**SISTEMAS INFORMÁTICOS**

**TAREA 4.1**

**ALUMNO:** Alejandro García de la Cruz **CURSO:** 1º DAM E-Learning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Enunciado.** | | | | |
| Disponemos de 5 peticiones de CPU para los procesos A, B, C , D y E, cuyos tiempos de CPU y momentos de creación se muestran en la siguiente tabla: | | | | |
| **PROCESO** | | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | | 0 | 3 | 2 |  |  |
| B | | 1 | 6 | 1 |  |  |
| C | | 3 | 3 | 3 |  |  |
| D | | 9 | 5 | 2 |  |  |
| E | | 11 | 5 | 1 |  |  |
| **Promedio:** | | | | |  |  |

El Cronograma se hará para cada algoritmo en una tabla igual a la siguiente. En este caso, por ejemplo, para el algoritmo Round-Robin Q=4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ROUND-ROBIN  Q=4 | | | | | | | | | | | | | | | | | | | | |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | .... |  |  |  |  |  |  |

 Se rellenarán únicamente las casillas en las que el proceso esté ejecutándose, poniendo R (Running), y cuando esté en estado E (listo, en espera).

Además se rellenarán, para cada algoritmo, los tiempos de espera, de retorno y el promedio de ambos tiempos, para cada proceso

Hacer el diagrama de ocupación de CPU, utilizando los siguientes algoritmos:

1. Round-Robin, con Q=4
2. Round-Robin, con Q=3
3. Por Prioridad. Para este algoritmo, se tendrá en cuenta la columna PRIORIDAD. Se considera que 1 es prioridad máxima.
4. El más corto primero (SJN)
5. El primero en llegar, primero en ejecutarse (FIFO)
6. El tiempo restante más corto (SRTN)

**1.- Round-Robin, con Q=4**

El método Round-Robin, es un método para seleccionar todos los elementos de un grupo de manera equitativa y en un orden racional, normalmente comenzado por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

Cada proceso tiene asignado un intervalo de tiempo de ejecución, llamado CUANTUM o CUANTO (en este ejemplo, CUANTUM = CUANTO = Q = 4).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 0 | 3 |
| B | 1 | 6 | 1 | 13 | 6 + 13 = 18 |
| C | 3 | 3 | 3 | 4 | 3 + 4 = 7 |
| D | 9 | 5 | 2 | 7 | 5 + 7 = 12 |
| E | 11 | 5 | 1 | 6 | 5 + 6 = 11 |
| **Promedio:** | | | | 30/5 = 6 | 51/5 = 10.2 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROUND-ROBIN  Q=4** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | E | E | R | R | R | R | E | E | E | E | E | E | E | E | E | E | E | R | R |  |  |  |  |
| C |  |  |  | E | E | E | E | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  | E | R | R | R | R | E | E | E | E | E | E |  | R |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | E | E | E | R | R | R | R | E | E |  | E | R |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |

**2.- Round-Robin, con Q=3.**

En este caso, la planificación se resuelve con el mismo método (Round-Robin) pero con Q=3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 0 | 3 |
| B | 1 | 6 | 1 | 11 | 6 + 11 =17 |
| C | 3 | 3 | 3 | 3 | 3 + 3 = 6 |
| D | 9 | 5 | 2 | 6 | 5 + 6 = 11 |
| E | 11 | 5 | 1 | 6 | 5 + 6 = 11 |
| **Promedio:** | | | | 26/5 = 5.2 | 48/5 = 9.6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROUND-ROBIN  Q=3** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | E | E | R | R | R | E | E | E | E | E | E | E | E | E | R | R | R |  |  |  |  |  |  |
| C |  |  |  | E | E | E | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  | R | R | R | E | E | E | E | E | E | R | R |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | E | R | R | R | E | E | E | E | E |  | R | R |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |

**3.- Por Prioridad. Para este algoritmo, se tendrá en cuenta la columna PRIORIDAD. Se considera que 1 es prioridad máxima.**

En este algoritmo, tal y como nos pide el ejercicio, planificaremos en función de la prioridad, es decir, **PRIORIDAD 1 = MÁS PRIORITARIO** y así sucesivamente.

Si un proceso coincide con otro que sea más prioritario (valor numérico más bajo), éste primer proceso tendrá que esperar (aun habiendo llegado antes), y el recién llegado obtendrá la actividad de CPU hasta ser finalizado su proceso. Este método es de tipo **APROPIATIVO.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 6 | 3 + 6 = 9 |
| B | 1 | 6 | 1 | 0 | 6 + 0 = 6 |
| C | 3 | 3 | 3 | 16 | 3 + 16 = 19 |
| D | 9 | 5 | 2 | 5 | 5 + 5 = 10 |
| E | 11 | 5 | 1 | 0 | 5 + 0 = 5 |
| **Promedio:** | | | | 27/5=5.4 | 49/5=9.8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **POR PRIORIDAD** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | E | E | E | E | E | E | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | R | R | R | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E | R |  | R | R |  |
| D |  |  |  |  |  |  |  |  |  | R | R | E | E | E | E | E | R | R | R |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | R | R | R | R | R |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |

(CORREGIDO)

**4.- El más corto primero (SJN)**

Este método normalmente es **NO APROPIATIVO**, es decir, una vez que empiece tiene que finalizar el proceso totalmente, y en este algoritmo, tienen prioridad los que ocupan menos tiempo, es decir, los más cortos **“SHORTER JOB FIRST” (SJF** en inglés**)** .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 0 | 3 + 0 = 3 |
| B | 1 | 6 | 1 | 5 | 6 + 5 = 11 |
| C | 3 | 3 | 3 | 0 | 3 + 0 = 3 |
| D | 9 | 5 | 2 | 3 | 5 + 3 = 8 |
| E | 11 | 5 | 1 | 6 | 5 + 6 = 11 |
| **Promedio:** | | | | 14/5 = 2.8 | 36/5 = 7.2 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SJN** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | E | E | E | E | E | R | R | R | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  | E | E | E | R | R | R | R | R |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | E | E | E | E | E | E | R | R | R | R | R | R |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |

Nota: He consultado una página web en la que dice que puede existir el sistema APROPIATIVO, es decir, CON DESALOJO, esto es:

“Si se incorpora un nuevo proceso a la cola de listos y éste tiene un ciclo de CPU menor que el ciclo de CPU del proceso que se está ejecutando, entonces dicho proceso es desalojado y el nuevo proceso toma la CPU.”

https://www.udg.co.cu/cmap/sistemas\_operativos/planificacion\_cpu/sjf/sjf.html

**5.- El primero en llegar, primero en ejecutarse (FIFO)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 0 | 3 + 0 = 3 |
| B | 1 | 6 | 1 | 2 | 6 + 2 = 8 |
| C | 3 | 3 | 3 | 6 | 3 + 6 = 9 |
| D | 9 | 5 | 2 | 3 | 5 + 3 = 8 |
| E | 11 | 5 | 1 | 6 | 5 + 6 = 11 |
| **Promedio:** | | | | 17/5=3.4 | 39/5 = 7.8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FIFO: FIRST INSIDE, FIRST OUTSIDE** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | E | E | R | R | R | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  | E | E | E | E | E | E | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  | E | E | E | R | R | R | R | R |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | E | E | E | E | E | E | R | R | R | R | R | R |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |

**6.-** **El tiempo restante más corto (SRTN)**

Con este método, el proceso al que le quede menos tiempo para ser resuelto, es el que tiene prioridad (este algoritmo se suele usar para beneficiar un poco a los que necesitan más tiempo de CPU, pero llegaron antes que muchos de los demás procesos).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **MOMENTO DE CREACIÓN** | **TIEMPO DE CPU** | **PRIORIDAD**  **(solo para algoritmo Por Prioridad)** | **Tiempo de Espera** | **Tiempo**  **de Retorno** |
| A | 0 | 3 | 2 | 0 | 3 + 0 = 3 |
| B | 1 | 6 | 1 | 5 | 6 + 5 = 11 |
| C | 3 | 3 | 3 | 0 | 3 + 0 = 3 |
| D | 9 | 5 | 2 | 3 | 5 + 3 = 8 |
| E | 11 | 5 | 1 | 6 | 5 + 6 = 11 |
| **Promedio:** | | | | 14/5=2.8 | 36/5 = 7.2 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SRTN** | | | | | | | | | | | | | | | | | | | | | | | | |
| A | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | E | E | E | E | E | R | R | R | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  | R | R | R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  | E | E | E | R | R | R | R | R |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  | E | E | E | E | E | E | R | R | R |  | R | R |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  | 20 | 21 | 22 |