Programación de Procesos y Servicios

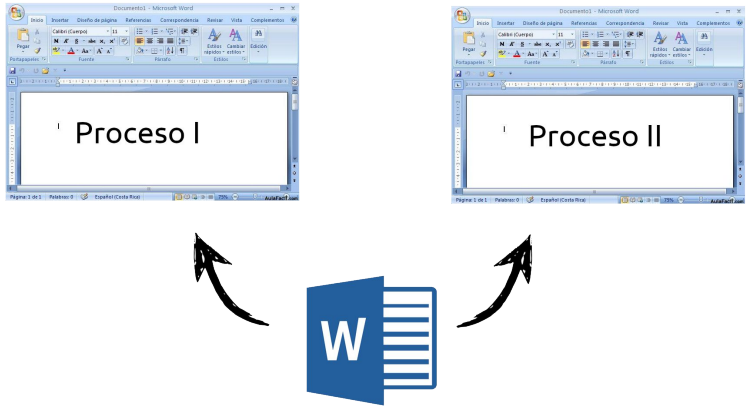
UT 0: Introducción a los Procesos
2º DAM

Conceptos Básicos

Programa: Se puede considerar un programa a toda la información (tanto código como datos) almacenada en disco de una aplicación que resuelve una necesidad concreta para los usuarios.

Proceso: Programa en ejecución. Este concepto no se refiere únicamente al código y a los datos, sino que incluye todo lo necesario para su ejecución.

- Contador de Programa
- Imagen de memoria
- Estado del procesador



Conceptos Básicos

Ejecutable: Un fichero ejecutable contiene la información necesaria para crear un proceso a partir de los datos almacenados de un programa.

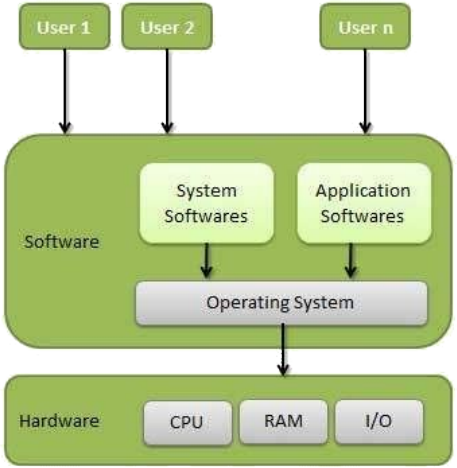
Demonio: Proceso no interactivo que está ejecutándose continuamente en segundo plano, es decir, es un proceso controlado por el sistema sin ninguna intermediación del usuario. Suelen proporcionar un servicio básico para el resto de procesos.

- cron
- ssh
- Garbage Collector en Java

Conceptos Básicos

Sistema Operativo: Intermediario entre el usuario y las aplicaciones que utiliza y el hardware del ordenador.

- Ejecutar los programas del usuario. Es el encargado de crear los procesos a partir de los ejecutables de los programas y de gestionar su ejecución para evitar errores y mejorar el uso del computador.
- Gestionar los recursos de manera eficiente.



Programación Concurrente

La **computación concurrente** permite la posibilidad de tener en ejecución al mismo tiempo múltiples tareas interactivas. Es decir, permite realizar varias cosas al mismo tiempo. Las tareas se pueden ejecutar en:

- Un único procesador (multiprogramación): Aunque parece que varios procesos se ejecutan simultáneamente, sólo un proceso puede estar en ejecución en cada momento.

  Este concepto se denomina programación concurrente. La programación concurrente no mejora el tiempo de ejecución global de los programas ya que se ejecutan intercambiando unos por otros en el procesador. Sin embargo, permite que varios programas parezca que se ejecuten al mismo tiempo

Programación Concurrente

La **computación concurrente** permite la posibilidad de tener en ejecución al mismo tiempo múltiples tareas interactivas. Es decir, permite realizar varias cosas al mismo tiempo. Las tareas se pueden ejecutar en:

- Varios núcleos en un mismo procesador (multitarea): Cada núcleo podría estar ejecutando una instrucción diferente al mismo tiempo. En este caso todos los cores comparten la misma memoria por lo que es posible utilizarlos de forma coordinada mediante lo que se conoce por programación paralela.

  La programación paralela permite mejorar el rendimiento de un programa si este se ejecuta de forma paralela en diferentes núcleos ya que permite que se ejecuten varias instrucciones a la vez.

Programación Concurrente

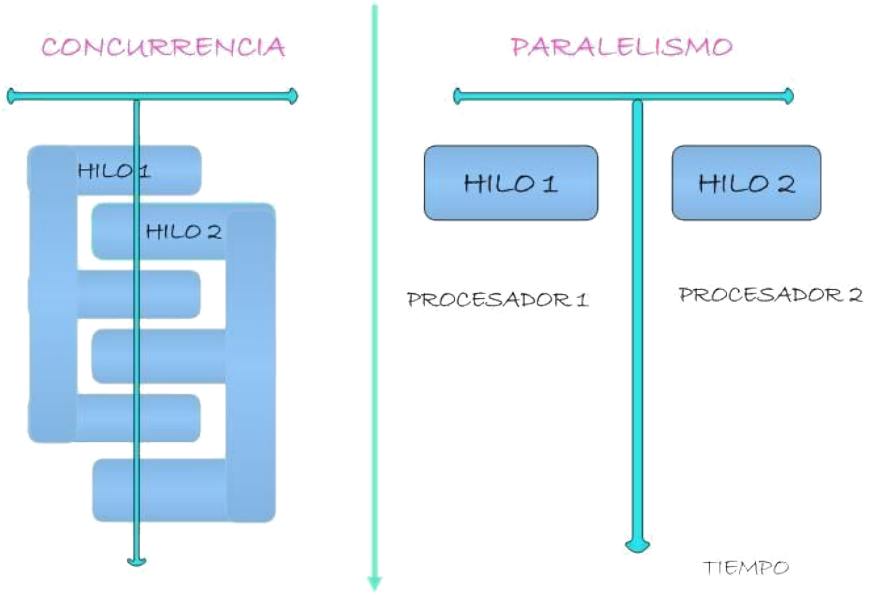
La **computación concurrente** permite la posibilidad de tener en ejecución al mismo tiempo múltiples tareas interactivas. Es decir, permite realizar varias cosas al mismo tiempo. Las tareas se pueden ejecutar en:

- Varios ordenadores distribuidos en red: La programación distribuida posibilita la utilización de un gran número de dispositivos (ordenadores) de forma paralela, lo que permite alcanzar elevadas mejoras en el rendimiento de la ejecución de programas distribuidos.

  Sin embargo, como cada ordenador posee su propia memoria, imposibilita que los procesos puedan comunicarse compartiendo memoria, teniendo que utilizar otros esquemas de comunicación más complejos y costosos a través de la red que los interconecte

Programación Concurrente

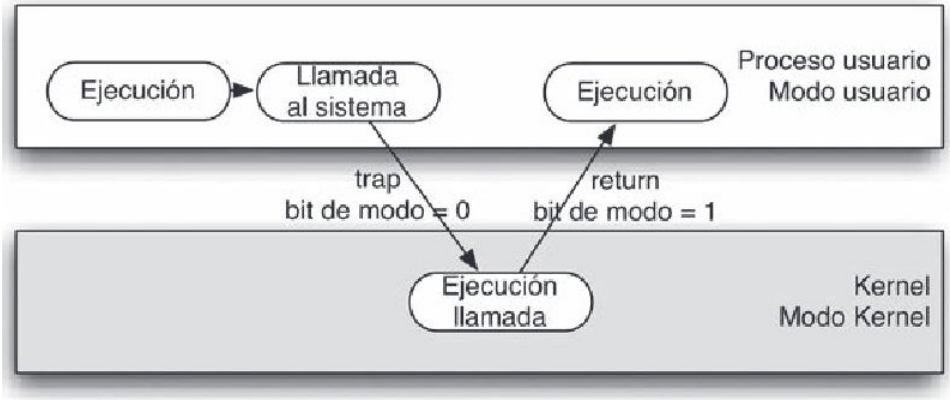
La **computación concurrente** permite la posibilidad de tener en ejecución al mismo tiempo múltiples tareas interactivas. Es decir, permite realizar varias cosas al mismo tiempo. Las tareas se pueden ejecutar en:



Funcionamiento del SO

El **kernel** es la parte central del sistema operativo, encargada de gestionar los recursos del ordenador y responder a las **interrupciones** que suspenden temporalmente la ejecución de un proceso para manejar eventos específicos.

El kernel actúa mediante llamadas al sistema, que permiten a los programas de usuario interactuar con los recursos del sistema de forma segura, evitando ejecutar directamente instrucciones peligrosas.



Funcionamiento del SO

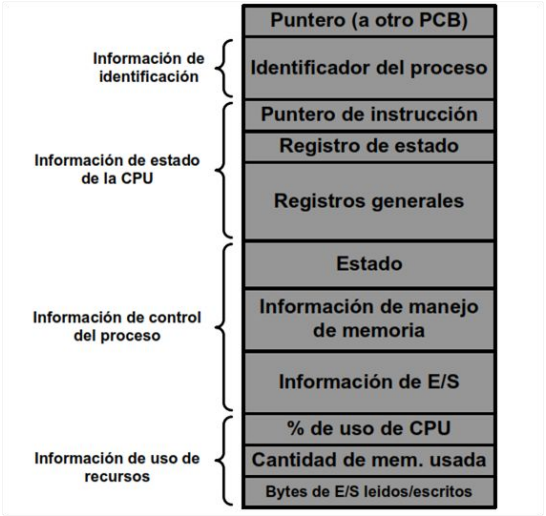
El **modo dual** es una característica del hardware que permite al sistema operativo protegerse. El procesador tiene dos modos de funcionamiento indicados mediante un bit:

- Modo usuario (1): Utilizado para la ejecución de programas de usuario.
- Modo kernel (0): Las instrucciones del procesador más delicadas solo se pueden ejecutar si el procesador está en modo kernel.

Procesos

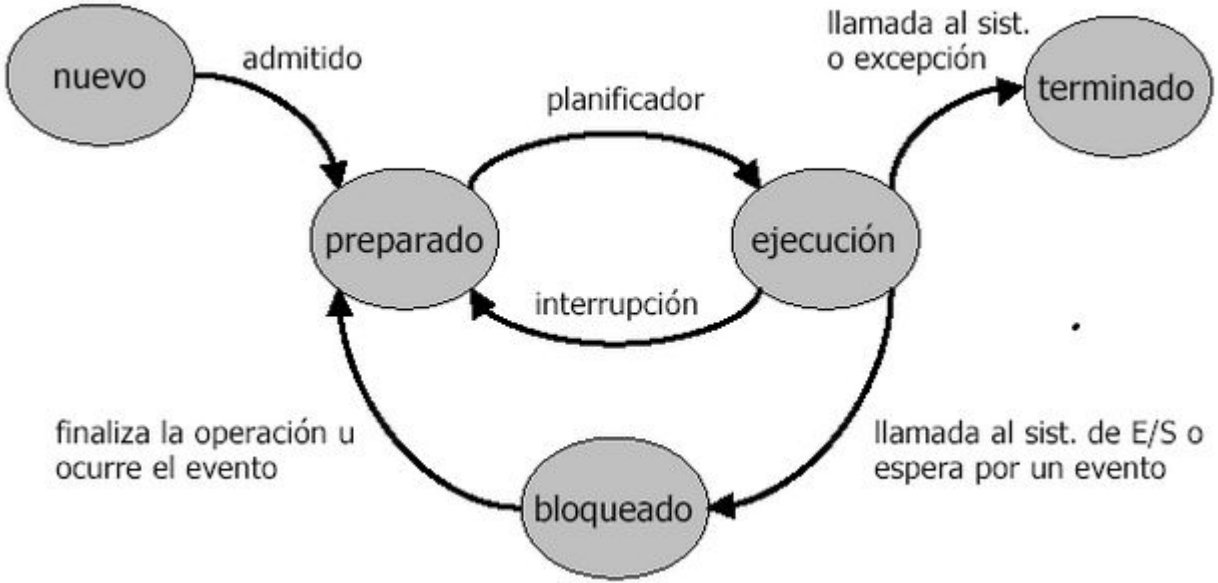
Proceso: Programa en ejecución que realiza una serie de instrucciones de manera independiente. Cada proceso tiene su propio espacio de memoria asignado, que incluye el código ejecutable, los datos y la pila de ejecución.

- La estructura más representativa es el **bloque de control de proceso (PCB)**, donde se almacena la información necesaria para la gestión del mismo.



Procesos

El sistema operativo es el encargado de poner en ejecución y gestionar los procesos. Para su correcto funcionamiento, a lo largo de su ciclo de vida, los procesos pueden cambiar de **estado**.



Planificación de Procesos

Para gestionar las colas de procesos, es necesario un **planificador de procesos**. El planificador es el encargado de seleccionar los movimientos de procesos entre las diferentes colas. Existen dos tipos de planificación:

- A corto plazo
- A largo plazo

Para poder realizar una gestión óptima, el planificador emplea diferentes **algoritmos de planificación**.

Tipos de técnicas de planificación:

- Expulsiva: Un proceso puede ser desalojado de la CPU sin que haya finalizado, y se queda en la cola de procesos en espera.
- No Expulsiva: Un proceso no puede ser expulsado hasta que termina o se bloquea.

Algoritmos de Planificación

Para cada **proceso** debemos conocer:

- Tiempo de Entrada (T_I): Momento en que el proceso entra en el sistema
- Tiempo de Ejecución (T_X): Tiempo que el proceso necesita para su ejecución total
- Tiempo de Respuesta (T_R): Tiempo desde que el proceso llega al sistema hasta que se obtienen resultados
- Tiempo de Espera (T_E): Tiempo que el proceso pasa dentro del sistema en espera

$$T_E = T_R - T_X$$

Algoritmos de Planificación

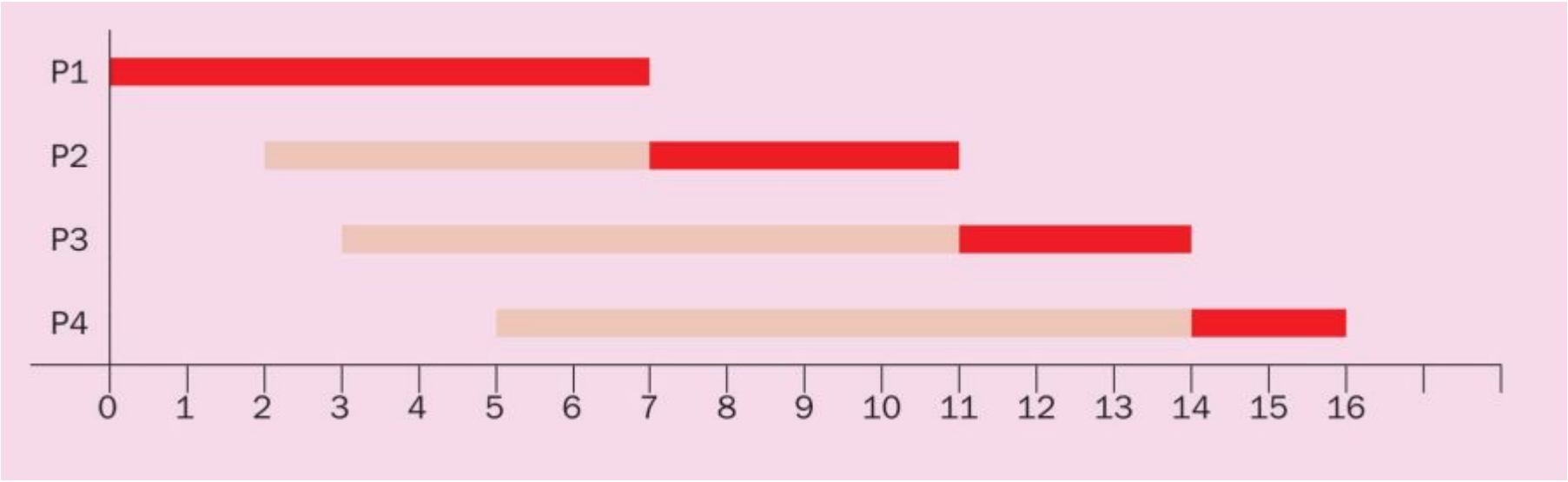
Algoritmo FIFO (First In First Out)

- Utilizando este algoritmo, los procesos se ejecutan en el orden de llegada.
- El primero que llega se empieza a ejecutar.
- Los siguientes deberán esperar su turno para poder empezar a ejecutarse.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algoritmos de Planificación

Algoritmo FIFO (First In First Out):



Algoritmos de Planificación

Algoritmo FIFO (First In First Out)

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	5	9
P3	8	11
P4	9	11
TIEMPOS MEDIOS	**5,5**	**9,5**

Algoritmos de Planificación

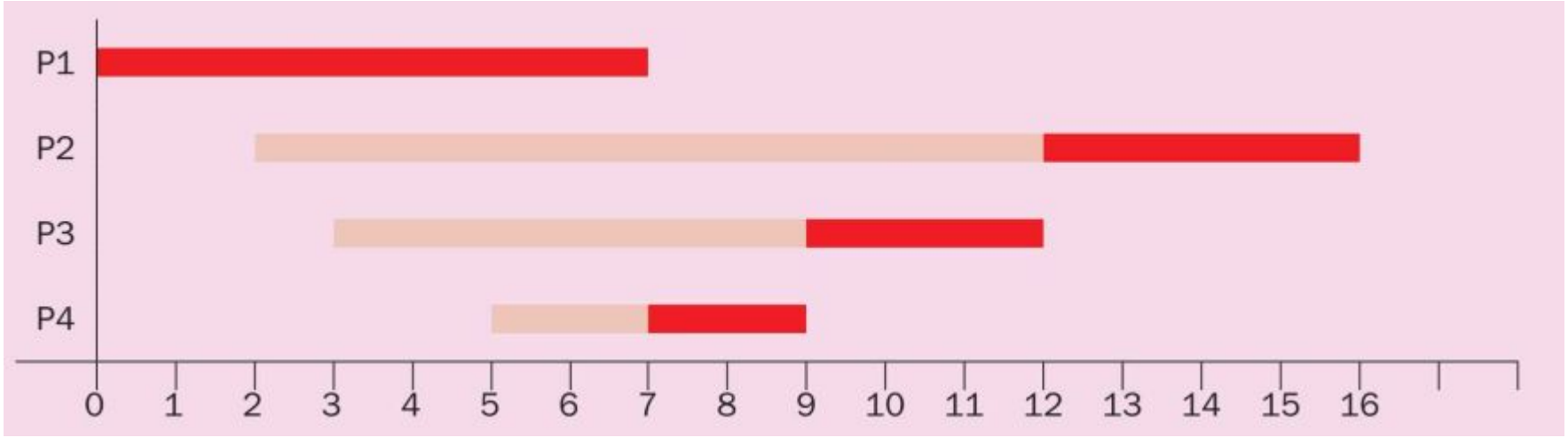
Algoritmo SJF (Short Job First)

- Se coge el proceso más corto de los que están esperando para usar la CPU.
- En caso de igual tiempo, se escoge el primero en entrar.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algoritmos de Planificación

Algoritmo SJF (Short Job First):



Algoritmos de Planificación

Algoritmo SJF (Short Job First)

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	10	14
P3	6	9
P4	2	4
TIEMPOS MEDIOS	**4,5**	**8,5**

Algoritmos de Planificación

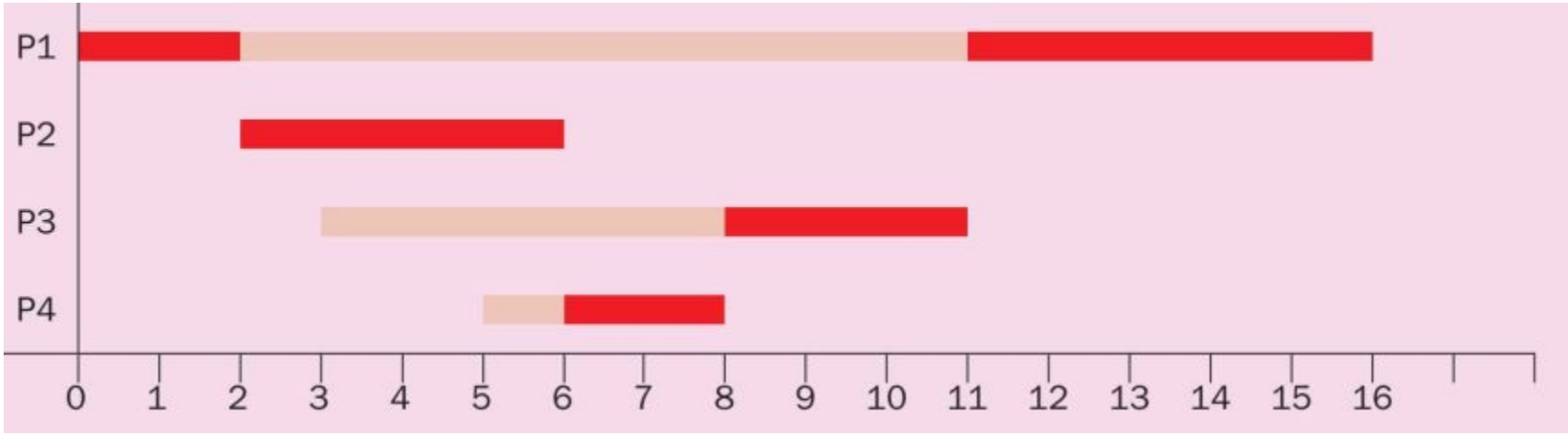
Algoritmo SRTF (Short Remaining Time First)

- Se coge el proceso en espera al que le quede menor tiempo para terminar
- En caso de empate, el primero en entrar
- **Algoritmo expulsivo**: Si mientras se está ejecutando un proceso llega otro que le quede menos tiempo para acabar, lo desplaza

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algoritmos de Planificación

Algoritmo SRTF (Short Remaining Time First)



Algoritmos de Planificación

Algoritmo SRTF (Short Remaining Time First)

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	0	4
P3	5	8
P4	1	3
TIEMPOS MEDIOS	**3,75**	**7,75**

Algoritmos de Planificación

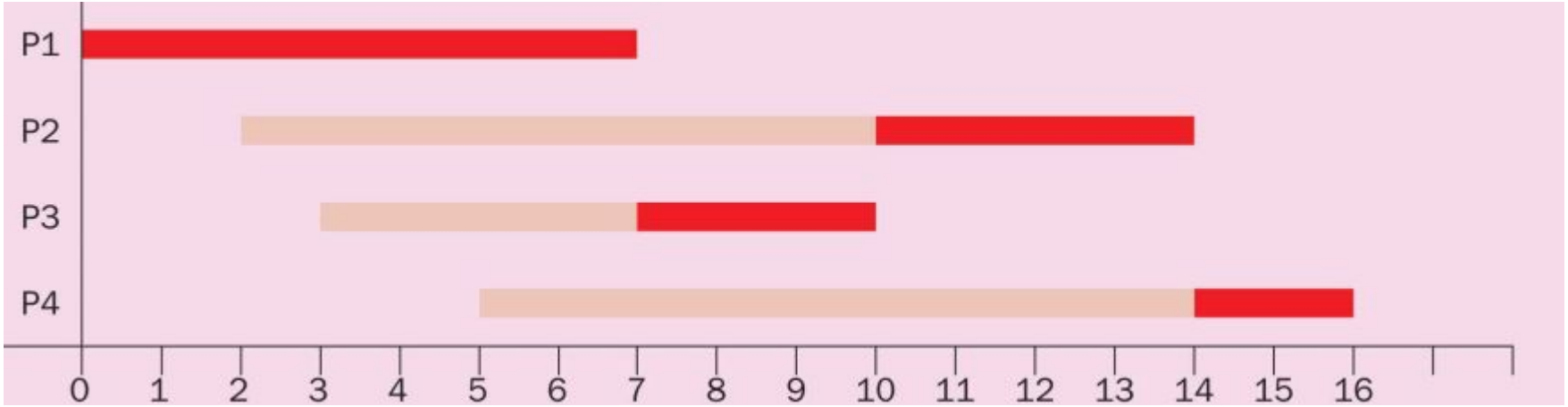
Algoritmo por Prioridades

- Cada proceso tiene una prioridad.
- El orden de entrada en la CPU será según la prioridad. En caso de empate, el primero en llegar

PROCESO	TIEMPO DE LLEGADA	PRIORIDAD	TIEMPO DE EJECUCIÓN
P1	0	4	7
P2	2	2	4
P3	3	1	3
P4	5	3	2

Algoritmos de Planificación

Algoritmo por Prioridades



Algoritmos de Planificación

Algoritmo por Prioridades

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	8	12
P3	4	7
P4	9	11
TIEMPOS MEDIOS	**5,25**	**9,25**

Algoritmos de Planificación

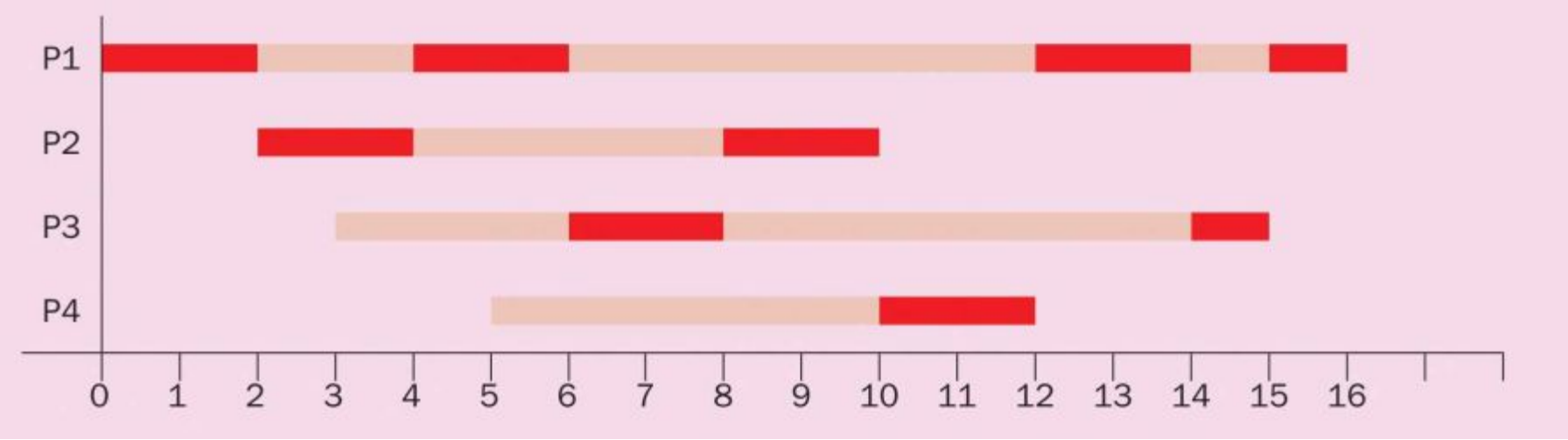
Algoritmo Round Robin (RR)

- Se va dando tiempo de ejecución a cada proceso que esté en espera.
- Se debe establecer un tiempo o quantum (q), tras el cual el proceso abandona la CPU y da paso al siguiente en orden de entrada.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algoritmos de Planificación

Algoritmo Round Robin (RR)



Algoritmos de Planificación

Algoritmo Round Robin (RR)

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	4	8
P3	9	12
P4	5	7
TIEMPOS MEDIOS	**6,75**	**10,75**

Cambio de Contexto

Cuando el procesador pasa a ejecutar otro proceso el sistema operativo debe guardar el contexto del proceso actual y restaurar el contexto del proceso que el planificador a corto plazo ha elegido ejecutar. La salvaguarda de la información del proceso en ejecución se produce cuando hay una interrupción.

Se conoce como **contexto** a:

- Estado del proceso
- Estado del procesador: valores de los diferentes registros del procesador.
- Información de gestión de memoria: espacio de memoria reservada para el proceso.

El cambio de contexto es tiempo perdido, ya que el procesador no hace trabajo útil durante ese tiempo

Árbol de procesos

Aunque el responsable del proceso de creación de un nuevo proceso es el sistema operativo, el nuevo proceso se crea siempre por petición de otro proceso.

Cualquier proceso en ejecución siempre depende del proceso que lo creó, estableciéndose un vínculo entre ambos. A su vez, el nuevo proceso puede crear nuevos procesos, formándose lo que se denomina un **árbol de procesos.**