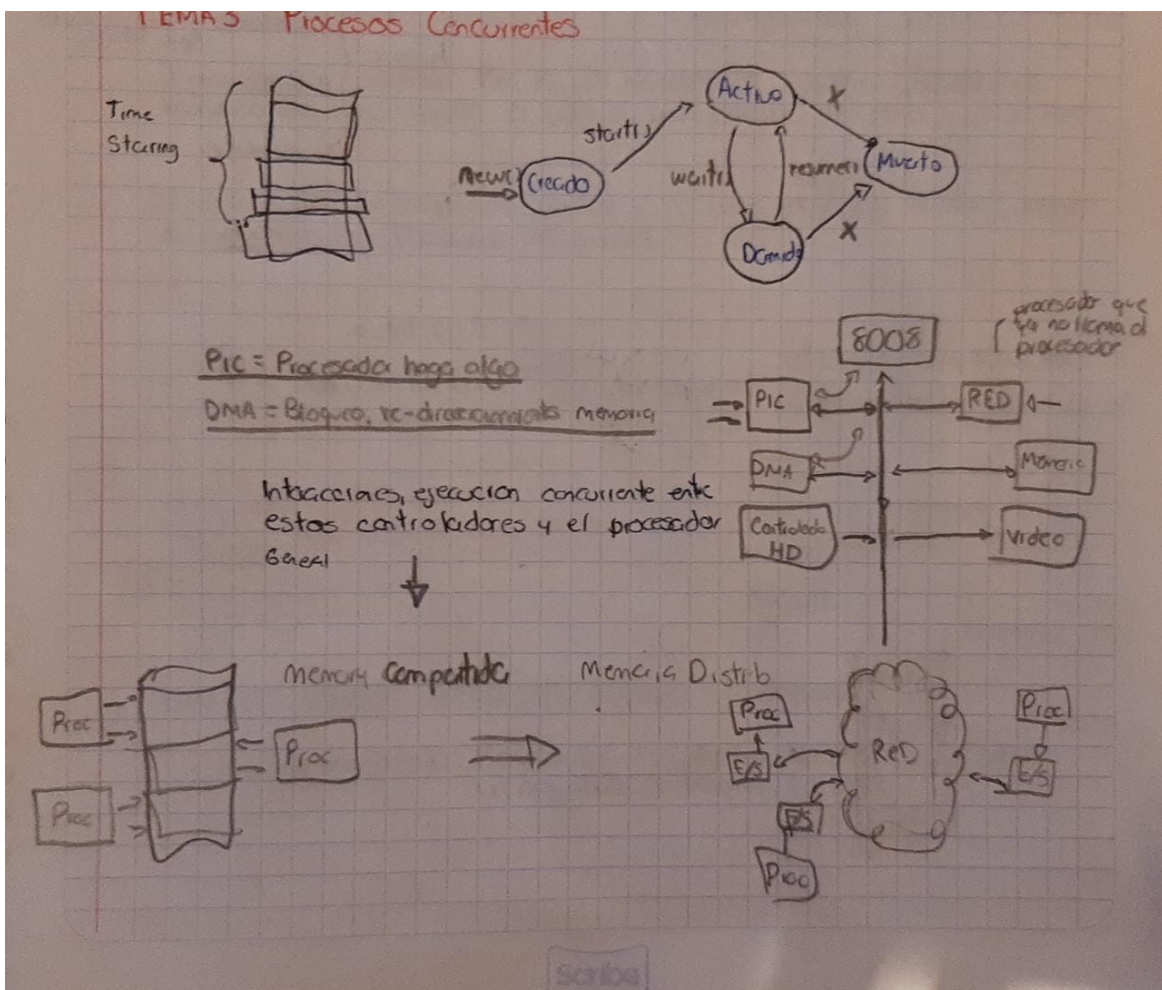
Instrucciones:

Autocontención.

CARACTERISTICAS DEL PROCESO SECUENCIAL:

1. Que el programa haga lo que debe hacer.
2. El proceso termine.

TEMA 3: PROCESOS CONCURRENTES.



Lenguaje Programación.

Expresión de comunicación (organiza la memoria.)

1. Memoria Compartida: Usa Variable compartida.

Co-operating Sequenlid E.W.D.

1. **Contexto:** Memoria compartida. **Procesos concurrentes comunicandose por variables compartidas**
2. **Problema:** Variables compartidas tienen un valor \rightarrow **Integridad con exclusión mutua**
3. **Solución:** Semaforo \rightarrow **Exclusión mutua** \rightarrow **Sección critica**
Se mezclan los procesos P_a, P_b al momento de ejecutar **FDXR** al momento que no escribe el valor de una variable, el otro igual lo hace.

* El valor de la variable compartida cambia y no tiene integridad y hace perdida de actualizaciones. (**Procesador secuencial**).

Sección Critica: Parte del codigo que modifica la variable compartida.

```
1.                int a = 1;
2.  PA for(for 20 veces):      PB for(for 20 veces):
3.      a++                      a++
4.
5.  //El que lee, ¿qué lee?
6.  //El que escribe, ¿qué escribe?
7.  // esto cuando
```

En un cruce, en donde se cruzan los autos por decir, los autos son los procesos y la intercepción es la **Sección critica**.

Semaforo: Variable entera NO-Negativa.

1. **P:** Disminuye el valory en caso bloquea.
2. **V:** Aumenta el valor.

P:(Operación bloqueante): Proceso que lo invoque y si da negativo, este se bloquea y no sigue.

Sincronización en Variable Compartida.

No anidar seccion critica.

1. Semaforo → Bloqueante.
2. Algoritmo de Decker → **Espera activa**.
3. Region Critica
4. Monitor.

ESPERA ACTIVA: En el caso del algoritmo de becker el c_2 si no cambia y sigue esperando, regresa y sigue sin cambiar, es como si diera vueltas en un carrusel, por tanto es dar vueltas y esperar y esperar y esperar.
 Consume tiempo del procesador (**Polling**).

Productor-Consumidor.

