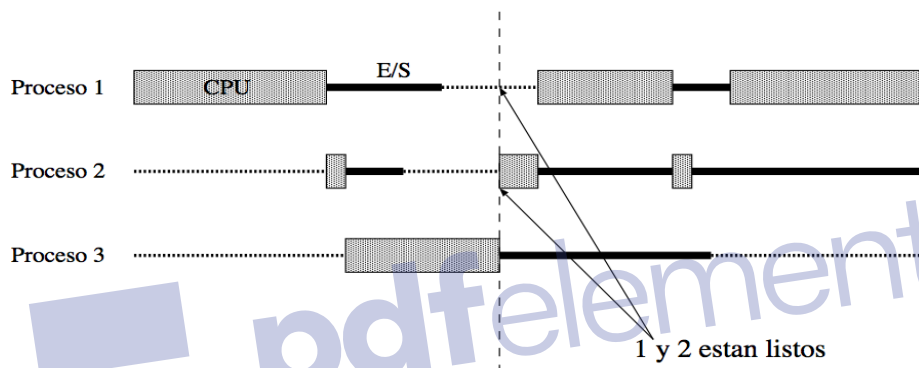


1. Planificador: concepto y objetivos
2. Planificación: medidas de tiempo
3. Algoritmos de planificación
 - 3.1. Algoritmos no apropiativos
 - 3.2. Algoritmos apropiativos
4. Tipos de planificación
 - 4.1. Planificación en Sistemas tiempo real
 - 4.2. Planificación con hebras
 - 4.3. Planificación con multiprocesadores

1. Planificador: concepto y objetivos

- El planificador (scheduler) es la parte del SO que decide a qué proceso preparado se le da paso a CPU.
- Varios procesos o threads listos pueden competir por la CPU
- Los procesos se suelen comportar alternando ráfagas de CPU y de E/S



- Objetivos del planificador (Leer diapositivas)

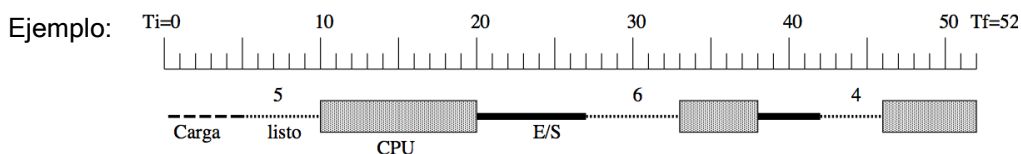
2. Medidas de tiempo

- *Tiempo de retorno*(t_R): Tiempo que está un proceso en el sistema, desde que se inicia (T_i) hasta que finaliza (T_f). Incluye: t_{CPU} y $t_{E/S}$

$$t_R = T_f - T_i$$
- *Tiempo de espera*(t_E): es el tiempo de retorno quitando CPU y E/S

$$t_E = t_R - t_{CPU} - t_{E/S}$$
- *Tiempo de servicio*(t_S): Tiempo que consumiría si fuese el único proceso existente, es decir, el tiempo de retorno menos el tiempo de espera.

$$t_S = t_R - t_E = t_{CPU} + t_{E/S}$$
- *Índice de servicio* (i_S): $i_S = t_S / t_R$



Tiempo de **retorno**: $t_R = T_f - T_i = 52 - 0 = 52$

Tiempo de **CPU**: $t_{CPU} = 10 + 5 + 6 = 21$

Tiempo de **E/S**: $t_{E/S} = 7 + 4 = 11$

Tiempo de **servicio**: $t_S = t_{CPU} + t_{E/S} = 32$

Tiempo de **espera**: $t_E = t_R - t_S = 52 - 32 = 20$

Tiempo de **índice de servicio**: $i_S = 32/52 = 0,615$

3. Algoritmos de planificación

- *Planificación no apropiativa (non-preemptive)*: deja ejecutar al proceso en CPU hasta que éste para por bloqueo (inicio E/S), espera por otro proceso o terminación voluntaria.

- *Planificación apropiativa*: el planificador puede desalojar al proceso en CPU durante su ejecución y cambiarlo por otro. Necesita una interrupción de reloj para poder ejecutarse en períodos regulares de tiempo (quantum)

Video recomendado:

a. <https://www.youtube.com/watch?v=jxGnKR3JoOw>

b. <https://www.youtube.com/watch?v=xQDi62YZuuw>

3.1. Algoritmos no apropiativos

Caso de ejemplo: Sean 5 procesos (A, B, C, D y E)

	A	B	C	D	E
Llegada:	2	4	1	5	3
Servicio:	8	3	4	7	6

- **Algoritmo First-Come-First-Served (FCFS) o (FIFO)**:

Primero en llegar, primero en ser servido. Esto es, el primero proceso en llegar es el primero en ejecutarse sin importar su rafaga de tiempo de CPU.

Del caso de ejemplo, estos se ejecutarían en el mismo orden de llegada: C | A | E | B | D

- **Algoritmo Shortest Job First (SJF)**: Primero el más corto

Del caso de ejemplo: el orden de ejecución sería: B | C | E | D | A

3.2. Algoritmos apropiativos

- **Algoritmo Shortest Remaining Time First (SRTF)**

Cada vez que entran trabajos se interrumpe el actual y se compara el tiempo restante de éste con el de los entrantes.

Si hay un trabajo nuevo más corto que lo que le falta al actual en CPU, echamos el actual y metemos el nuevo.

Es una versión apropiativa de SJF.

Del ejemplo base, primero se ejecuta C, luego es detenido la ejecución de este proceso es detenido y se ejecuta B, luego se termina C, y se continua con E, D y por último A.

- **Algoritmo Round-Robin (RR)**

La ejecución se realiza destinando a cada proceso un quantum de tiempo.

Se suele decir que es el más justo.

Del ejemplo, Los procesos se ejecutan en el orden de llegada de manera cíclica con una rafaga de un quantum destinado a cada ejecución.

4. Tipos de planificación

(Leer diapositiva)













