

Sistemas Operativos

Cursada 2022

Comisión S21 y S22

Planificación de la CPU

Cuando nace la multiprogramación, lo que vamos a tener es el:

Uso compartido de la CPU entre múltiples procesos/Hilos

Dos tipos de procesos:

- **Orientados a la CPU**
- **Orientados a la E/S**
- **Intermedios** (sobre dispositivos rápidos)

Planificación de la CPU

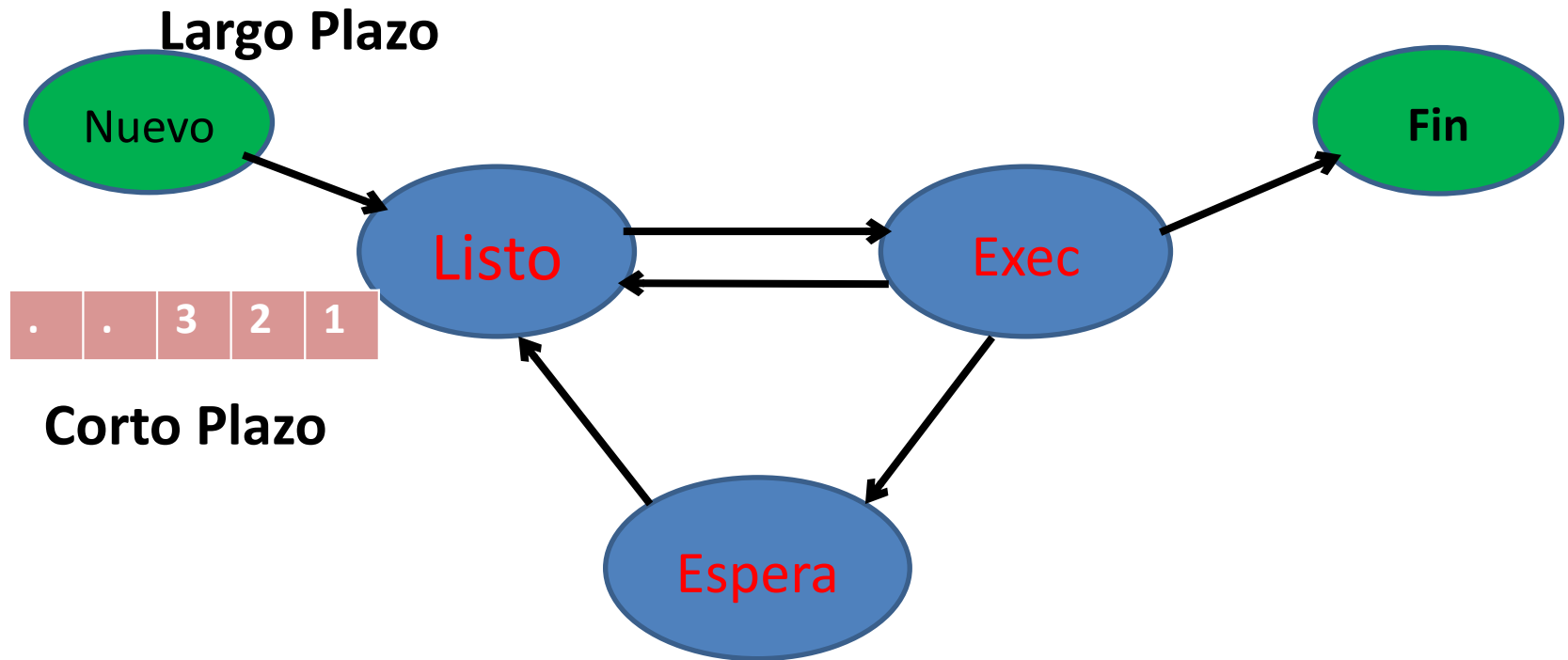
Procesos/Hilos

- **Orientados a CPU – Ráfagas largas**
- **Orientados a la E/S – Ráfagas cortas**

Otro objetivo de la multiprogramación era Lograr el máximo uso de la CPU y para esto tenemos que ?

Planificar (buscar un equilibrio)

El planificador de largo plazo trataba de buscar ese equilibrio



Planificación de la CPU

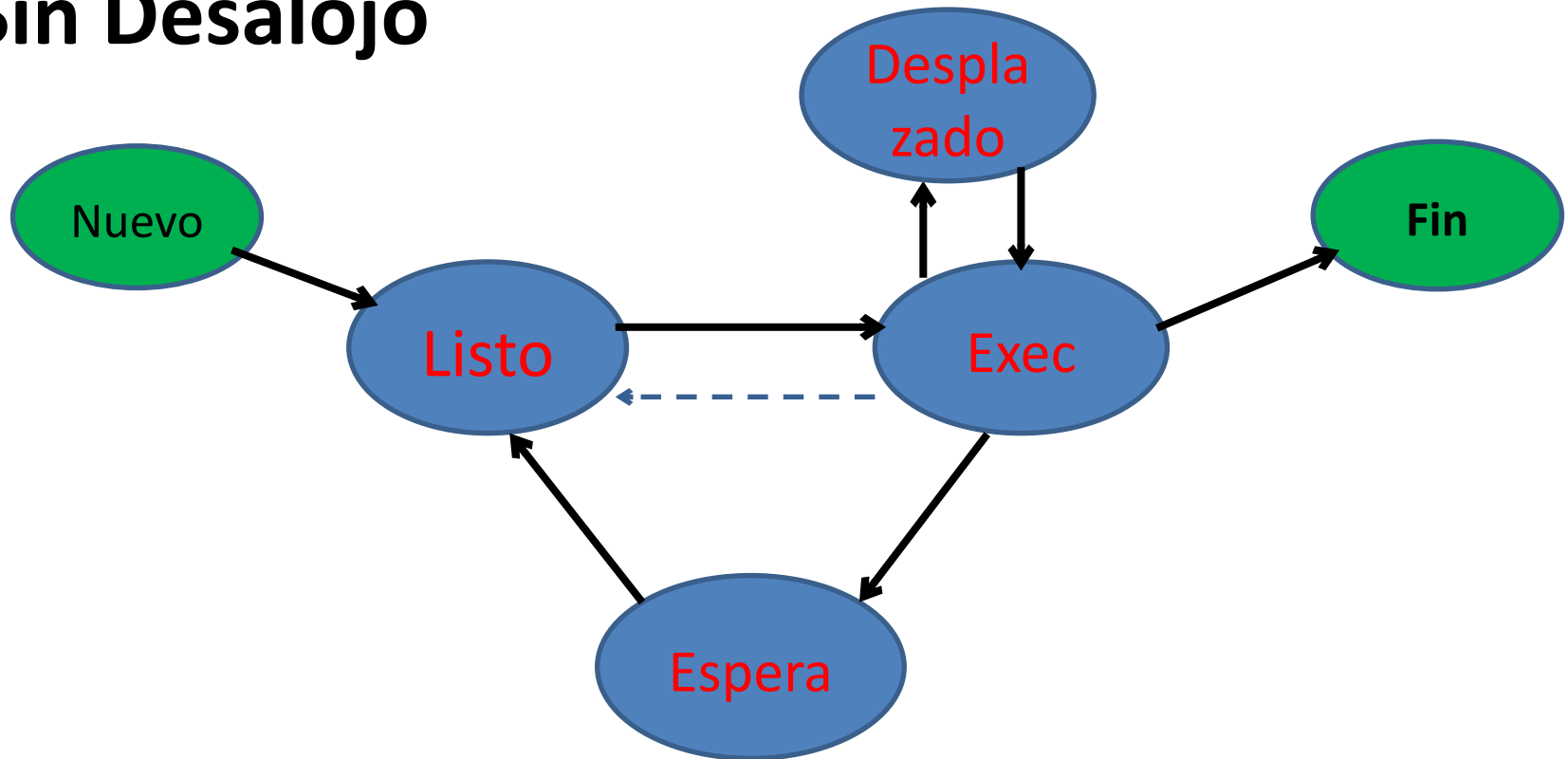
- **EL planificador de corto plazo va a trabajar sobre un conjunto (No lista) de procesos listos.**
- **Teniendo en cuenta que la CPU trabaja de a ráfagas**
- **Vamos a tener varios algoritmos de planificación de la CPU**
- **Distintos criterios (depende en que equipos y organización estemos)**

Planificación de la CPU

Otro concepto a tener en cuenta en los algoritmos de planificación es

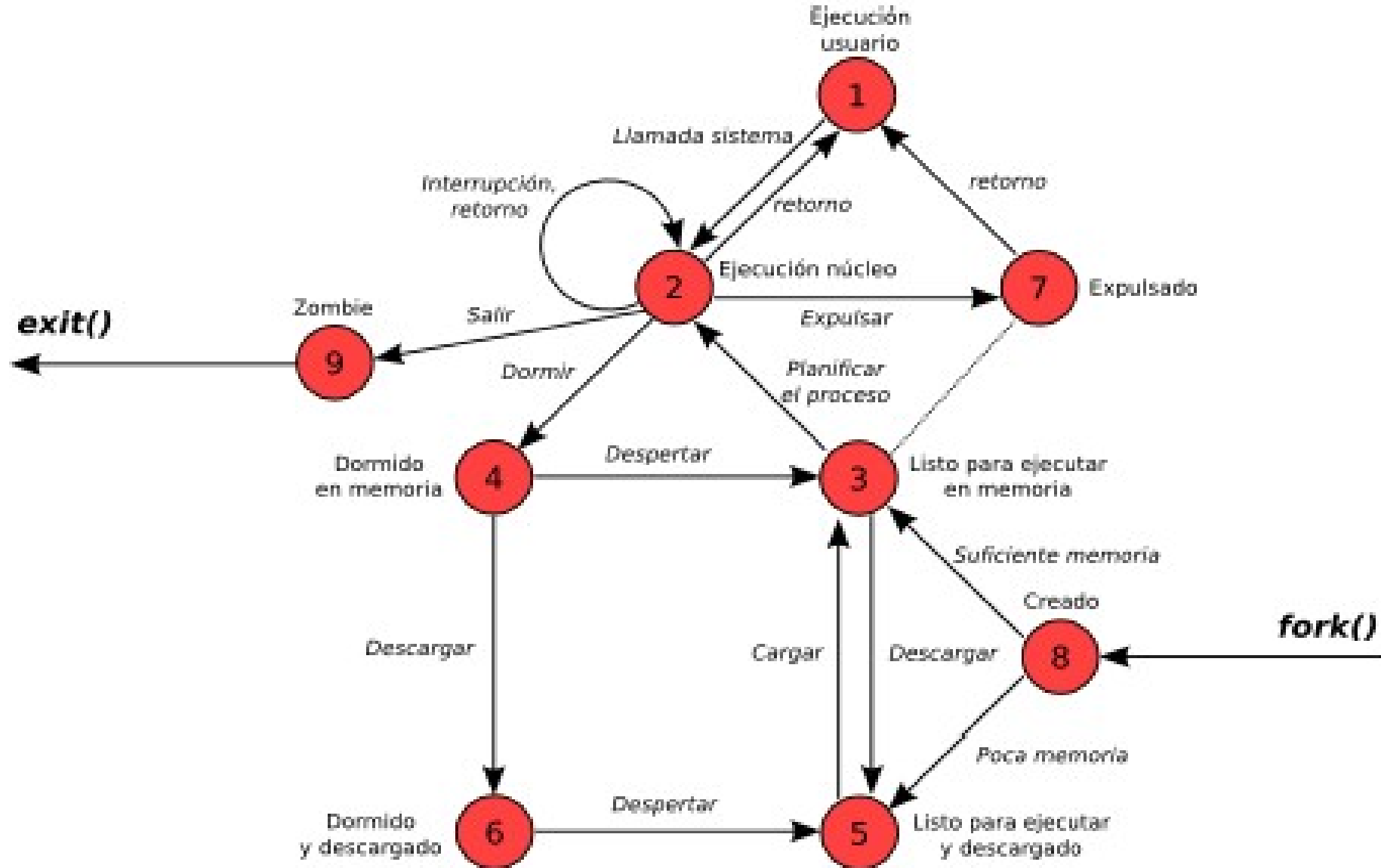
➤ **Con desalojo**

➤ **Sin Desalojo**



Planificación de la CPU

Diagrama de 9 estados de Unix



Planificación de la CPU

Algoritmos de planificación

- **FCFS (First – Come – First - Served) un FIFO**
- **SJF (Shortest – Job – First)**
- **SRTF (Short – Remaining – Time - First)**
- **Prioridad**
- **RR (Round – Robin)**
- **RR (Round – Robin) - Variante**
- **Multicola**

Algoritmo FCFS

Ejemplo FCFS (nonpreemptive)

Job	Instante llegada	Tiempo de CPU	Instante E/S	Tiempo E/S
A	0	4		
B	3	2		
C	3	2	1	2
D	5	3		

Penaliza ciertos tipos de procesos

FCFS - No Apropiativo

Reloj	Listo	Ejecucion	Espera	End	Comentario
0	A	SO			CS
1		A			
2		A			
3	ABC	SO			CS
4	BC	A			
5	ABCD	SO			CS
6	BCD	A			
7	BCD	SO		A	CS
8	CD	B			
9	CD	B			
10	CD	SO		AB	CS
11	D	C			
12	D	SO	C		CS
13	D	D	C		
14	DC	SO			CS
15	C	D		ABC	
16	C	D			
17	C	SO		ABD	CS
18		C		ABD	
19		SO		ABDC	CS

Algoritmo SJF

Ejemplo SJF (nonpreemptive)

$$T_{(n+1)} = \alpha \cdot t_0 + (1 - \alpha)t_n$$

Job	Instante llegada	Tiempo de CPU	Instante E/S	Tiempo E/S
A	0	4		
B	2	3		
C	2	2	1	2
D	5	2		

SJF - No Apropiativo

Reloj	Listo	Ejecucion	Espera	End	Comentario
0	A	SO			CS
1		A			
2	ACB	SO			CS
3	CB	A			
4	CB	A			
5	ACDB	SO			CS
6	CDB	A			
7	CDB	SO		A	CS
8	DB	C			
9	DB	SO	C		CS x E/S
10	B	D	C		
11	CDB	SO			CS
12	DB	C			
13	DB	SO		AC	CS
14	B	D			
15	B	D			
16	B	SO		ACD	CS
17		B			
18		B			
19		B			
20		SO		ACDB	CS

Algoritmo SRTF

Ejemplo SRTF (preemptive)

También llamado SJF Apropiativo

Job	Instante de llegada	Tiempo de CPU
A	0	7
B	3	6
C	5	3
D	7	5
E	12	3

SRTF - Apropiativo

Reloj	Listo	Ejecucion	Espera e/S	End	Comentario
0	A	SO			CS
1		A			
2		A			
3	AB	SO			CS
4	B	A			
5	CAB	SO			CS
6	AB	C			
7	CADB	SO			CS
8		C			
9		C			
10	ADB	SO		C	CS
11	DB	A			
12	AEDB	SO			CS
13	EDB	A			
14	EDB	A			
15	EDB	A			
16	EDB	SO		CA	CS
17	DB	E			
18	DB	E			
19	DB	E			
20	DB	SO		CAE	CS
21	B	D			

Algoritmo de Prioridades

Ejemplo algoritmo Prioridades (preemptive)

Job	Hora de Llegada	Tiempo de CPU	Prioridad
A	0	3	3
B	3	4	1
C	3	2	3
D	5	4	4
E	7	2	2

Prioridades iguales (inconveniente)

Starvation o muerte por inanición (inconveniente)

Limite al numero de atención (solución)

Prioridades relativas (solución)

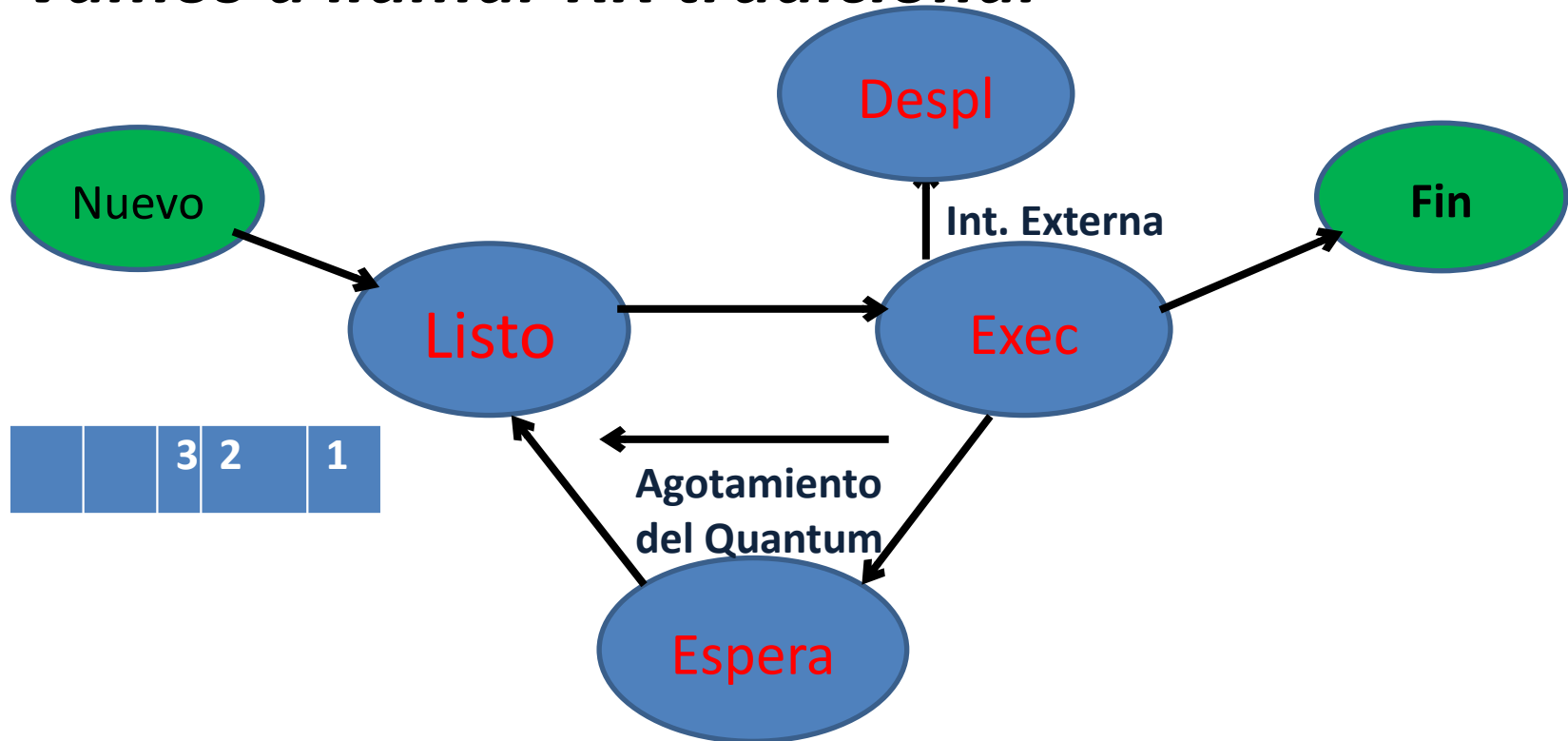
Prioridades - Apropiativo

Reloj	Listo	Ejecucion	Espera	End	Comentario
0	A	SO			CS
1		A			
2		A			
3	BAC	SO			CS
4	AC	B			
5	BACD	SO			CS
6	ACD	B			
7	BEACD	SO			CS
8	EACD	B			
9	EACD	B			
10	EACD	SO		B	CS
11	ACD	E			
12	ACD	E			
13	ACD	SO		BE	CS
14	CD	A			
15	CD	SO		BEA	CS
16	D	C			
17	D	C			
18	D	SO		BEAC	CS
19		D			
20		D			
21		D			
22		D			
23		SO		BEACD	CS

Algoritmo de Round Robin

Ejemplo algoritmo RR

Lo vamos a llamar RR tradicional



Es un algoritmo injusto?

Que pasa con los procesos orientados a CPU?

Algoritmo de Round Robin

Ejemplo algoritmo RR

Job	Hora de Llegada	Tiempo de CPU	Instante - E/S
A	0	5	2 – 1
B	2	4	
C	4	3	1– 2
D	10	4	

Algoritmo Round Robin Q=3 – No Apropiativo

Reloj	Listo	Exec	Espera	End	Comentario
0	A	SO			CS
1		A			
2	A B	SO			Proceso nuevo
3	B	A			Vuelta hasta cumplir q
4	B C	SO	A		No cumple q xq hace E/S
5	B C A	SO			CS – IE – A al final
6	C A	B			
7	C A	B			
8	C A	B			Q=3 y vuelve al final
9	C A B	SO			
10	C A B D	SO			
11	A B D	C			
12	A B D	SO	C		
13	B D	A	C		
14	A B D C	<u>SO</u>			IE
15	B D C	A			
16	B D C	A			
17	B D C	SO		A	CS
18	D C	B			
19	D C	SO		A B	CS
20	C	D			
21	C	D			
22	C	D			Q=3 y D vuelve al final
23	C D	SO			CS
24	D	C			
25	D	C			
26	D	SO		A B C	CS
27		D			
28		SO		A B C D	CS

Calculos para comparar Algoritmos

Tiempo de espera de cada proceso: el tiempo que estuvo en la cola de listo

Ej: $TeA = 11$; $TeB = 13$; $TeC = 17$; $TeD = 14$

$$Tespera_medio = (11 + 13 + 17 + 14)/4 = 13,75$$

Tiempo en E/S:

$$Te/sA = 1; Te/sC = 2$$

Tiempo de retorno: tiempo que el proceso estuvo en el sistema

$$Tretorno = Te + Texec + T e/s$$

$$TrA = 11 + 5 + 1 = 17; TrB = ?; TrC = ?; TrD = ?$$

$$Tretorno_medio = (TrA + TrB + TrC + TrD) / \text{cantidad de procesos}$$

Algoritmo de Round Robin

- *Variante que se uso en el kernel linux*
- *Doble cola y memoria de lo usado*

Se define un Quantum grande para los procesos

Ejemplo $q=30$

Activa



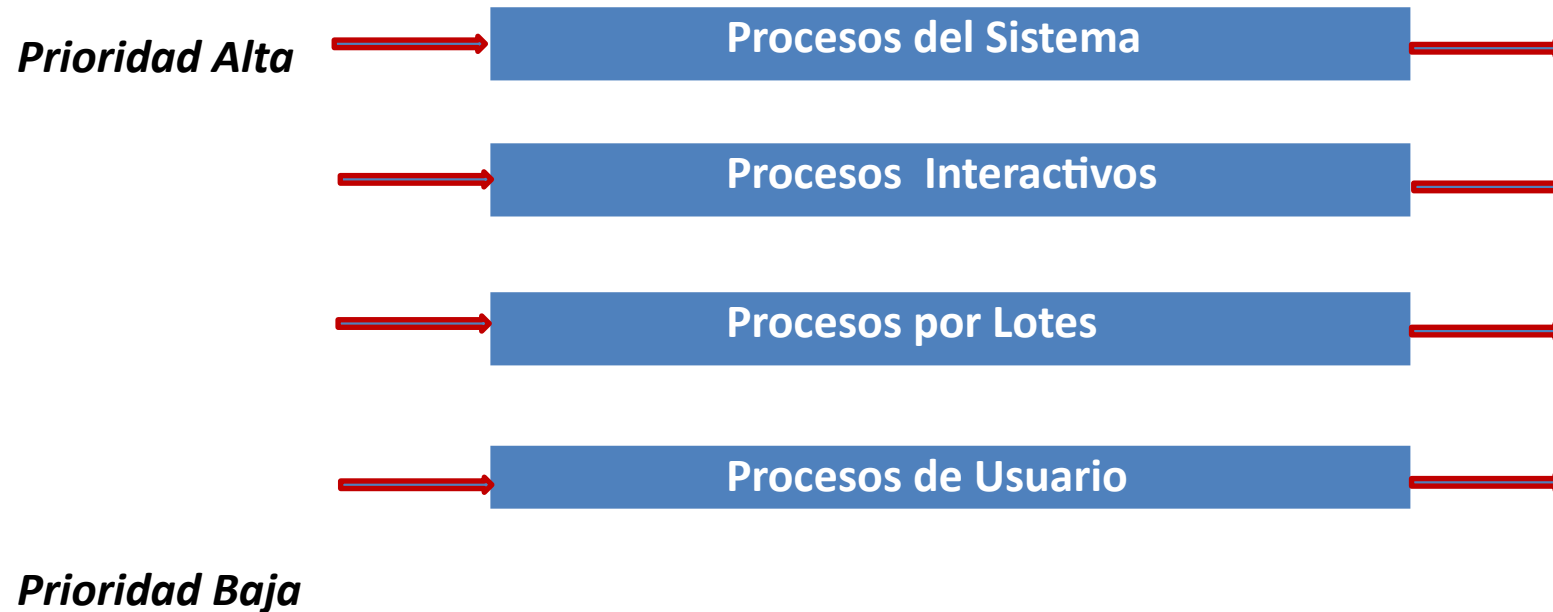
Pasiva



Planificación Colas niveles Múltiples

Los procesos se clasifican en grupos:

- ***Interactivos***
- ***Por lotes***
- ***Procesos del Sistema***
- ***Procesos de Usuario***

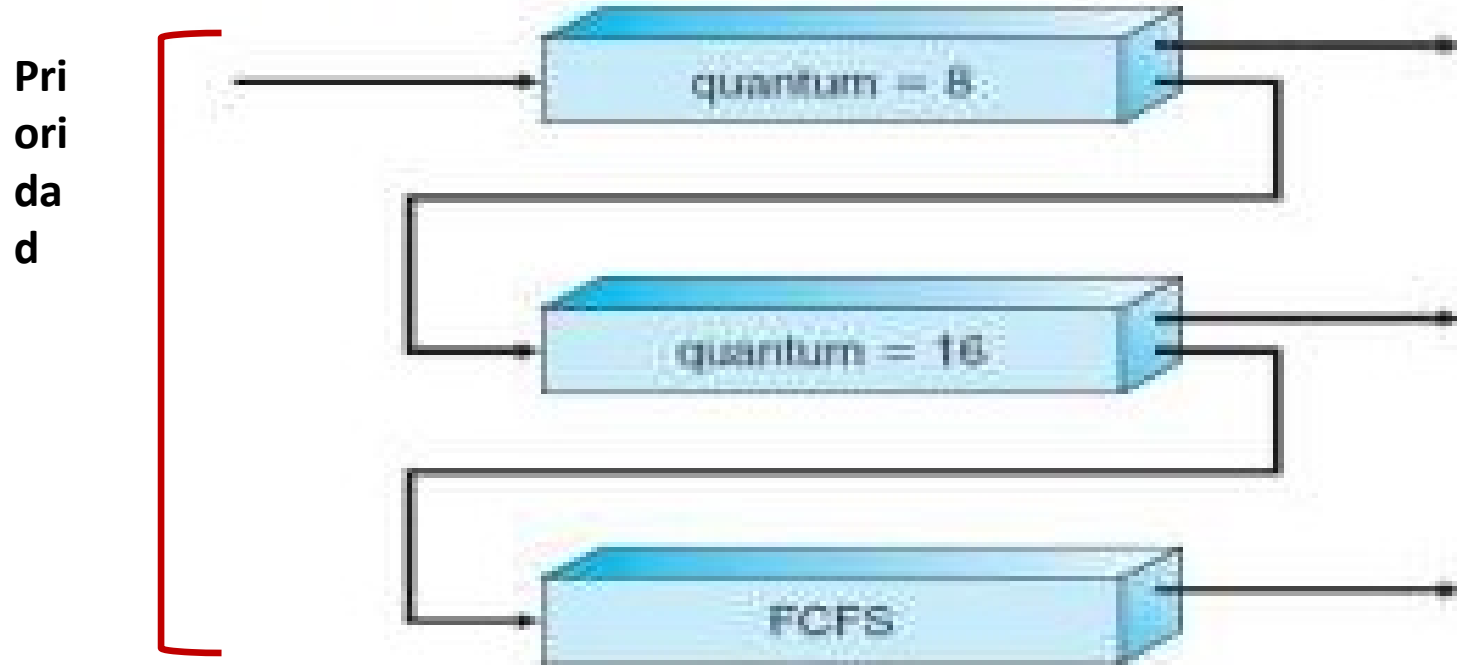


Planificación Colas niveles Múltiples y retroalimentación

Los procesos se mueven entre distintas colas

Separar procesos con distintas ráfagas de CPU

Las colas tendrían distintas prioridades



Fin clase