

# **Informe del algoritmo MLFQ**

## **Parcial 1**

Pontificia Universidad Javeriana

Sistemas Operativos

Profesor: Jefferson Peña Torres

Estudiante: Daniel Felipe Moncada Tello

19 de Septiembre 2025

# Funcionamiento del algoritmo MLFQ

El *Multi-Level Feedback Queue* (MLFQ) es un algoritmo de planificación de CPU que utiliza varias colas con diferente prioridad. Cada cola puede aplicar un algoritmo distinto (ejemplo: Round Robin, SJF o STCF) y con *quantums* de tiempo variables. Su característica principal es que los procesos cambian de cola dinámicamente dependiendo de su comportamiento de ejecución.

## Reglas básicas de funcionamiento

### Inicio del algoritmo

Todos los procesos entran en la cola de mayor prioridad (cola 1). Esta cola normalmente usa Round Robin con *quantum* pequeño, favoreciendo a procesos cortos o interactivos. La CPU siempre selecciona procesos de la cola de mayor prioridad que no esté vacía. Es decir, mientras haya procesos en una cola superior, no se atienden las colas inferiores.

### Manejo de los Quantum

Si un proceso agota su *quantum* sin terminar, se mueve a la siguiente cola (menor prioridad). Ejemplo: un proceso en la cola 1 con *quantum* = 2 que no termina, baja a la cola 2. Si un proceso termina dentro de su *quantum*, se libera sin bajar de cola. Las colas inferiores tienen *quantums* más largos o algoritmos diferentes. Así los procesos largos generan menos cambios de contexto.

### Colas distintas a RR

En colas con SJF (*Shortest Job First*) se ejecuta el proceso con la ráfaga más corta entre los disponibles. Con STCF (*Shortest Time to Completion First*) se ejecuta el proceso con la

ráfaga restante más corta, y si llega otro con menor tiempo restante, se interrumpe el actual (expropiativo).

## Comportamiento esperado del algoritmo

Para los procesos cortos, estos probablemente terminan en las primeras colas, obteniendo buena respuesta. Para los procesos largos, estos descienden progresivamente a colas con *quantums* más amplios o con algoritmos más eficientes para trabajos pesados.

## Cómo ejecutar el algoritmo implementado

La implementación del algoritmo MLFQ fue desarrollada en C++. El programa se compila normalmente con el compilador g++:

```
g++ main.cpp Process.cpp MLFQ.cpp
```

Para ejecutar el programa, los datos de entrada deben ser redirigidos desde un archivo de texto con el formato definido (ejemplo: PID;BT;AT;...). La salida también se redirige a un archivo de texto.

Ejemplo de uso:

```
a.exe < entrada.txt > salida.txt
```

Donde:

- `entrada.txt` contiene los procesos a simular.
- `salida.txt` recibirá los resultados de la ejecución, incluyendo tiempos de espera (WT), retorno (TAT), respuesta (RT) y finalización (CT).

## Ventajas y conclusión del algoritmo MLFQ

- **Flexible:** combina varios algoritmos en uno.
- **Justo:** los procesos cortos son atendidos rápido, los largos siguen teniendo CPU en colas inferiores.
- **Adaptativo:** ajusta la prioridad de los procesos según cómo se comportan.

El MLFQ combina lo mejor de varios algoritmos de planificación y se adapta en tiempo real, lo que lo convierte en una estrategia potente para sistemas operativos multitarea, aunque a costa de mayor complejidad en su implementación y configuración.