



## Entrega 4 - SOII

Juan Cruz De La Torre, Bautista Marelli, Francisco Alcacer

June 8, 2021

### 1 Ejercicio 1

En efecto, una política de multicolos con dos colas de prioridad y una ronda egoísta resultan similares al comparar la cola de mayor prioridad de la multicola con la cola de procesos aceptados, y la de menor prioridad con la de procesos nuevos. Sin embargo, estas políticas se comportarán de manera diferente dependiendo de los valores que tomen los parámetros  $a$  y  $b$  de la ronda egoísta. Si  $a > 0$ , inevitablemente sucederá que procesos de la cola de procesos nuevos eventualmente pasarán a la de procesos aceptados, probablemente siendo ejecutados antes que un proceso de mayor prioridad que llegó más tarde. En cambio, si  $a = 0$ , sucederá que solo podrán ejecutarse procesos de menor prioridad cuando no haya otros de mayor prioridad, de manera equivalente a lo que sucede con la multicola.

### 2 Ejercicio 2

Sabemos que aging consiste en degradar procesos que recibieron mucha atención y la retroalimentación inversa trata sobre promover procesos que recibieron

menor atención. Estas políticas se encuentran dentro de la retroalimentación de multinivel con feedback, en distinción de las multicolos con prioridad sin estos mecanismos. Una de las ventajas de estas mejoras, es la resolución de la hambruna de los procesos ya que con estas estrategias los procesos se pueden mover entre las diferentes colas, garantizando que serán atendidos eventualmente. Una desventaja es que estos algoritmos son más complejos al requerir más atención a los parámetros que determinan la promoción o degradación de procesos, así como también que pueden generar como consecuencia muchos procesos en una cola y pocos en otra, perjudicando el desempeño general de las multicolos.

### 3 Ejercicio 3

Para este ejercicio, ejecutamos un caso de prueba de un trabajo práctico desarrollado en Estructuras de Datos y Algoritmos I, que consiste en un corrector ortográfico que toma un listado de palabras válidas y un texto, y busca errores ortográficos y sugiere alternativas. Ejecutando *time ./main listado-general.txt texto.txt out.txt* obtuvimos como respuesta:

```
Diccionario listado-general.txt cargado exitosamente.  
CARGA DE DICCIONARIO: 0.195850s  
CORRECCIÓN DEL TEXTO: 10.357246s  
CANTIDAD DE PALABRAS LEÍDAS: 13038  
CANTIDAD DE PALABRAS INCORRECTAS: 4512
```

```
real    0m10.591s  
user    0m10.400s  
sys     0m0.191s
```

A partir de esa información, podemos decir ciertas cosas sobre los tiempos de ejecución del proceso. En primer lugar podemos ver que el tiempo de usuario es 10.4s, el de sistema es 0.191s y el real es de 10.591s. A partir de esos valores, podemos decir varias cosas más.

El tiempo de uso del procesador es la suma del tiempo de sistema y el tiempo de usuario, siendo en este caso 10.591s. Esto es, el porcentaje de utilización del CPU es de 100% aproximadamente.

En este caso, el proceso no pasa tiempo desocupado y el tiempo de núcleo es el mismo que el tiempo de sistema o la diferencia es insignificante, debido a que posiblemente no se hayan realizado demasiados cambios de contexto durante la ejecución del mismo. El tiempo de núcleo, es como mínimo el

tiempo de sistema y como máximo la diferencia entre el tiempo real y el tiempo de usuario.

- tiempo de núcleo:  $0.191s$
- tiempo de sistema:  $0.191s$
- tiempo de usuario:  $10.4s$
- tiempo de uso del procesador:  $10.591s$
- tiempo desocupado:  $0s$
- utilización del CPU:  $100\%$