**GUIA DE EJERCICIOS**

***NOMENCLATURA QUE ENCONTRARAN EN LOS EJECICIOS***

**A** = En qué momento arriba el proceso a la ready queue.

**CPU** = tiempo de ejecución dentro de la cpu (medido en ráfaga, porque pueden ser varias), esto también estará supeditado a la arquitectura.

**I/O, E/S** **o W**= *input/output, entrada/salida o Wait*, pueden estar numerados, representan la cantidad de tiempo y los diferentes tipos de entradas salidas ya que puede haber más de una.

**RQ** = Ready Queue o Cola de Listos

1. Resuelva el siguiente ejercicio en un Gantt con el algoritmo **FIFO**, se sabe que la CPU esta vacía y el ejercicio comienza en Tiempo cero.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proceso** | **A** | **CPU** | **I/O** | **CPU** |
| A | 0 | 5 | 5 | 5 |
| B | 0 | 10 | 10 | 5 |
| C | 0 | 5 | - | - |
| D | 0 | 10 | 5 | 5 |

***TIPS***

*“A tener en cuenta, cuando todos los procesos arriban al mismo tiempo resolveremos su ordenamiento en la READY QUEUE por orden alfabético.”*

*“Una vez que un proceso entre en el ciclo de vida y empate en su llegada a la READY QUEUE se definirá ese empate por FIFO ósea el que mas tiempo haya estado en el sistema.”*

1. Resuelva el siguiente ejercicio en un Gantt con el algoritmo **FIFO**, se sabe que la CPU está vacía y el ejercicio comienza en Tiempo cero.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proceso** | **A** | **CPU** | **I/O** | **CPU** |
| A | 5 | 5 | 5 | 5 |
| B | 2 | 10 | 10 | 5 |
| C | 0 | 5 | - | - |
| D | 4 | 10 | 5 | 5 |

1. Resuelva los dos ejercicios anteriores con el algoritmo **SJF y SPN,** saque sus conclusiones entre los dos “Shortest”.
2. Resuelva el siguiente ejercicio con los tres algoritmos Apropiativos (***FIFO, SJF y SPN***)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proceso** | **A** | **CPU** | **I/O1** | **I/O2** | **CPU** |
| A | 5 | 5 | - | 5 | 4 |
| B | 2 | 10 | 10 | - | 5 |
| C | 0 | 5 | - | 20 | 5 |
| D | 4 | 10 | - | 15 | 5 |
| F | 1 | 4 | 5 | - | 6 |
| G | 3 | 2 | 10 | - | 7 |

1. Resuelve el siguiente ejercicio de Expulsivo utilizando el algoritmo ***Round Robin “RR=5”.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proceso** | **A** | **CPU** | **I/O1** | **I/O2** | **CPU** |
| A | 5 | 15 | - | 5 | 10 |
| B | 0 | 10 | 10 | - | 5 |
| C | 6 | 5 | - | 20 | 10 |
| D | 3 | 10 | - | 15 | 15 |
| F | 2 | 5 | 5 | - | 5 |

1. Resolvamos el ejercicio 5 pero con el algoritmo **STR**.
2. Resolvamos este ejercicio de 2 núcleos sabiendo que la CPU1 posee un Quantum de RR=2 la CPU2 posee un Quantum de RR=4, cada CPU posee su READY QUEUE correspondiente. Todos los procesos arriban a la READY 1, luego va bajando según su tiempo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proceso** | **A** | **CPU** | **I/O1** | **I/O2** | **CPU** |
| A | 5 | 12 | - | 5 | 10 |
| B | 0 | 10 | 10 | - | 4 |
| C | 6 | 8 | - | 20 | 10 |
| D | 3 | 10 | - | 15 | 8 |
| F | 2 | 4 | - | - | - |



A tener en cuenta según el esquema o arquitectura que poseemos cuando un proceso no sale a I/O y se vence su Quantum en CPU2 el mismo retorna a la READY QUEUE 2, solo se retomara a la CPU1 cuando se vuelva de una I/O.

**Tips:**

*“Traten de llevar el orden de cada RQ, porque cuantas mas CPUs tenga el ejercicio mas complicado es seguirlo. El orden es la base de todo.”*

1. Ejercicio de Hilos, las características son las siguientes, se posee solo una CPU la cual maneja una planificación de RR=4, algunos procesos manejan hilos con una Biblioteca SFJ, con el siguiente diagrama:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROCESO** | **ARRIBO** | **CPU** | **I/O1** | **I/O2** | **CPU** |
| A | 0 | 2 | - | - | 2 |
| ULT1.1 |  | 6 | 4 | - | 2 |
| ULT1.2 |  | 8 | - | 4 | 2 |
| B | 3 | 4 | - | 2 | 2 |
| C | 1 | 2 | 3 | - | 2 |
| KLT | 2 | 5 | - | 2 | 1 |

1. Ahora utilicemos el esquema anterior del ejercicio 8 pero la Biblioteca de hilos será RR=2 y la CPU mantendrá el mismo esquema RR=4