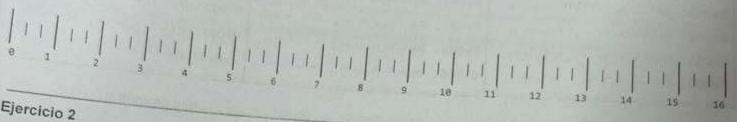
Sistemas Operativos, Final 5/7, vi

Apellidos:	5/7, viernes 📾 4 julio 🕾 2025			
2:	Nombres:			
Ejercicio 1	4:	5:	6:	

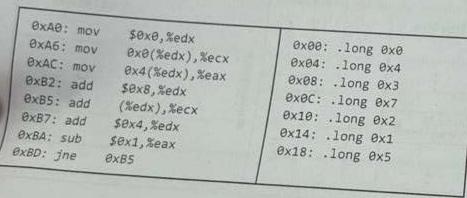
Planificar con Round Robin Q=2 para los siguientes procesos que tienen mezcla entre cómputo CPU y espera I/O. Suponga que hay 3 (tres) núcleos de ejecución, y es por eso que cada celda de tiempo está subdividida en tres partes.

A Proceso	Inicio	CPU	200	cion, y
В	0	010	10	CPU
C		3	6	
		4	3	
	6	10		-



# Ejercicio 2

Mostrar la secuencia de accesos a la memoria virtual (traza) que se produce al ejecutar este programa assembler x86\_32 que comienza en la dirección de memoria 0хA0. La versión original de este código salió en el número <u>abril de 1967 de</u> Cosmopolitan.





### Traza:

 $\theta x \leftarrow L$ ,  $\theta x \vdash L$  $\theta x \leftarrow L$ ,  $\theta x \leftarrow L$ ,

# Ejercicio 3

Tenemos el viejo esquema i386 de paginación (10,10,12)→(20,12).
Supongamos que tenemos el registro de paginación apuntando al marco físico CR3=0x0C0CA.

CR3 }

# 0x0C0CA dir : marco fís. , X W R V 0x3FF: 0x----, --- 0x003: 0x----, --- 0x002: 0x----, --- 0x001: 0x----, ----

## 0x0C01A

------

dir : marco fís. , X W R V

0x3FF: 0x----, ----

0x003: 0x----, ----

0x002: 0x----, ----

0x000: 0x----, ----

### **OXOCAFE**

dir : marco fís. , X W R V

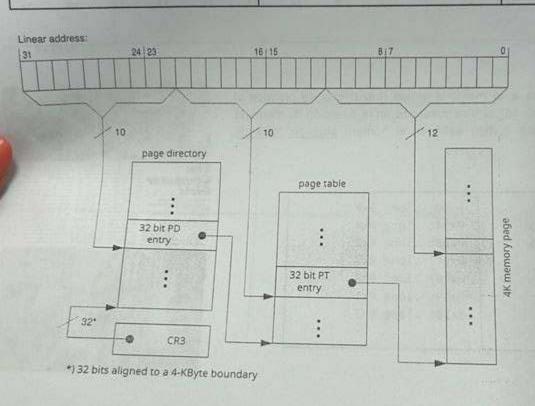
0x3FF: 0x----, ----

0x003: 0x----, ----

0x002: 0x----, ----

0x001: 0x----, ----

0x000: 0x----, ----



### Bits de control

V: válido

R: se puede leer, readable

W: se puede escribir, writable

X: se puede ejecutar, executable

a) Completar el PD y la PT para que mapee las direcciones virtuales a físicas como sigue.

0x00000BAD -> 0x0060DBAD

0xC0000FEE -> PF

b) Continuando el mapeo, hacer que además mapee de virtual a física:

0xFFFFF000 -> 0x0C0CA000

0xFFC00000 -> 0x0CAFE000

# Ejercicio 4

Para el siguiente multiprograma tomar atomicidad línea a línea y « es el operador de desplazamiento lógico a la izquierda.

	s=0	Constitute of the second
while(s<32) {	while(s<32) {	while(s<32) {
S = S + 1 ;	s = s * 2;	s = s << 3
)	)	1
	s=?	

a) Dar todos los valores posibles de s al terminar el multiprograma, ordenados de menor a mayor.

b) Agregar semáforos y su inicialización (dentro del mismo programa de arriba), para que siempre de el valor MÍNIMO.

### Ejercicio 5

a) Escribir la solución a la región crítica usando Semáforos.

while(1) {	while(1) {
CS0	CS1

b) Escribir la solución a la región crítica usando Peterson.

white/ex c	turn=0 && !flag0 && !flag1	
while(1) {	while(1) {	
CS0		
	CS1	
	}	

Ejercicio 6

Complete el diagrama de accesos a las diferentes partes de las estructuras de datos en disco (horizontal) para una secuenci
de syscalis que recibe el sistema operativo (vertical, el tiempo avanza hacia abajo).

Deberá colocar read o write en cada lugar que corresponda. Explique brevisimamente cada uno.

	data bitmap	inode bitmap	root	foo inode	bar inode	root data	foo data	bar data	bar data	bar data
create (/foo/bar)								[v]	[1]	[2]
write()	Marie and	000	dis pury	un aman						
write()				MET MAN		to talking			Market St.	Parking Street
write()										Apple Mark Mark Mark Mark Mark Mark Mark Mark
write()										120