

# 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

```
Command being timed: "make"
  User time (seconds): 6.41
  System time (seconds): 1.91
  Percent of CPU this job got: 99%
  Elapsed (wall clock) time (h:mm:ss or m:ss): 0:08.32
  Voluntary context switches: 5319
  Involuntary context switches: 1655
```