Entrega 4

Sistemas Operativos II

Mellino, Natalia Farizano, Juan Ignacio

Ejercicio 1

- **Ronda egoísta (SRR):** favorece a los procesos que ya han pasado tiempo ejecutando antes que a los recién llegados. Los nuevos procesos se forman en la cola de *procesos nuevos* y se avanza únicamente con la cola de *procesos aceptados*. Para SRR se emplean los parámetros a y b donde a indica el ritmo al cual se incrementa la prioridad de los procesos en la cola de *procesos nuevos* y b, el ritmo de incremento de prioridad para los *procesos aceptados*. Cuando la prioridad de un proceso nuevo alcanza la prioridad de un proceso aceptado, el nuevo se vuelve aceptado. Si la cola de aceptados queda vacía, se acepta el proceso nuevo con mayor prioridad.
- **Multicolas con prioridad:** tenemos una cantidad arbitraria de colas que tienen distintas prioridades. En cada cola tenemos a su vez procesos con prioridades. El planificador elegirá únicamente entre los procesos que estén formados en la cola de mayor prioridad. Sólo cuando estos procesos terminen (o sean enviados a alguna otra cola), el planificador continuará con aquellos que estén en las siguientes colas.

Ambas políticas serán iguales en el caso que tengamos procesos cortos ya que en la política de multicolas con dos clases de prioridad estos procesos terminarán sus tareas sin haber sido degradados a la segunda cola de prioridad. Entonces, si pensamos esto en términos de SRR, la cola de mayor prioridad en la política de multicolas sería la cola de *procesos aceptados* en SRR y la cola de menor prioridad sería la de *procesos nuevos*. Por ello, al ser cortos los procesos, estos se ejecutarán por completo en la cola de mayor prioridad haciendo que esta actúe como la cola de *procesos aceptados*.

Ejercicio 2

Aging:

- **Ventajas:** resulta favorecedor para los procesos cortos ya que pueden terminar sus tareas sin ser degradados.
- **Desventajas:** los únicos procesos beneficiados son los recién llegados, mientras que los procesos más largos son castigados e incluso podrían llegar a sufrir inanición.

Retroalimentación inversa:

- **Ventajas:** el promover un proceso a una cola de más alta prioridad permite evitar que dicho proceso sufra inanición.
- **Desventajas:** si la degradación es menor a la retroalimentación inversa, de cierta forma se favorecen a aquellos procesos que acaparan el procesador por mucho tiempo.

Ejercicio 3

Ejecutamos el comando time utilizando el simulador del Juego de la Vida de Conway realizado en Sistemas Operativos I como trabajo práctico.

```
>> time ./simulador entrada.game

real    1m7,990s
user    1m7,642s
sys     0m0,144s
```

Podemos ver que el tiempo de uso del procesador es de 1m7,786s que provienen de sumar el tiempo de usuario y el tiempo de sistema. A partir de estos tiempos, podemos calcular que el porcentaje de utilización del CPU de este proceso fue aproximadamente de 99.69%.
El tiempo desocupado es nulo ya que un proceso se está ejecutando y el procesador nunca se encuentra desocupado en el periodo de tiempo medido. Por lo tanto, al restar el tiempo de uso del procesador del tiempo real obtenemos el tiempo de núcleo, que resultar ser de 0.204s.
Los tiempos pueden variar según la cantidad de procesos que tienen que ejecutarse cuando corremos nuestro programa. Si la cantidad es baja o nula, nuestro programa podrá acaparar por más tiempo el procesador, de esta forma el porcentaje de utilización aumenta llegando casi al 100% como se ve en nuestro ejemplo, por lo tanto no son necesarios muchos cambios de contexto y el tiempo de núcleo disminuye y el tiempo desocupado resulta ser nulo. Si tenemos una mayor cantidad de procesos que necesitan utilizar el procesador, el porcentaje de utilización para nuestro proceso disminuirá, el tiempo de uso será el mismo pero el tiempo real será mayor, a su vez, al tener que realizarse una mayor cantidad de cambios de contexto el tiempo de núcleo aumentará, el tiempo desocupado seguirá siendo nulo ya que hay procesos para ejecutar.