

#### 2.1. Algoritmos de planificación.

Veamos algunos de los algoritmos de planificación más comunes puestos en práctica por el planificador a la hora de asignar tiempos de UCP en la ejecución de procesos. Recordemos que, en este punto, aunque hablemos de procesos, servirá la misma forma de planificación para los hilos. Citaremos algunos de los algoritmos de planificación más importantes. Para ello, nos centraremos en un ejemplo que consistirá en la ejecución de cinco procesos, A, B, C, D y E, que irán llegando y solicitando ciclos de UCP. Según el algoritmo de planificación, necesitarán más o menos ciclos y acabarán de ejecutarse antes o después (ciclo final en las tablas de ejemplo) según las prioridades concedidas por la UCP a través del algoritmo de planificación. Suponemos en todos ellos que los procesos A, B, C, D y E llegan en los ciclos 0, 2, 4, 6 y 8, respectivamente, y con independencia del algoritmo utilizado. Y que cada uno de ellos necesita respectivamente 3, 6, 4, 5 y 2 ciclos de UCP para completar su ejecución. Ilustraremos a través de tablas en qué momento llegan, cuándo empiezan a ser atendidos por la UCP (ciclo inicial), cuál es el número de ciclos necesario para que se ejecute cada proceso y en qué momento acabarán de hacerlo (ciclo final).

Para apreciar más claramente la gestión que la UCP hace de estos procesos, lo ejemplificaremos todo con algunos casos prácticos. En realidad, para la obtención de las tablas que a continuación aparecen en cada algoritmo de planificación, sería necesario que primero hiciésemos el seguimiento gráfico que hacemos al final.

Los algoritmos de planificación son los siguientes:

##### A. Algoritmo de rueda.

Asigna rotativamente tiempos de ejecución a los diferentes procesos. También se llama de **Round-Robin** y en él la asignación de tiempos de ejecución a los procesos es la misma y de forma secuencial. A cada uno se le asigna el mismo **quantum** o intervalo de tiempo de ejecución. La selección entre los procesos se realiza mediante una cola **FIFO (First In First Out**, el primero en entrar es el primero en salir o ser servido). En la Tabla 2.1 vemos un ejemplo.

PROCESO	CICLO DE LLEGADA	CICLOS TOTALES DE UCP	CICLO INICIAL	CICLO FINAL
A	0	3	1	4
B	2	6	3	18
C	4	4	6	17
D	6	5	8	20
E	8	2	11	15

  

ASIGNACIÓN DE CICLOS DE UCP A LOS PROCESOS SEGÚN EL ALGORITMO DE PLANIFICACIÓN ROUND ROBIN O DE RUEDA																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	■	■	■																	
B		■		■		■		■		■		■		■		■		■		
C			■		■		■		■		■		■		■		■		■	
D				■		■		■		■		■		■		■		■		■
E					■		■		■		■		■		■		■		■	
LEYENDA																				
■	EN EJECUCIÓN																			
■	PREPARADO																			

**Tabla 2.1. Ejemplo de algoritmo Round-Robin.**

### 3 Gestión de los recursos de un sistema operativo – Documentos CEO

Como podemos apreciar, cuando llega un nuevo proceso y hay otro en ejecución, los ciclos de UCP se distribuyen entre ellos, pero ejecutándose antes un ciclo de UCP para el proceso que está en activo y no para el recién llegado, al que se le asignará su ciclo inmediatamente después.

#### B. Algoritmo FCFS (*First Come First Serve*).

Los ciclos de UCP asignados a cada proceso se asignan en función de una cola FIFO. Al primer proceso que llega se le asignan tiempos o ciclos de UCP hasta que termina completamente. A continuación, se ejecuta completo el siguiente proceso que hay en la cola FIFO y así sucesivamente hasta terminar con el último proceso (véase la Tabla 2.2).

PROCESO	CICLO DE LLEGADA	CICLOS TOTALES DE UCP	CICLO INICIAL	CICLO FINAL
A	0	3	1	3
B	2	6	4	9
C	4	4	10	13
D	6	5	14	18
E	8	2	19	20

ASIGNACIÓN DE CICLOS DE UCP A LOS PROCESOS SEGÚN EL ALGORITMO DE PLANIFICACIÓN FCFS																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	■	■	■																	
B		■	■	■	■	■	■	■	■											
C				■	■	■	■	■	■	■	■	■	■							
D						■	■	■	■	■	■	■	■	■	■	■	■	■		
E									■	■					■	■	■	■	■	■
LEYENDA																				
■	EN EJECUCIÓN																			
■	PREPARADO																			

Tabla 2.2. Ejemplo de algoritmo FCFS.

### 3 Gestión de los recursos de un sistema operativo – Documentos CEO

#### C. Algoritmo STR (*Short Time Remainder*).

Este algoritmo permite asignar el tiempo de ejecución de forma prioritaria a los procesos más cortos con el fin de ejecutarlos en el menor tiempo posible. Así, si se está ejecutando un proceso y llega otro, independientemente de la duración que tenga el nuevo, interrumpirá el que esté en ejecución y el siguiente en consumir ciclos de UCP será el más corto de los que esperan en la cola (Tabla 2.3).

**EJEMPLO STR.**

PROCESO	CICLO DE LLEGADA	CICLOS TOTALES DE UCP	CICLO INICIAL	CICLO FINAL
A	0	3	1	3
B	2	6	4	9
C	4	4	12	15
D	6	5	16	20
E	8	2	10	11

ASIGNACIÓN DE CICLOS DE UCP A LOS PROCESOS SEGÚN EL ALGORITMO DE PLANIFICACIÓN ROUND ROBIN O DE RUEDA																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A																					
B																					
C																					
D																					
E																					
LEYENDA																					
	EN EJECUCION																				
	PREPARADO																				

**Tabla 2.3.** Ejemplo de algoritmo STR.

### 3 Gestión de los recursos de un sistema operativo – Documentos CEO

#### D. Algoritmo SRTF (*Shortest Remaining Time First*).

Variedad del STR donde la asignación de ciclos de UCP va en función del proceso que tenga menos ciclos pendientes de terminar. Cuando llega un nuevo proceso, se estiman los ciclos que le quedan tanto al proceso en ejecución como a los que llegan después. De la cola, terminará el que tenga menos ciclos pendientes de terminar y así sucesivamente hasta finalizar con todos los procesos planificados (véase Tabla 2.4.)

##### EJEMPLO SRTF.

PROCESO	CICLO DE LLEGADA	CICLOS TOTALES DE UCP	CICLO INICIAL	CICLO FINAL
A	0	3	1	3
B	2	6	4	15
C	4	4	5	8
D	6	5	16	20
E	8	2	9	10

ASIGNACIÓN DE CICLOS DE UCP A LOS PROCESOS SEGÚN EL ALGORITMO DE PLANIFICACIÓN ROUND ROBIN O DE RUEDA																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	■	■	■																	
B		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
C				■	■	■	■	■	■	■	■									
D						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
E								■	■	■	■									
LEYENDA																				
■	EN EJECUCION																			
■	PREPARADO																			

Tabla 2.4. Ejemplo de algoritmo SRTF.