

# 18/03/2025 - Sistemas Operativos - Ude@

## Avisos

1. Novedad: Jueves: Laboratorio (Danny)  
Viernes: Teoría (Henry) (Solo por esta semana).

2. Red Hat Academy (Opcional)

- Administración de sistemas (terminal Linux) ☐
- • Kubernetes [\*] ✓
- • Podman [\*\*] ✓

3. Github

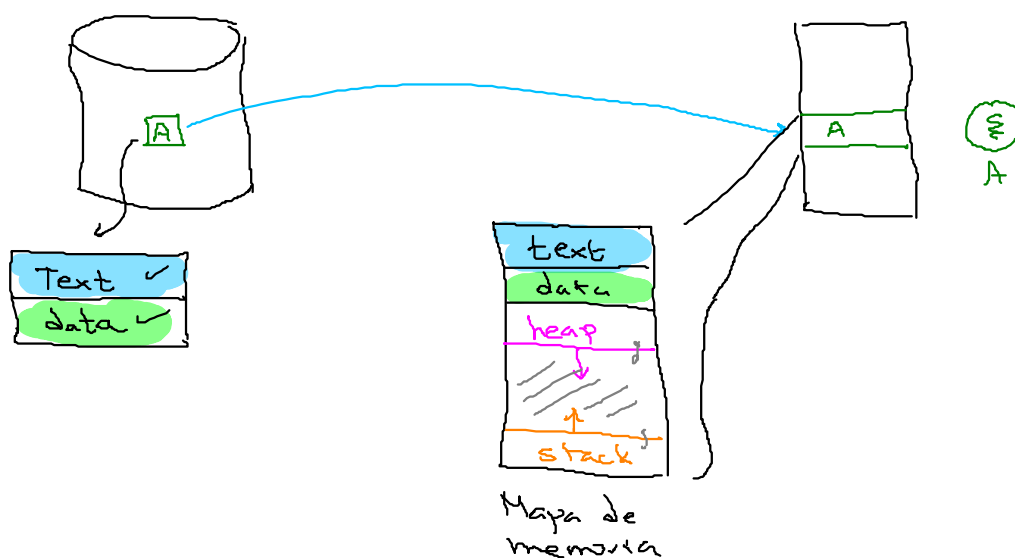
4. Apuntes

## Clase anterior.

1. Programa

.vs.

Proceso

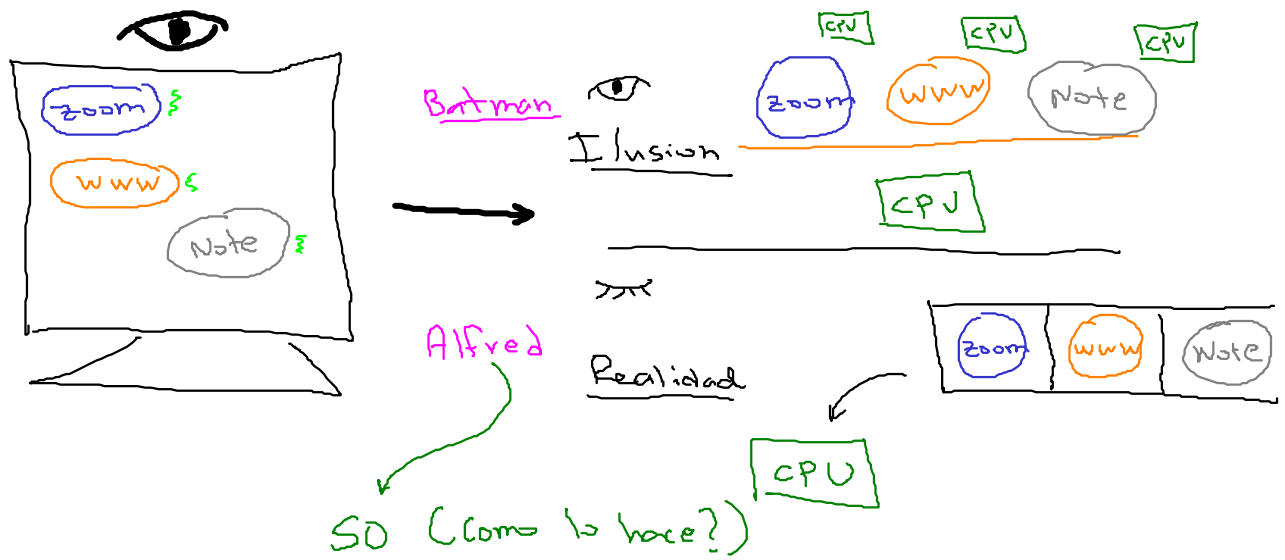


2. Proceso → Abstracción

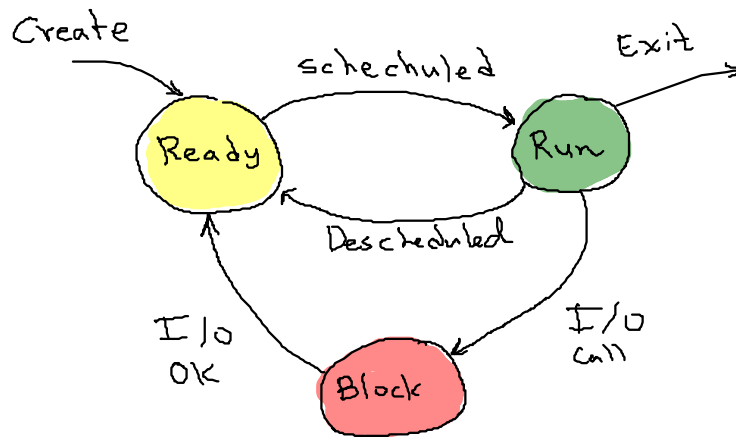


Proceso = CPU + Mem + I/O

Como es posible que con una sola CPU se puedan (aparentemente) correr varios programas?



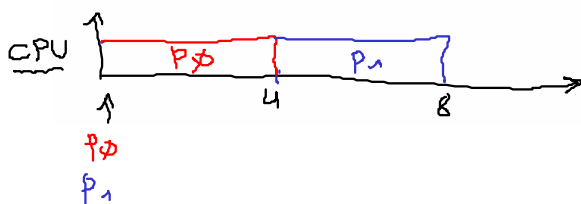
### 3. Model de 3 estados



Caso 1:  $P_0$  y  $P_1$  Sin I/O

Time	Process <sub>0</sub>	Process <sub>1</sub>	Notes
1	Running	Ready	
2	Running	Ready	
3	Running	Ready	
4	Running	Ready	Process <sub>0</sub> now done
5	Exit	Running	
6	-	Running	
7	-	Running	
8	-	Running	Process <sub>1</sub> now done

Tracing Process State: CPU Only



El simulador **process-run.py** y las instrucciones de uso se encuentran en el siguiente [link](#)

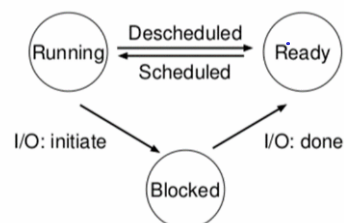
→ 0-100 (Uso de CPU)

```

./process-run.py -l 4:100,4:100 -p -c
  
```

Time	PID: 0	PID: 1	CPU	IOs
1	RUN: cpu	READY	1	
2	RUN: cpu	READY	1	
3	RUN: cpu	READY	1	
4	RUN: cpu	READY	1	
5	DONE	RUN: cpu	1	
6	DONE	RUN: cpu	1	
7	DONE	RUN: cpu	1	
8	DONE	RUN: cpu	1	

Stats: Total Time 8  
 Stats: CPU Busy 8 (100.00%)  
 Stats: IO Busy 0 (0.00%)



El simulador **process-run.py** y las instrucciones de uso se encuentran en el siguiente [link](#)

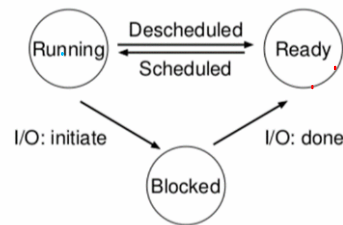
Time	Process <sub>0</sub>	Process <sub>1</sub>	Notes
1	Running	Ready	
2	Running	Ready	
3	Running	Ready	
4	Blocked	Running	Process <sub>0</sub> initiates I/O
5	Blocked	Running	so Process <sub>1</sub> runs
6	Blocked	Running	
7	Ready	Running	I/O done
8	Ready	Running	Process <sub>1</sub> now done
9	Running	-	
10	Running	-	Process <sub>0</sub> now done

Tracing Process State: CPU and I/O

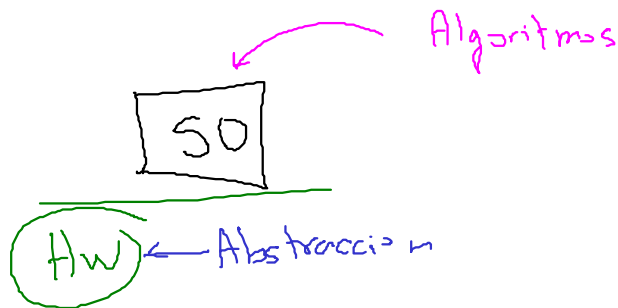
```
./process-run.py -s 1 -l 6:80,5:100 -L 3 -c -p
```

Time	PID: 0	PID: 1	CPU	IOs
1	RUN:cpu	READY	1	
2	RUN:io	READY	1	
3	BLOCKED	RUN:cpu	1	1
4	BLOCKED	RUN:cpu	1	1
5	BLOCKED	RUN:cpu	1	1
6*	READY	RUN:cpu	1	
7	READY	RUN:cpu	1	
8	RUN:io_done	DONE	1	
9	RUN:cpu	DONE	1	
10	RUN:cpu	DONE	1	
11	RUN:cpu	DONE	1	
12	RUN:cpu	DONE	1	

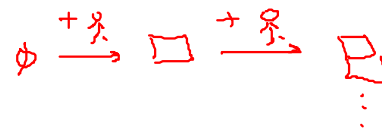
Stats: Total Time 12  
Stats: CPU Busy 12 (100.00%)  
Stats: IO Busy 3 (25.00%)



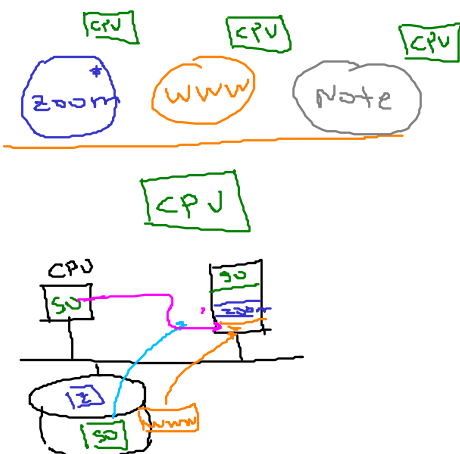
#### 4. Estructuras de datos



Agenda



Recordemos



Que estructuras de programación usamos

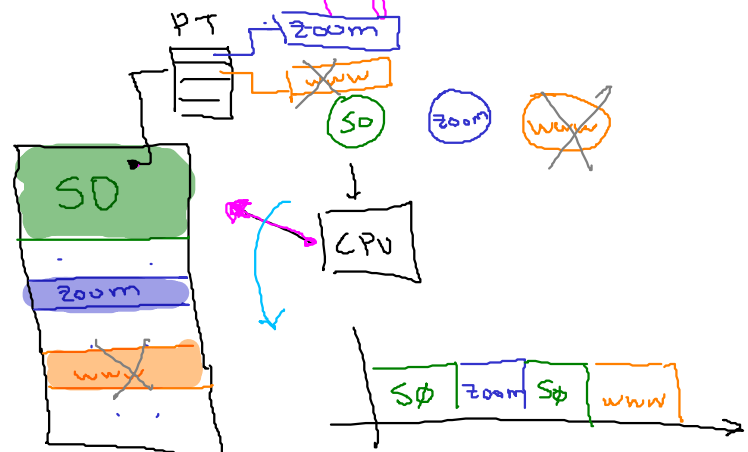
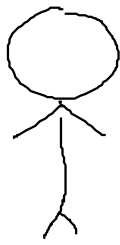


Tabla de proceso

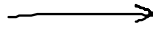
Requisitos



Modelo



QPlan



code

(c)

Cual es el codigo?

1. Lista procesos

2. PCB