-EJERCICIO 1-

Supongamos que llegan al sistema 5 procesos tal como se muestra en la siguiente tabla:

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE CPU
P1	2	4
P2	0	7
P3	1	3
P4	3	2
P5	5	5

Representar la ejecución en el tiempo de cada proceso para los siguientes algoritmos de planificación: (poner en el cuadro correspondiente E-Espera P-Proceso ejecutándose)

FCFS.-

P1			E	E	E	E	E	E	E	E	P	P	P	P							
P2	P	P	P	P	P	P	P														
P3		E	E	E	E	E	E	P	P	P											
P4				E	E	E	E	E	E	E	E	E	E	E	P	P					
P5						E	E	E	E	E	E	E	E	E	E	E	P	P	P	P	P
	1		3		5		7		9		11		13		15		17		19		21

En el algoritmo FCFS, los procesos usan la CPU en el orden en que llegaron. Es un algoritmo no apropiativo, es decir, a medida que llegan los procesos van usando la CPU y cuando hacen uso de ésta, no la abandonan hasta que se bloqueen por E/S o finalicen.

- El Primer proceso que entra a ejecutarse es el P2 y se ejecuta hasta el final.
- Mientras se está ejecutando llegaron el resto de los procesos y se pusieron a la espera.
- Una vez finalizo el proceso P2 se empieza a ejecutar el P3 ya que es el siguiente en orden de llegada, y se ejecuta hasta que finaliza.
- Seguidamente se ejecuta el P1.
- Una vez finalizado el P1 se ejecuta el siguiente en orden de llegada que es el P4.
- Y para finalizar después de terminar de ejecutar el proceso P4, se ejecuta el P5.

SJF.-

P1			E	E	E	E	E	E	E	E	E	E	P	P	P	P					
P2	P	P	P	P	P	P	P														
P3		E	E	E	E	E	E	E	E	P	P	P									
P4				E	E	E	E	P	P												
P5						E	E	E	E	E	E	E	E	E	E	E	P	P	P	P	P
	1		3		5		7		9		11		13		15		17		19		21

Es un algoritmo no apropiativo. Cuando los procesos llegan a la cola de listos, éstos se ordenan según quien tiene el menor tiempo de CPU. El que queda a la cabeza, es quien pasa a usar la CPU y se ejecuta hasta que finaliza.

- En el instante 0 miramos que procesos están disponibles y entre los que tenemos escogemos el que menos tiempo de ejecución necesite.
- Como en el instante 0 sólo tenemos el P2, es este quien entra a ejecutarse hasta el final y según van llegando el resto de procesos se ponen a la espera.
- En el momento que termina de ejecutarse el proceso P2, de entre los que están esperando se escoge el que menor tiempo de CPU necesita (proceso P4).
- A continuación, después de terminar el proceso P4, el siguiente en ejecutarse será el P3 (ya que de los que quedan es el que menor tiempo de CPU necesita).
- Después de ejecutar el proceso P3, sólo nos quedan P1 y P5, cómo P1 es el que menos tiempo de ejecución necesita se ejecuta este.
- Y para finalizar se ejecuta el proceso P5.

SRTF.-

P1			E	E	E	E	P	P	P	P											
P2	P	E	E	E	E	E	E	E	E	E	E	E	E	E	E	P	P	P	P	P	P
P3		P	P	P																	
P4				E	P	P															
P5						E	E	E	E	E	P	P	P	P	P						
	1		3		5		7		9		11		13		15		17		19		21

Es similar al SJF, con la diferencia de que si un nuevo proceso pasa a listo se activa el dispatcher para ver si es más corto que lo que queda por ejecutar del proceso en ejecución. Si es así, el proceso en ejecución pasa a listo y su tiempo de estimación se decremento con el tiempo que ha estado ejecutándose. Es un proceso apropiativo (hay procesos que se apropian de tiempos de CPU de otros procesos).

- En el instante 0 sólo tenemos el P2, es este quien entra a ejecutarse.
- En el instante 1 llega el proceso P3, como este tiene de tiempo de CPU 3 y al proceso que se está ejecutando le queda 6 de CPU, se apropia de su tiempo de CPU para entrar a ejecutarse. El proceso P3 vuelve a la lista de Listos, quedándole aún de tiempo de CPU 6.
- En el instante 2, llega el proceso P1 pero como tiene de tiempo de CPU 4 y al proceso que se está ejecutando le queda 2, se continúa ejecutando P3.
- En el instante 3, llega el proceso P4 pero como tiene de tiempo de CPU 2 y al proceso que se está ejecutando le queda 1, se continúa ejecutando P3.
- En el instante 4, el proceso P3 termino de ejecutarse, y empieza a ejecutarse el proceso P4 por ser el que menos tiempo de CPU necesita.
- En el instante 5 llega el proceso P5 pero como tiene tiempo de CPU 5, se continúa ejecutando el proceso P4 porque sólo le quedan 1 tiempos de CPU.
- En el instante 6, el proceso P4 termino de ejecutarse y el siguiente a entrar a ejecutarse es el proceso P1 por tener menos tiempo de tiempo de CPU.

- En el instante 10, el proceso P1 termino de ejecutarse y el siguiente a entrar a ejecutarse es el proceso P5 ya que este necesita 5 tiempos de CPU y al proceso P2 aún le quedan 6.
- Para finalizar se ejecuta el proceso P2 por ser el único que queda.

RR (Round Robin) Con QuanTum 4 .-

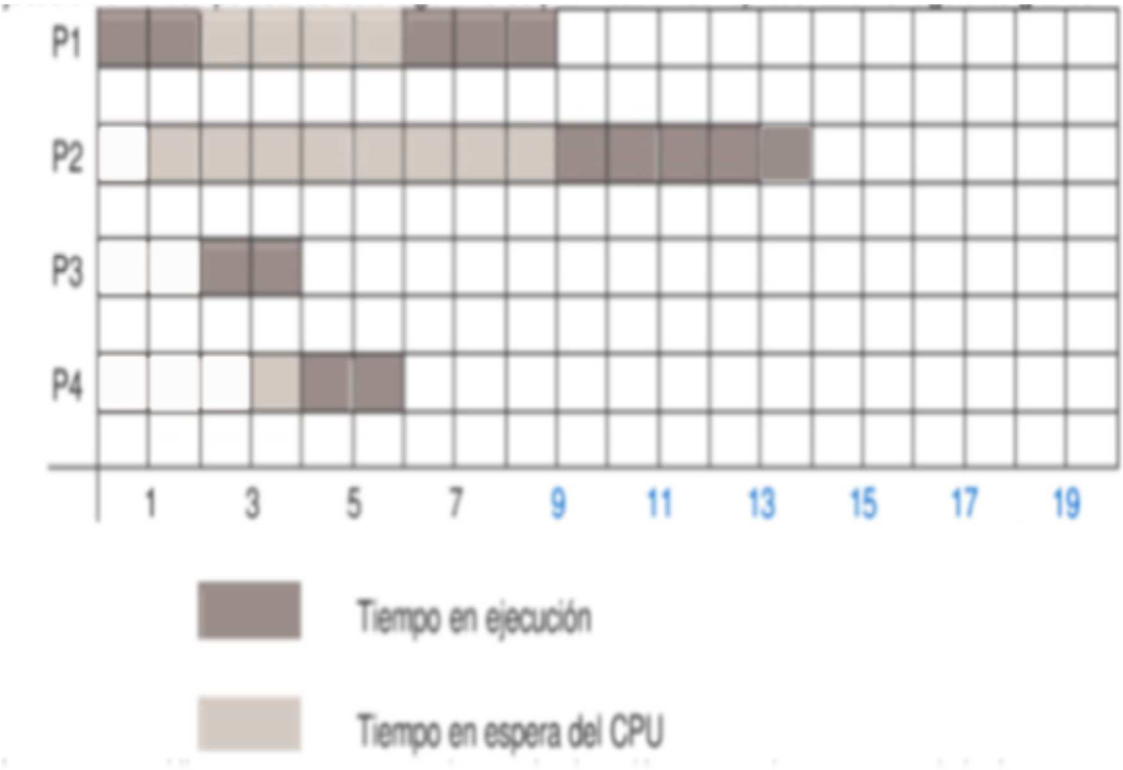
P1			E	E	E	E	E	P	P	P	P										
P2	P	P	P	P	E	E	E	E	E	E	E	E	E	E	E	E	E	P	P	P	
P3		E	E	E	P	P	P														
P4				E	E	E	E	E	E	E	E	P	P								
P5						E	E	E	E	E	E	E	E	P	P	P	P	E	E	E	P
	1		3		5		7		9		11		13		15		17		19		21

El algoritmo RR es apropiativo. El proceso puede abandonar antes la CPU en caso de que finalice o se bloquee por una E/S; en el caso contrario, en el momento que se le acabe el quantum de tiempo en la CPU, vuelve a la cola de listos. La cola de procesos es FIFO (primero en entrar primero en salir) y se estructura como una cola circular (cuando llega al último vuelve a empezar con el primero).

- El Primer proceso que entra a ejecutarse es el P2 y se ejecuta hasta que se le acaba el quantum de tiempo (4).
- En el instante 4, al proceso P2 se le acabo su quantum y aún le queda por ejecutar 3 tiempos de CPU, P2 pasa de nuevo a su posición en la cola listos y el siguiente en la cola es el proceso P3 que entra a ejecutarse.
- En el instante 7 el proceso P3, antes de que se le acabase su quantum de tiempo en la CPU, finaliza su ejecución siendo P1 el siguiente proceso en la ejecución.
- En el instante 11, el proceso P1 finaliza su quantum y también termina el proceso ya que tenia de tiempo 4. El siguiente proceso a ejecutarse es el P4.
- En el instante 13 el proceso P4, antes de que se le acabase su quantum de tiempo en la CPU, finaliza su ejecución siendo P5 el siguiente proceso en la ejecución.
- En el instante 17 el proceso P5 termina su quantum, y como la cola FIFO ya se terminó pasamos de nuevo al principio de la cola (por eso lo de la cola circular).
- De entre los procesos que todavía no han terminado de ejecutarse el siguiente es el P2 que le quedan aún 3 tiempos de CPU.
- P2 antes de terminar su quantum finaliza su ejecución y pasa a ejecutarse el proceso P5 que sólo le queda 1 tiempo de CPU.

-EJERCICIO 2-

En el siguiente gráfico de los tiempos de ejecución de unos procesos en la CPU:



En relación al gráfico contestar a las siguientes preguntas:

- ¿En que instante se produce una apropiación de la CPU por parte de otro proceso?
  En el instante 2 se produce una apropiación de la CPU
- ¿Qué proceso se apropia de la CPU?
  Es el Proceso P3 quien se apropia de la CPU
- ¿A qué proceso le quita la CPU?
  Al proceso P1 le quita la CPU, quedándole aún 3 tiempos de CPU por ejecutar

-EJERCICIO 3-

Considerar un controlador de disco con la cabeza lectora posicionada en la pista 99 y la dirección de búsqueda creciente. La cola de peticiones es la siguiente:

Peticiones:	**70**	**142**	**10**	**2**	**80**	**145**	**17**	**125**

Mostrar el orden en que se atienden las peticiones según los siguientes algoritmos:

FCFS:

Disco:	2	10	17	70	80	99	125	142	145
Orden en el que atiende a las peticiones:	4º	3º	7º	1º	5º	▲	8º	2º	6º

Primero en llegar, primero en ser servido) es la planificación más sencilla. Como se desprende de su propio nombre se dará servicio a las solicitudes de acceso a disco de la cola según el orden de llegada de dichas solicitudes. Esta planificación hará uso de una cola tipo FIFO (First In, First Out – Primero en entrar, primero en salir).

SSF:

Disco:	2	10	17	70	80	99	125	142	145
Orden en el que atiende a las peticiones:	5º	4º	3º	2º	1º	▲	6º	7º	8º

Atiende primero la solicitud de la cola de solicitudes pendientes que quiere acceder al cilindro más cercano al de la solicitud actual, que se está procesando. Es decir, atiende primero la petición que requiere el menor movimiento de la cabeza de lectura/escritura desde su posición actual.

Primero el más cercano al 99 es el 80, después el más cercano al 80 el 70 y así hasta llegar al 2 que el más cercano de los que quedan es el 125 y después de este el 142 y 145.

SCAN:

Disco:	2	10	17	70	80	99	125	142	145
Orden en el que atiende a las peticiones:	8º	7º	6º	5º	4º	▲	1º	2º	3º

También llamada algoritmo del ascensor porque se comporta como tal: va dando servicio a las solicitudes que van encontrando en el sentido en el que se van desplazando las cabezas de lectura/escritura por el disco. Cuando no hay más solicitudes en ese sentido, o se llega al extremo, se invierte el sentido para hacer lo mismo otra vez pero yendo hacia el otro lado. Por tanto, en este algoritmo es necesario tener un bit que indique el sentido del movimiento.

C-SCAN:

Disco:	2	10	17	70	80	99	125	142	145
Orden en el que atiende a las peticiones:	4º	5º	6º	7º	8º	▲	1º	2º	3º

El brazo del disco se mueve en un único sentido, y de forma circular. Sólo se atenderá la petición más cercana en el sentido en el que estemos recorriendo el disco. Una vez alcanzada la última pista, volvemos a la primera pista.

Primero partimos de la pista 99 y vamos hacia la derecha cogiendo

125 - 142 - 145

A continuación el brazo vuelve a la primera pista recorriendo hacia la derecha y cogiendo:

2 - 10 – 17 – 70 - 80