6.3.2 Tipos de planificaciones

Entre as distintas planificacións existen dous tipos principais:

- Apropiativa: unha vez que se lle otorgou a CPU a un proceso, pódeselle retirar.
- Non apropiativa: unha vez que se lle otorgou a CPU a un proceso, non se lle pode retirar.

Á operación de desaloxar un proceso da CPU para que outro empece a executarse chámaselle **cambio de contexto.** Esta operación debe realizarse coa maior rapidez posible e nela débese almacear a información sobre o estado da execución do proceso, para que cando volva a estar en execución, se reinicie desde o mesmo punto e cos mesmos valores que tiña cando se desaloxou.

6.3.2 Tipos de planificaciones

Existen unha serie de algoritmos de planificación que lle indican ao planificador qué proceso (dos que están en espera) se debe elexir para pasar a executarse. Estos algoritmos deben **maximizar a utilización da CPU** evitando que esté libre nalgún momento e **minimizar o tempo de resposta** de cada proceso.

En cada algoritmo saberemos para cada proceso:

- Tempo de entrada ou de chegada ao sistema (T_0): é o momento no que o proceso entra no sistema.
- Tempo de execución (T_x): tempo que o proceso necesita para a súa execución total.

Para cada proceso interésanos obter:

- Tempo de resposta ou tempo de retorno (T_R): tempo que pasa desde que o proceso chega ao sistema ata que se obteñen os resultados.
- Tempo de espera (T_E): tempo que o proceso está esperando no sistema.

$$T_E = T_R - T_x$$

6.3.3 Algoritmo FIFO

- First Input First Output, First Come First Served → Primeiro en entrar, primeiro en sair.
- Utilizando este algoritmo, os procesos execútanse en orden de chegada. O primeiro que chega empeza a executarse e os seguintes deberán esperar o seu turno para poder empezar a executarse.
- Non apropiativo

6.3.3 Algoritmo FIFO

ACTIVIDAD RESUELTA

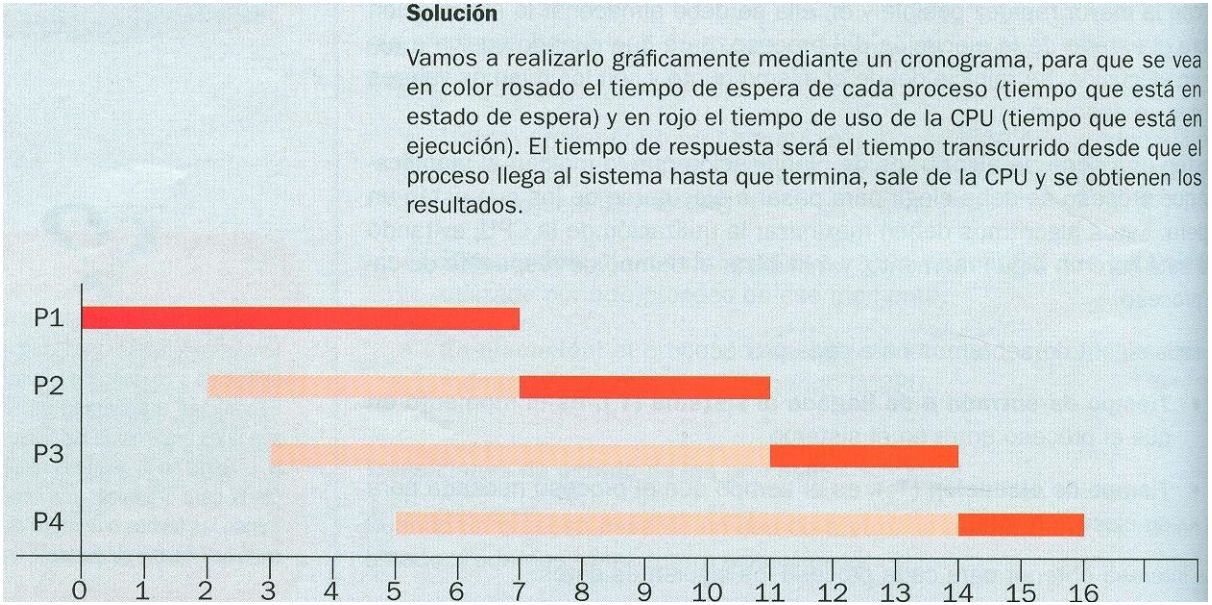
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo FIFO o FCFS.

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

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**

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



6.3.4 Algoritmo SJF

- Shortest Job First → Traballo máis corto primeiro.
- Este algoritmo colle, dos que están esperando, o proceso máis corto para asignarlle o uso da CPU. En caso de igualdade aplícase o FIFO.
- Non apropiativo.

6.3.4 Algoritmo SJF

ACTIVIDAD RESUELTA

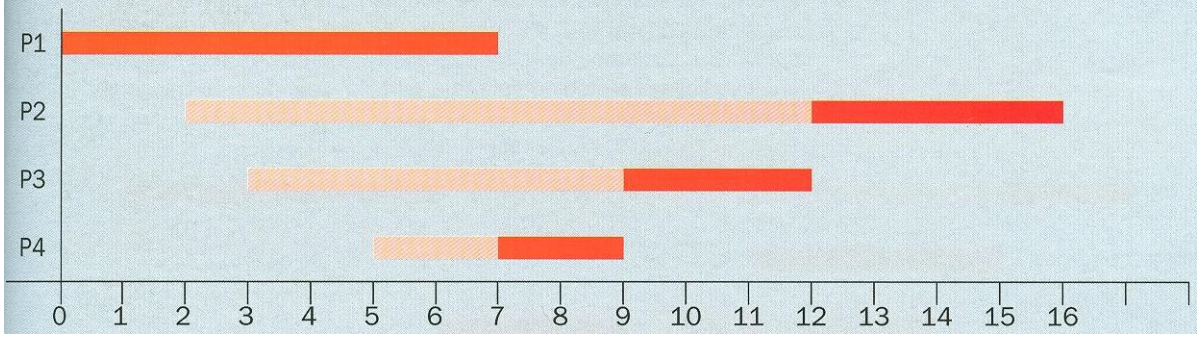
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo SJF.

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

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**

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



6.3.5 Algoritmo SRTF

- Short Remaining Time First → Menor tempo restante primeiro.
- Este algoritmo vai seleccionando, dos que están esperando, ao proceso que lle quede menor tempo para terminar a súa execución. En caso de empate utilízase o FIFO.
- Apropiativo: se mentres se está executando un proceso chega outro ao que lle queda menos tempo para acabar a súa execución, o proceso que se está executando é desprazado por este novo proceso producíndose un cambio de contexto.

6.3.5 Algoritmo SRTF

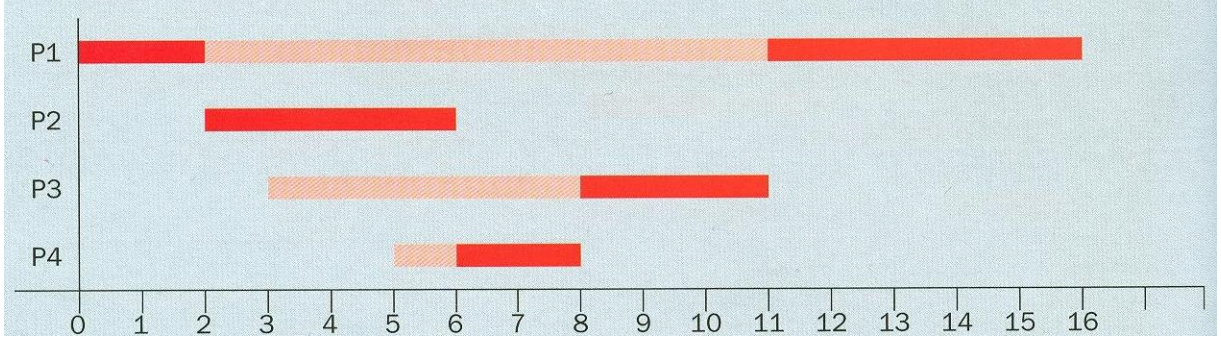
ACTIVIDAD RESUELTA

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

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**

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



6.3.6 Algoritmo por prioridades

- Asocia a cada proceso unha prioridade.
- A orde de entrada na CPU será según a prioridade e en caso de empate aplícase o FIFO.
- A prioridade sóese indicar cun número enteiro de modo que canto máis baixo sexa, maior prioridade terá o proceso.
- Existen variantes deste algoritmo, que pode ser apropiativo ou non apropiativo, e tamén se lle pode ir cambiando a prioridade a un proceso, é dicir, facelo máis prioritario a medida que leve máis tempo esperando a CPU para evitar que procesos pouco prioritarios se posterguen indefinidamente se chegan procesos máis prioritarios.
- Non favorece aos procesos que necesiten máis ou menos tempo de CPU senón que optimiza o tempo de resposta dos máis prioritarios.

6.3.6 Algoritmo por prioridades

ACTIVIDAD RESUELTA

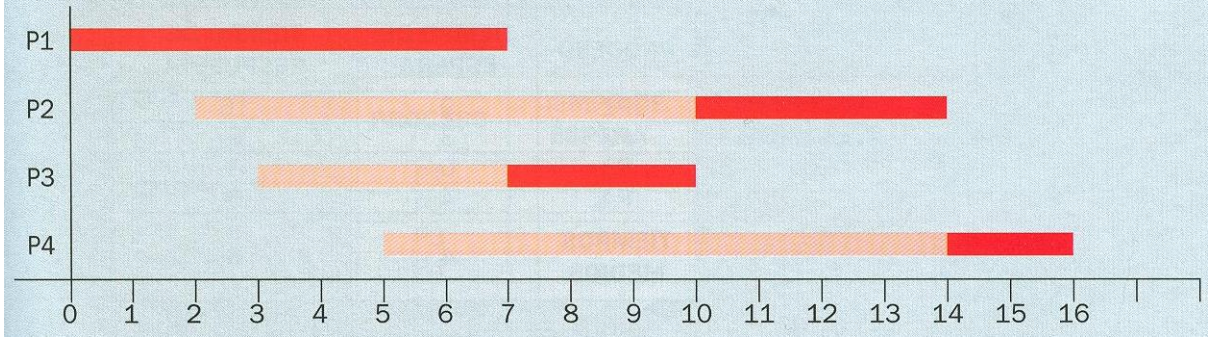
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo por prioridades no expulsivo.

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

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**

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



6.3.6 Algoritmo por prioridades

ACTIVIDAD RESUELTA

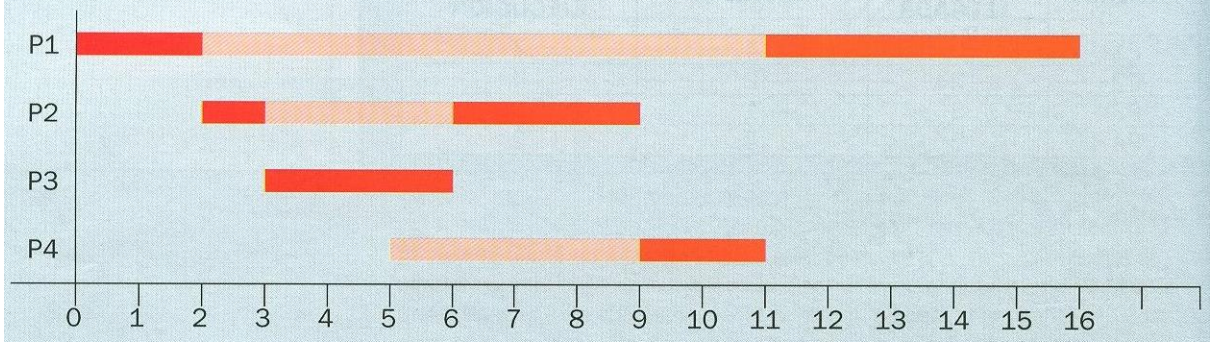
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo por prioridades expulsivo.

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

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	3	7
P3	0	3
P4	4	6
TIEMPOS MEDIOS	**4**	**8**

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



6.3.7 Algoritmo Round Robin

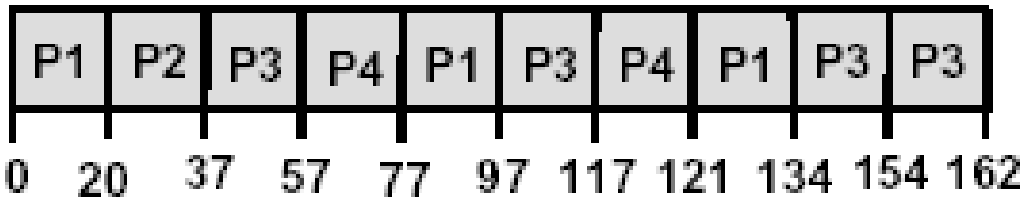
- Este algoritmo vai dando tempo de execución a cada proceso que está en espera.
- Débese establecer un tempo ou quantum (q), tras o cal o proceso abandona a CPU e dá paso ao seguinte, seguindo a orden FIFO.
- Se non hai procesos en espera, o que está usando a CPU seguirá ata que chegue algún outro proceso.
- Algoritmo apropiativo.

6.3.7 Algoritmo Round Robin

- Ejemplo:

Proceso	*Ráfaga CPU*
P1	53
P2	17
P3	68
P4	24

  - Con q=20
    - ✓ Diagrama de Gant para la planificación:



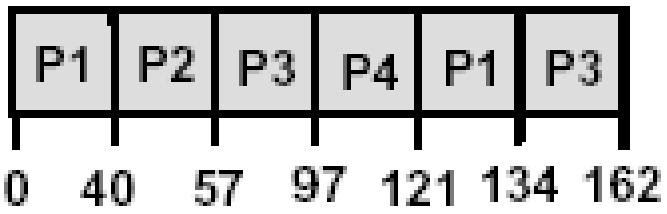
    - ✓ 9 cambios de contexto

6.3.7 Algoritmo Round Robin

- Ejemplo:

Proceso	*Ráfaga CPU*
P1	53
P2	17
P3	68
P4	24

  - Con q=40
    - ✓ Diagrama de Gant para la planificación:



    - ✓ 5 cambios de contexto