

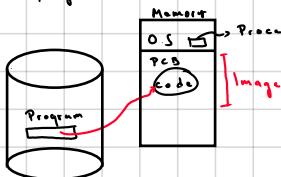
Program code → class, pgs... etc

set of Data:

- Associated to Program Code

#### Process control block

El programa lo sube a memoria y separa espacio de memoria, pone un encabezado, etc luego



Que hay en PCB:

- Identifiers
- State
- Priority
- Program Counter → Que hay en el procesador
- Memory Pointers → Que archivos tengo abiertos Generan una imagen del procesador
- Processor Context Data
- I/O status Information
- Accounting Information → bytes/recursos del proceso

#### Dispatcher → zona de procesador que permite hacer cambios de proceso.

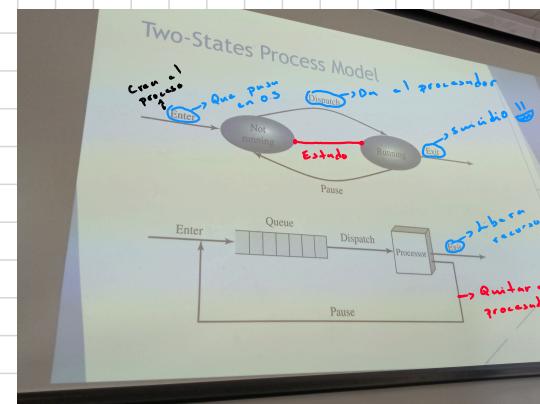
→ Debe ser muy rápido, si no, genera cuello de botella

↳ Escrito en Assembly, MUY optimizado

#### Corazón del sistema operativo

#### Process States:

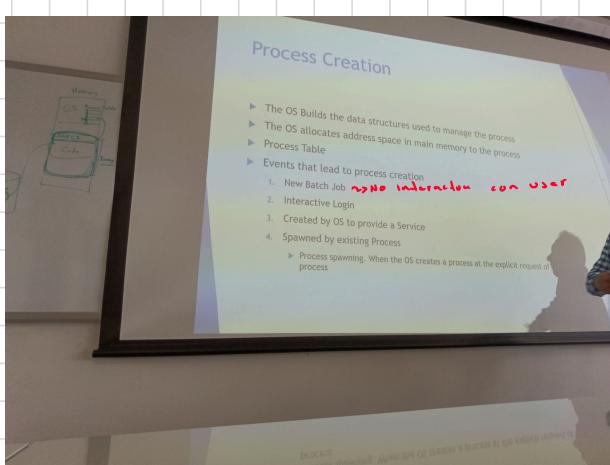
- Describe el comportamiento del proceso.
- Situation del proceso (Acabó, inicio, corriendo...)
- Ciclo de vida del proceso:
  - De creación a terminación



- DAS más complicado de SO: Un arreglo FIJO

→ Ningún SO usa esto, solo es para ilustrar

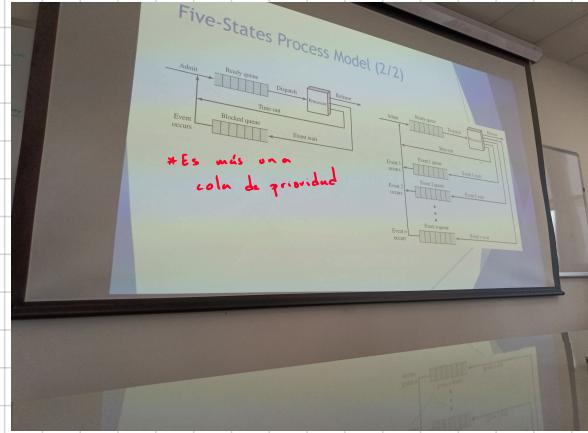
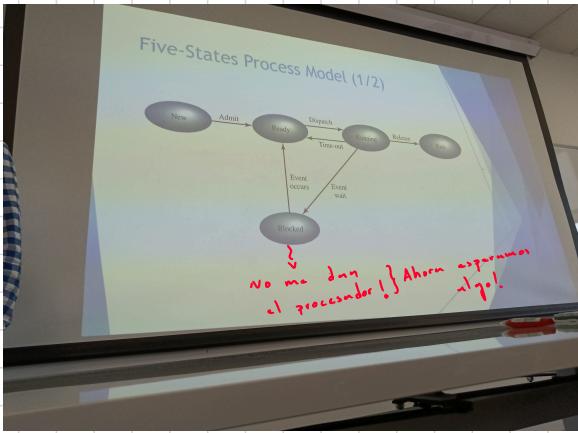
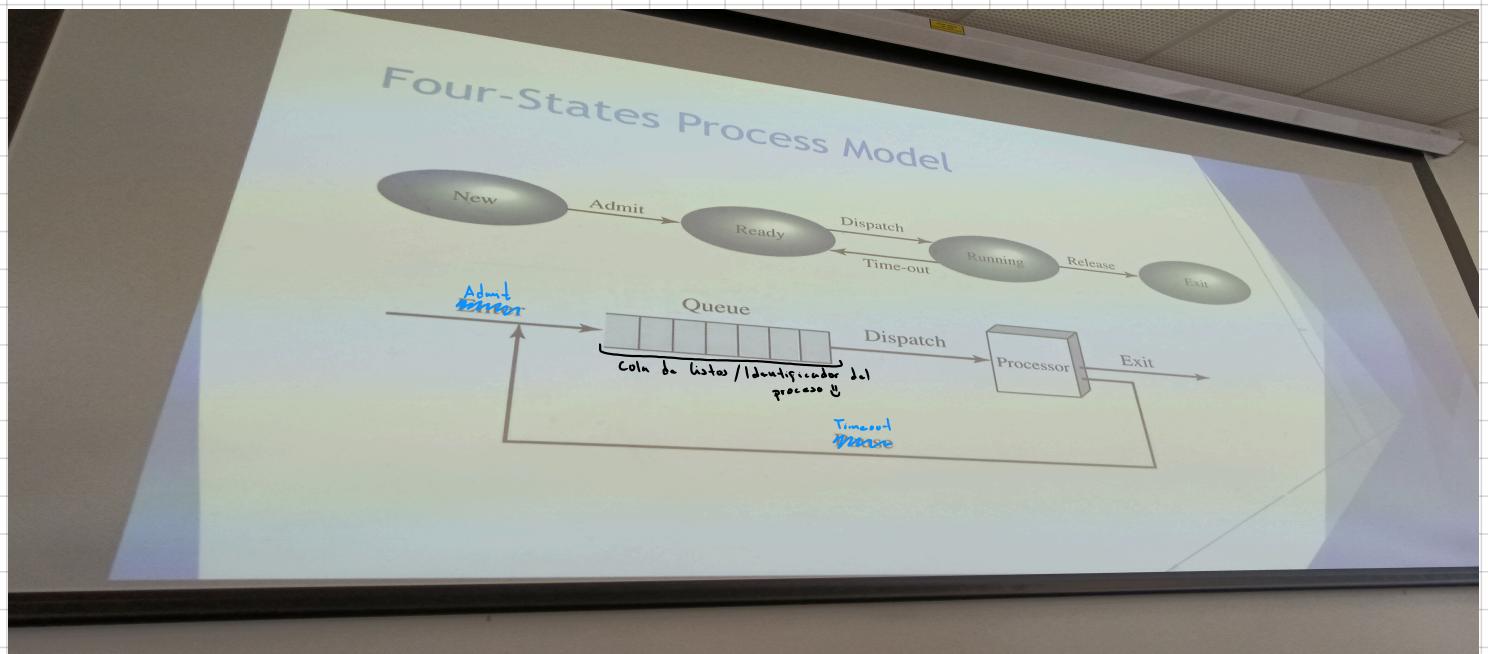
- Cuando creo un proceso, al final de la cola



#### Terminación de un proceso:

- Soltar recursos
- Eventos que llevan a terminar:
  - Normal termination
  - Time Limit Exceeded
  - Memory unavailable
  - Bounds violation
  - Production error
  - Arithmetic error
  - Time overrun
  - I/O failure
  - Invalid Instruction
  - Privileged Instruction
  - Data misuse
  - Operator or OS intervention
- Parent Termination } Proceso padre -> hijo } No en todos
- Parent Request } El hijo debe acabar antes } En UNIX no hay herederos XD

## Primer und dato implementando



## I/O Queue - Event Queue

- Resource requested
- Process Requesting
- Request status

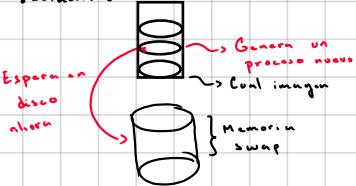
• Planificación (Scheduler) → Se encarga de la coda de eventos

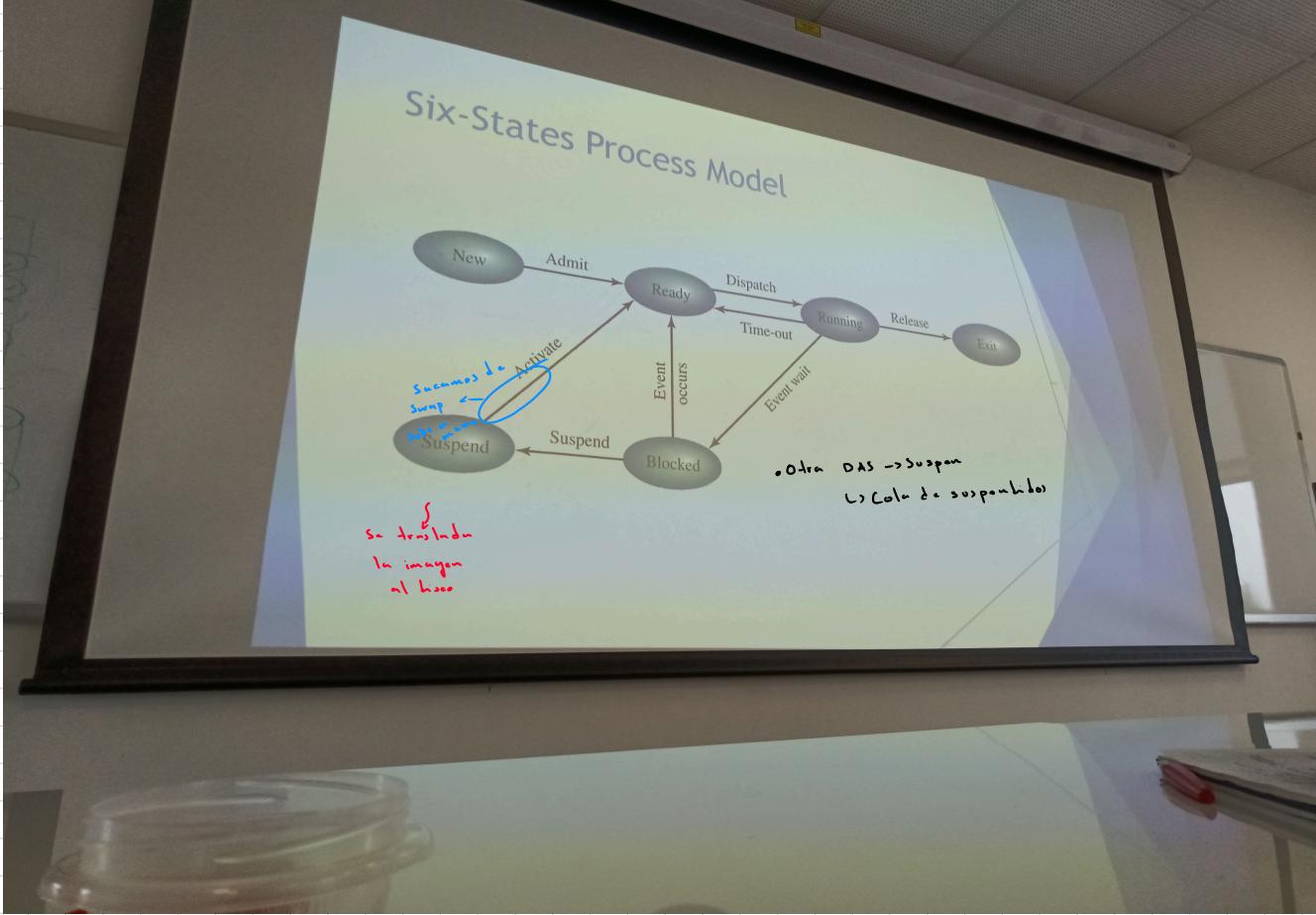


• En 5 estados → Ociooso al procesador.

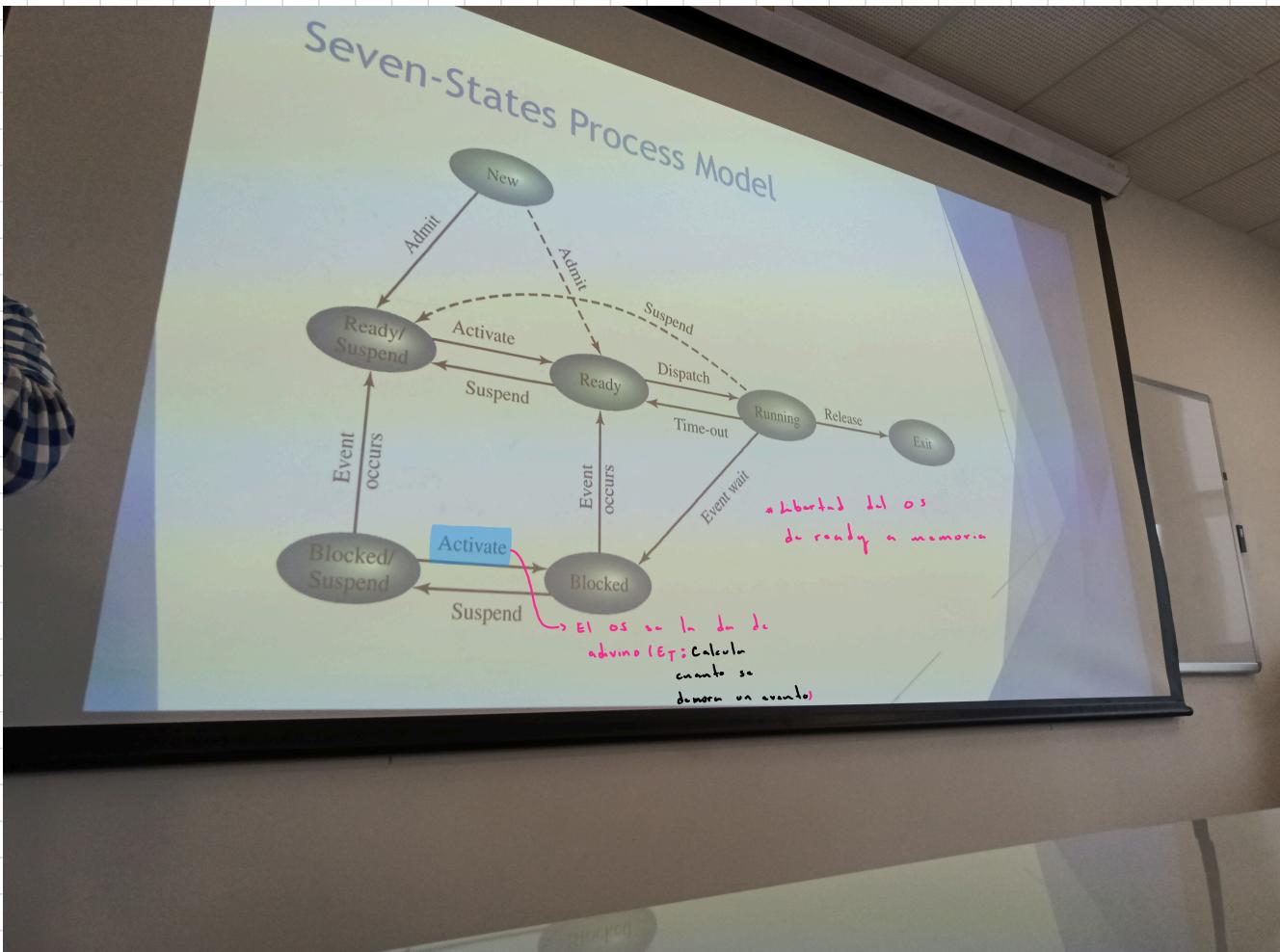
↳ No le quedó dar un proceso al procesador.

• Solución:





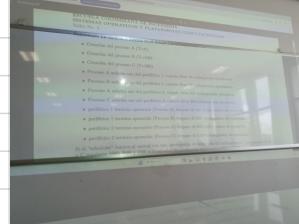
- Bien en programador on/off
- No funciona bien para el user



## Ejercicios - taller nro. 11

• Reloj de pared  $\rightarrow$  tiempo de ocurrencia

- Creación proceso A ( $T=0$ )
  - " " proceso B ( $T=100$ )
  - " " proceso C ( $T=200$ )



- Tiempo en el sistema operativo.
  - Tiempo en estado listo.
  - Tiempo en estado bloqueado.
  - Tiempo en la cola de eventos esperando por un periférico.
  - Tiempo en que realiza peticiones a periféricos.

2. Usando procesos de 6 estados, si ademas, un proceso se suspende él lleva un (1) nullificador bloqueo.

especifique cada proceso

  - Tiempo total en el sistema operativo.
  - Tiempo en estado listo.
  - Tiempo en estado bloqueado.
  - Tiempo en estado suspendido.
  - Tiempo en la cola de eventos esperando por un periférico.
  - Tiempo en que realiza peticiones a periféricos.

• Tiempo en 4-  
so suspende si lleva un (1) milsegundo bloq

