Trabajo Práctico N°4

**Planificación de la CPU**

1. Explique:
   1. La diferencia entre algoritmos “preemptive” y “non-preemptive”.

Los algoritmos non-preemptive o no apropiativos son aquellos en los que el proceso abandona el CPU voluntariamente. Mientras que en los preemptive o apropiativos son aquellos en que el núcleo del sistema operativo maneja el uso del CPU por parte del proceso.

* 1. ¿Cuál de los dos tipos provoca mayor “overhead” en el sistema y por qué?

Overhead: procesamiento que no es útil.

El algoritmo que genera mayor overhead son los apropiativos ya que como el SO toma las decisiones sobre el manejo del CPU por parte de los procesos, los cambios de procesos serán más frecuentes y son tareas del sistema operativo que generan sobrecargas en el sistema.

* 1. ¿Qué técnicas de planificación pueden degenerar en FCFS? ¿En qué condiciones?

La técnica Round Robin puede simular la técnica FCFS si el quantum otorgado es grande ya que le dará tiempo a los procesos de terminar, lo que se acercará al FCFS.

Si en SJF los procesos tienen el mismo tiempo de ejecución se ejecutará como FCFS.

En el caso del prioritario si lo procesos tuviesen igual prioridad también se convertirá en un FCFS.

1. El esquema de planificación basado en colas multinivel con realimentación visto en la teoría, normalmente se implementa aumentado el quantum a medida que baja la prioridad. ¿Qué efecto en los procesos se busca conseguir que es similar al producido por uno los algoritmos vistos? Explique porqué.

Se busca generar un efecto parecido al SJF donde se favorece al proceso de menor duración ya que cuando se crea el proceso, éste se asigna a la cola de mayor prioridad y con un quantum determinado. Al estar en la cola de mayor prioridad accederá al rápidamente al uso del CPU. Como es un proceso de corta duración, tiene muchas chances de terminar de ejecutarse.

1. Considere una variante del algoritmo Round Robin, en donde los elementos de la cola Ready son punteros a los distintos PCB.
   1. ¿Cuál es el efecto de poner dos punteros al mismo PCB en dicha cola?

Poner dos punteros al mismo PCB (Process Control Block) en la cola significa que el mismo proceso recibiría el doble de tiempo de CPU en comparación con otros procesos que solo tienen un puntero.

En otras palabras, el proceso tendría más oportunidades de ser ejecutado.

* 1. ¿Cuál sería la principal ventaja y desventaja de este esquema?

Ventaja:

Puedes priorizar procesos sin necesidad de modificar el algoritmo básico de Round Robin ni agregar un sistema complejo de prioridades. Solo basta con duplicar (o triplicar) punteros.

Desventaja:

La cola puede crecer artificialmente, ocupando más memoria y complicando la contabilidad de los procesos (por ejemplo, medir el tiempo total de CPU que consumen se vuelve más difícil).

También podría introducir inequidad si se abusa de la duplicación.

* 1. ¿Cómo modificaría el algoritmo Round Robin tradicional para lograr el mismo efecto que este algoritmo?

Una manera sería asignar diferentes "quantum" de tiempo a los procesos según su prioridad.

En vez de duplicar punteros, simplemente podrías:

Dar un quantum más grande a los procesos más importantes o permitir que repetidamente se les asigne CPU sin ser enviados al final de la cola, dependiendo de su "peso".

1. Se podría decir que el algoritmo de colas multinivel con realimentación beneficia en cierta medida a los procesos limitados por E/S ¿Por qué?

Sí, el algoritmo de colas multinivel con realimentación beneficia a los procesos limitados por E/S.

Esto sucede porque un proceso que necesita realizar una operación de entrada/salida normalmente no utiliza todo su quantum de CPU, ya que se bloquea rápidamente para esperar el recurso.

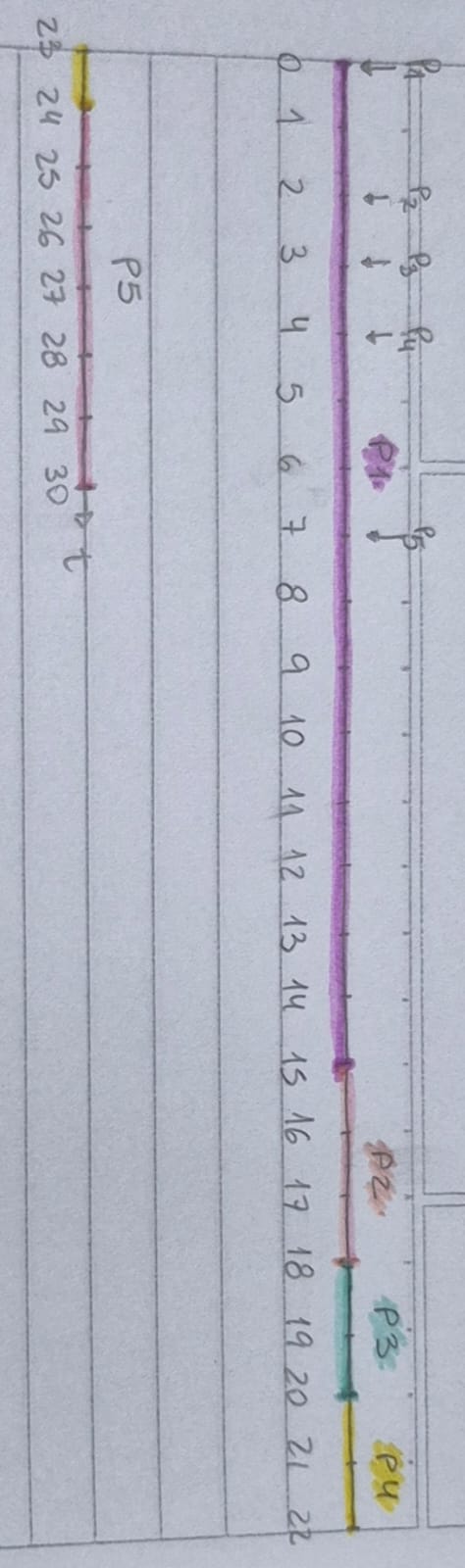
Al no consumir todo su tiempo asignado, el proceso se mantiene en una cola de alta prioridad.

Cuando la operación de E/S termina y el proceso vuelve al estado Ready, retoma su lugar en una cola de alta prioridad, obteniendo acceso al CPU antes que otros procesos más pesados que ya fueron degradados a colas de menor prioridad.

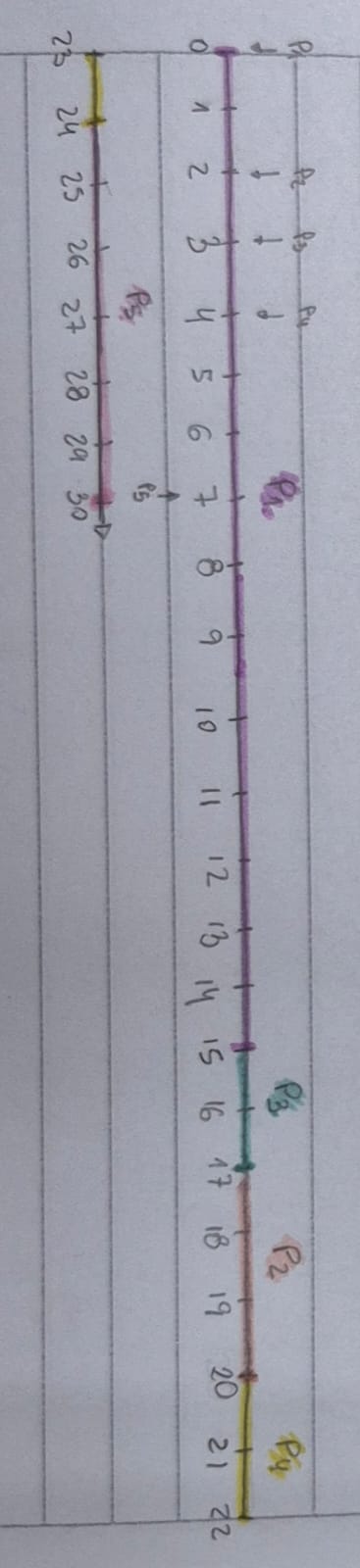
Esto permite que los procesos de E/S reciban atención más rápida, mejorando su tiempo de respuesta y favoreciendo el rendimiento global del sistema.

1. Suponga cinco procesos que ingresan al sistema en los tiempos 0, 2, 3, 4 y 7, y que sus tiempos de servicio (ejecución) son 15, 3, 2, 4 y 6, respectivamente.
   1. Represente en un diagrama de tiempo, los 5 procesos ejecutándose, utilizando los algoritmos FCFS, Round Robin y SJF no apropiativo. Utilice para el algoritmo de Round Robin un quantum igual al menor tiempo de servicio.

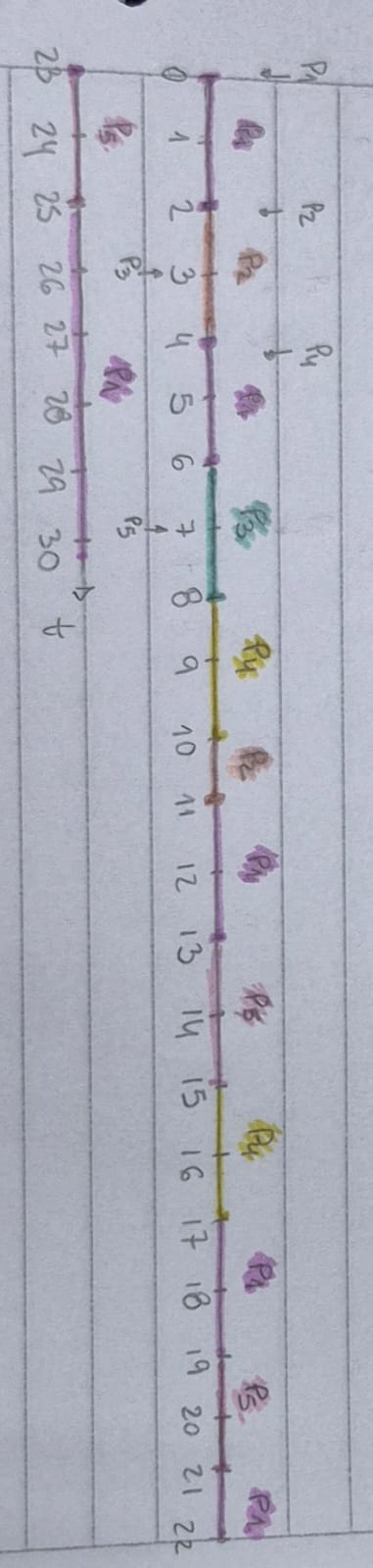
FCFS



SJF no apropiativo



Round Robin



* 1. Calcule el tiempo de ejecución (desde que llega hasta que finaliza) de cada uno de los 5 procesos y luego el promedio de los mismos para los algoritmos mencionados.

Para FCFS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 15 | 16 | 17 | 20 | 23 | 18.2 |

Para SJF no apropiativo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 15 | 18 | 14 | 20 | 23 | 18 |

Para round robin:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 30 | 9 | 5 | 13 | 18 | 15 |

* 1. Calcule el tiempo de espera de cada uno de los 5 procesos y luego el promedio de los mismos para los algoritmos mencionados.

Para FCFS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 0 | 13 | 15 | 16 | 17 | 12.2 |

Para SJF no apropiativo:

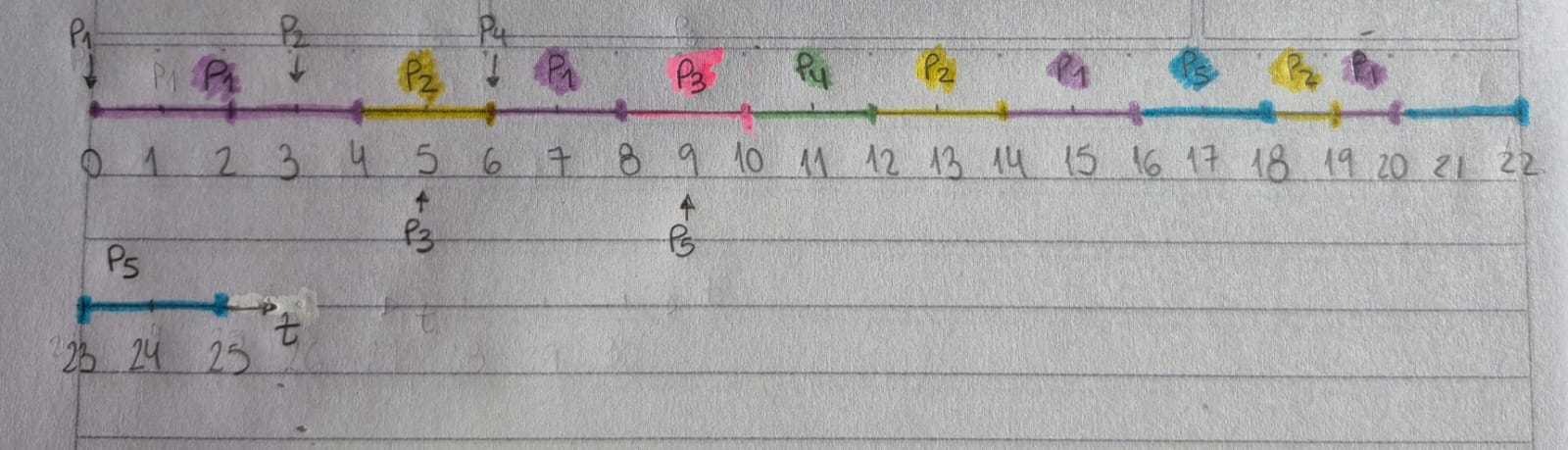
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 0 | 15 | 12 | 16 | 17 | 12 |

Para Round Robin:

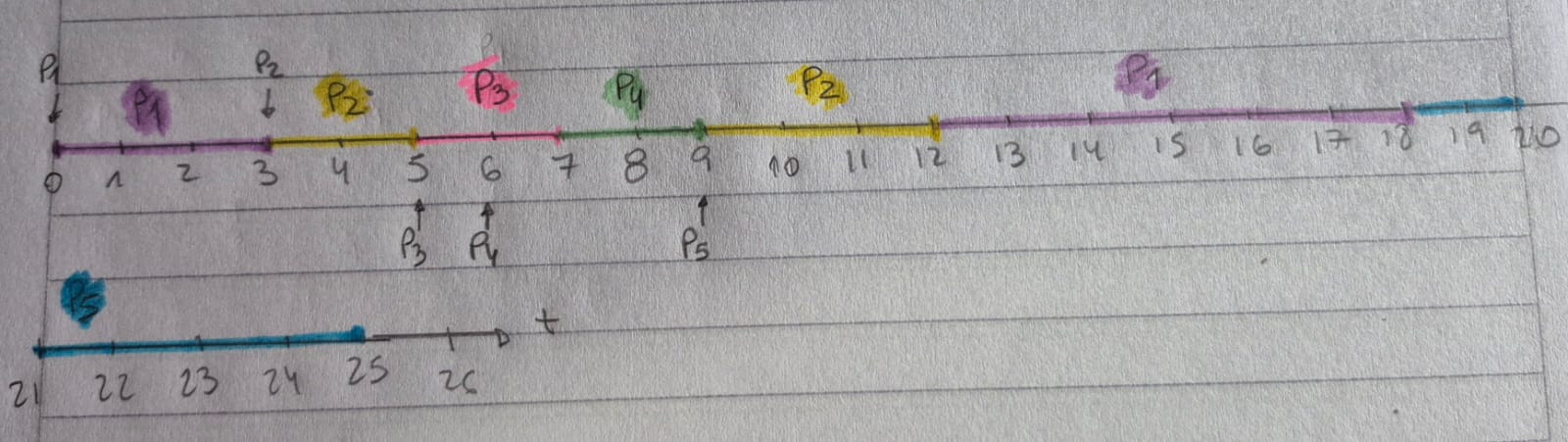
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 15 | 6 | 3 | 11 | 12 | 9.4 |

1. Considere que cinco procesos ingresan al sistema en los instantes de tiempo 0, 3, 5, 6 y 9, y que sus tiempos de ejecución son los siguientes: 9, 5, 2, 2 y 7, respectivamente.
   1. Represente en un diagrama de tiempo los cinco procesos ejecutándose, utilizando los algoritmos Round Robin y SJF apropiativo. Para el algoritmo de Round Robin utilice un quantum igual al menor tiempo de ejecución.

Round Robin



SJF apropiativo



* 1. Calcule el tiempo de ejecución (desde que llega hasta que finaliza) de cada uno de los 5 procesos y luego el promedio de los mismos para los algoritmos mencionados.

Round Robin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 20 | 16 | 5 | 6 | 16 | 12.6 |

SJF apropiativo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 18 | 9 | 2 | 3 | 16 | 9.6 |

* 1. Calcule el tiempo de espera de cada uno de los 5 procesos y luego el promedio de los mismos para los algoritmos mencionados.

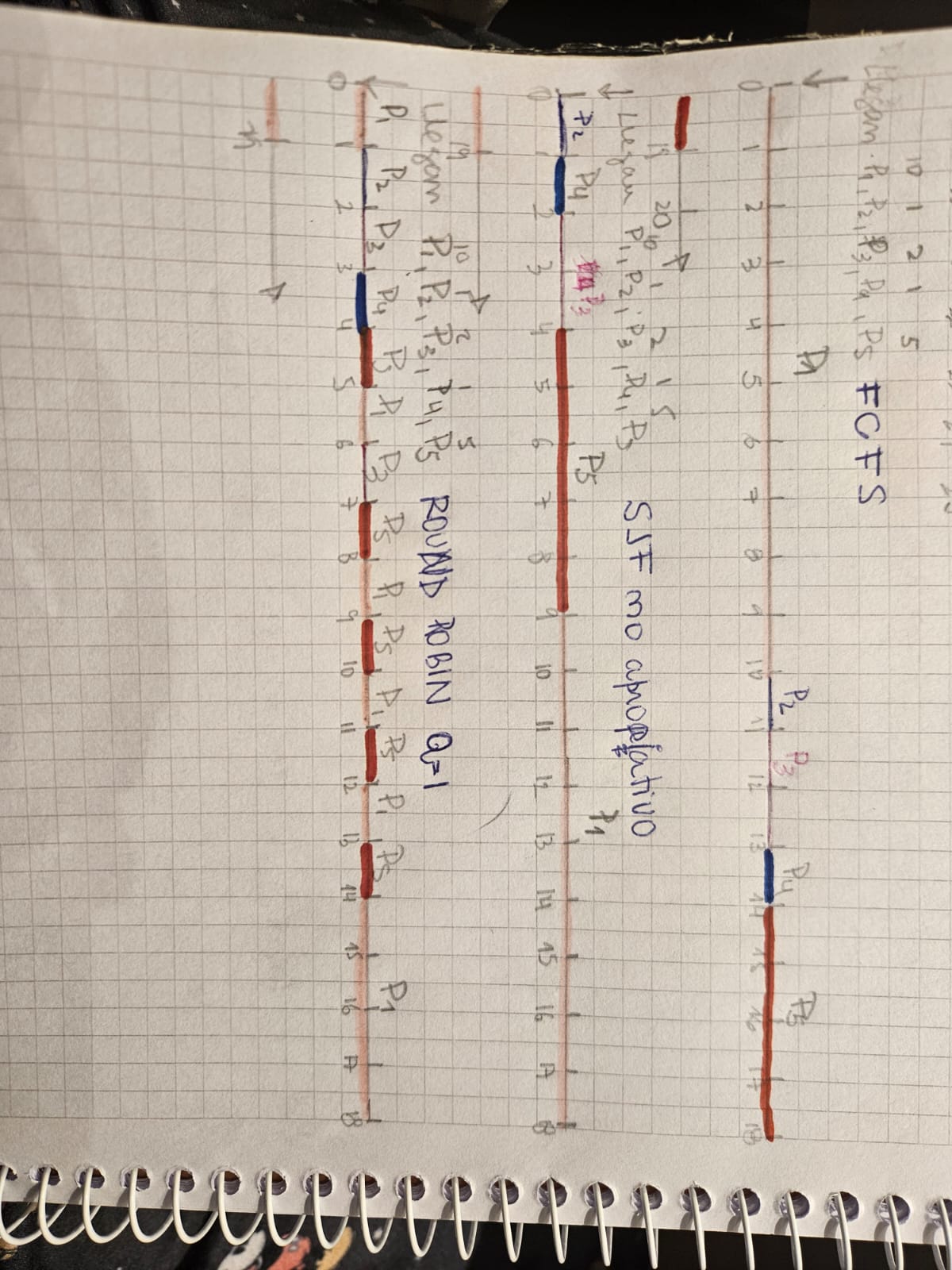
Round Robin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 11 | 11 | 3 | 4 | 9 | 7.6 |

SJF apropiativo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 9 | 4 | 0 | 1 | 11 | 5 |

1. Suponga que la cola de procesos preparados para ejecutar en la CPU contiene 5 procesos, siendo sus tiempos totales de ejecución 10, 1, 2, 1 y 5.
   1. Para cada uno de los algoritmos de planificación FCFS, Round Robin (quantum = 1) y SJF no apropiativo, represente el diagrama de ejecución en el tiempo.



* 1. Asuma que los procesos llegaron al mismo tiempo, pero fueron colocados en la cola en el orden mencionado. Calcule el tiempo de ejecución promedio.

Para FCFS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 10 | 11 | 13 | 14 | 19 | 13.4 |

Para SJF no apropiativo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 19 | 1 | 4 | 2 | 9 | 7 |

Para Round Robin:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 19 | 2 | 7 | 4 | 14 | 9.2 |

* 1. ¿Cuál de los algoritmos brinda el menor tiempo de espera promedio?

Para FCFS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 0 | 10 | 11 | 13 | 14 | 9.6 |

Para SJF no apropiativo:

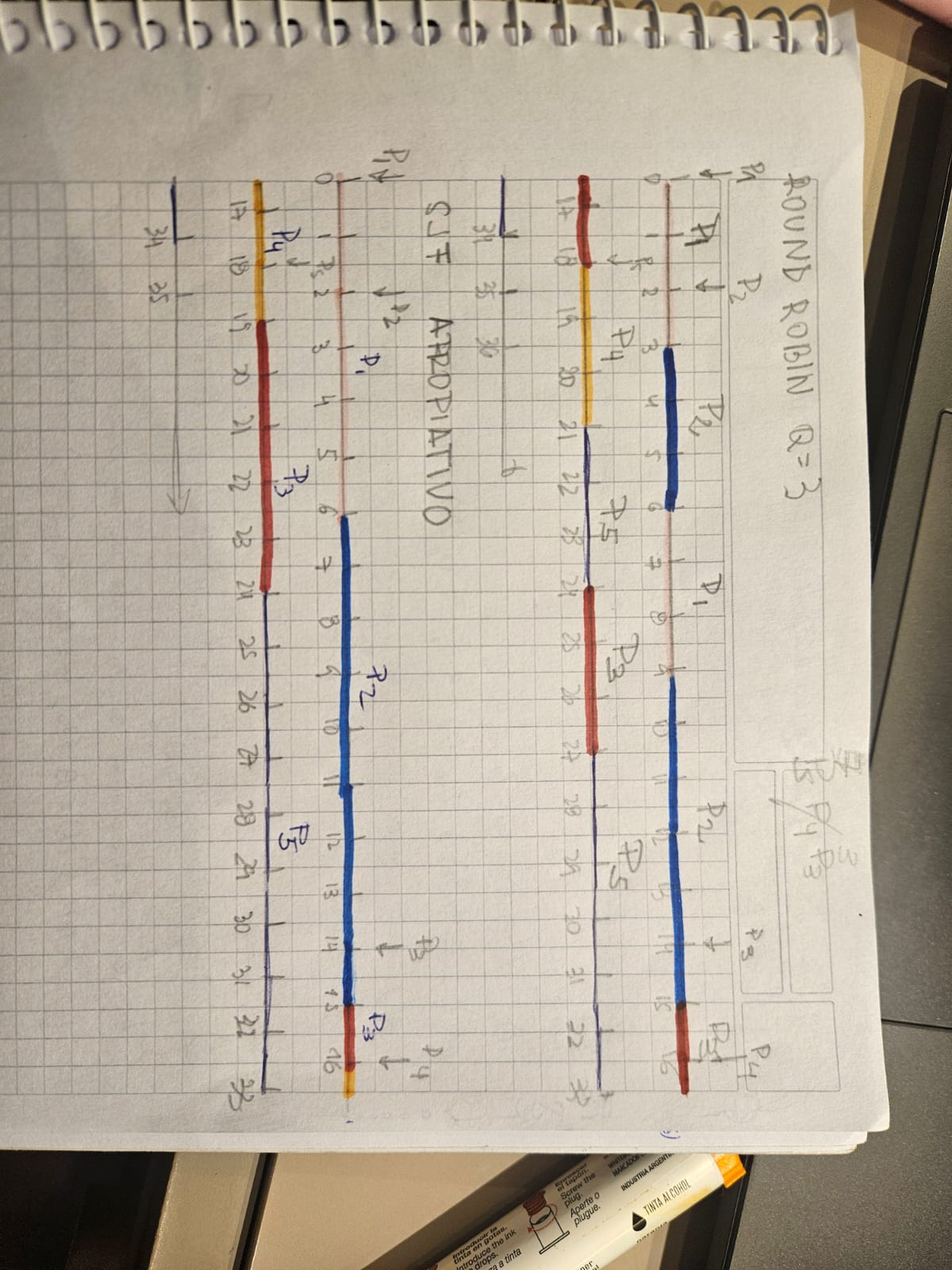
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 9 | 0 | 2 | 1 | 4 | 3.2 |

Para Round Robin:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 9 | 1 | 5 | 3 | 9 | 5.4 |

Menor tiempo de espera promedio: SJF no apropiativo.

1. Considere que cinco procesos ingresan al sistema en los instantes de tiempo 0, 2, 14, 16 y 18, y que sus tiempos de ejecución son los siguientes: 6, 9, 6, 3 y 10, respectivamente.
   1. Represente en un diagrama de tiempo los cinco procesos ejecutándose, utilizando los algoritmos *Round Robin* y *SJF* apropiativo. Para el algoritmo de *Round Robin* utilice un quantum *q = 3*.



* 1. Calcule el tiempo de ejecución (desde que llega hasta que finaliza) de cada uno de los procesos y luego el promedio de los mismos para los algoritmos mencionados.

Round Robin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 9 | 13 | 13 | 5 | 16 | 11.2 |

SJF apropiativo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 6 | 13 | 10 | 3 | 16 | 9.6 |

* 1. Calcule el tiempo de espera de cada uno de los procesos y luego el promedio de los mismos para los algoritmos mencionados.

Round Robin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 3 | 4 | 7 | 2 | 6 | 4.4 |

SJF apropiativo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | Promedio |
| 0 | 4 | 4 | 0 | 6 | 2.8 |