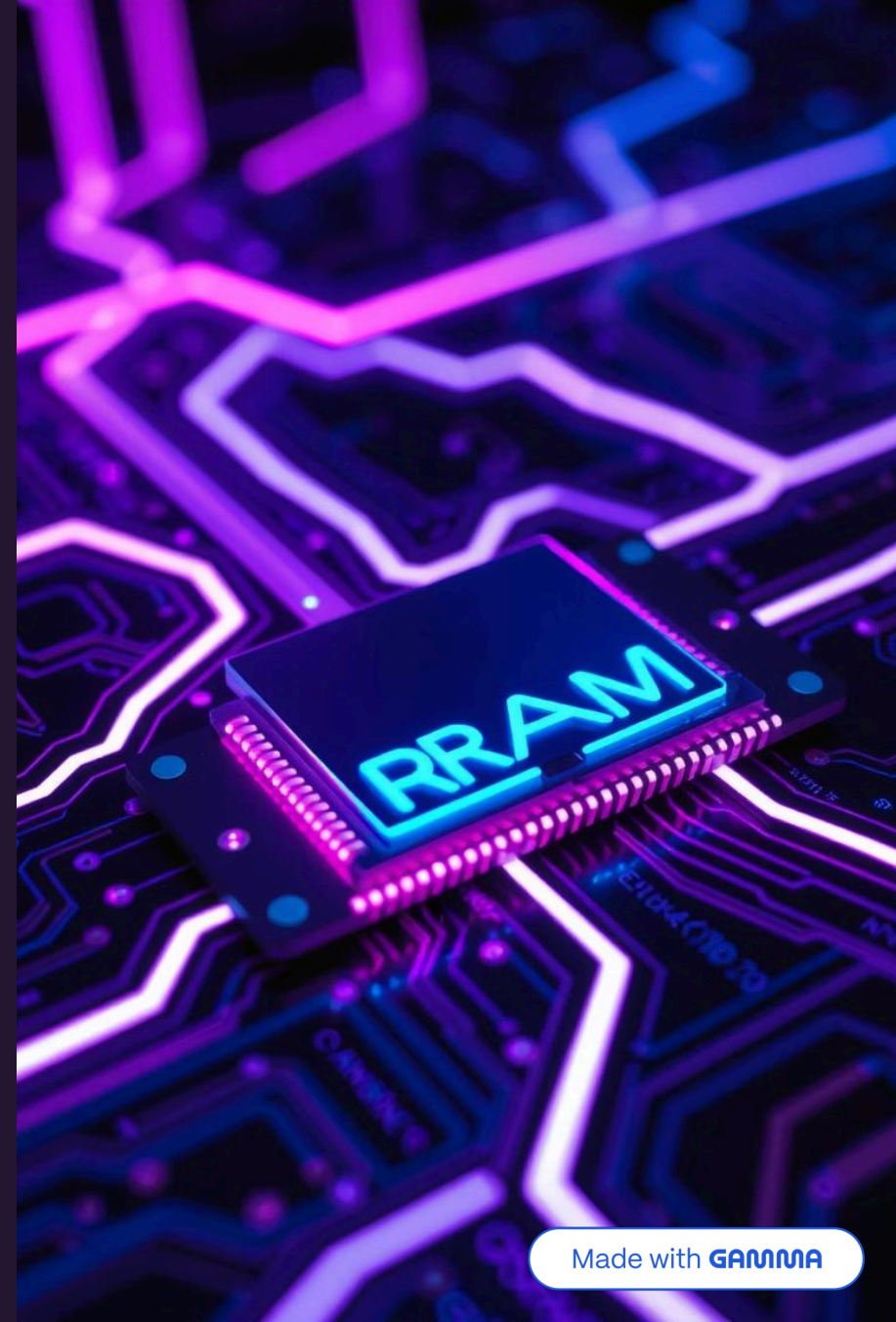


# Gestión de Memoria y Algoritmo LRU

Exploraremos los mecanismos internos de la gestión de memoria en sistemas operativos, centrándonos en el papel crucial de los algoritmos de reemplazo de páginas, especialmente LRU.

**U** por **Ulises Tobares**



# Manejo de Fallos de Página

## Detección y Reacción

Cuando el sistema operativo no encuentra una página en memoria, se produce una interrupción. Esto activa el mecanismo de manejo de fallos.

## Búsqueda en Disco

El sistema busca la página requerida en el espacio de intercambio (swap) del disco. Este proceso es lento pero necesario.

## Carga en Memoria

Una vez encontrada, la página se carga desde el disco a un marco disponible en la memoria RAM.



# Necesidad de Reemplazo de Páginas

## 1 Sin Marcos Libres

Si la memoria RAM está llena y no hay marcos disponibles, no se puede cargar la nueva página.

## 2 Selección Crítica

Es fundamental elegir una página para retirar y liberar espacio para la nueva. Una mala elección degrada el rendimiento.

## 3 Optimización de Rendimiento

El objetivo es minimizar los fallos de página futuros, optimizando el uso de la memoria física.

# FIFO>



## Algoritmo FIFO: El Más Simple

### Principio de Funcionamiento

FIFO (First-In First-Out) selecciona la página que lleva más tiempo en la memoria RAM para ser reemplazada.

### Ventajas

Es el algoritmo más sencillo de implementar. Requiere un seguimiento mínimo de las páginas.

### Desventajas y Anomalías

Sufre de la Anomalía de Belady: más marcos de memoria pueden llevar a más fallos de página, lo que es ineficiente.

# Algoritmo LIFO

## Principio de Funcionamiento

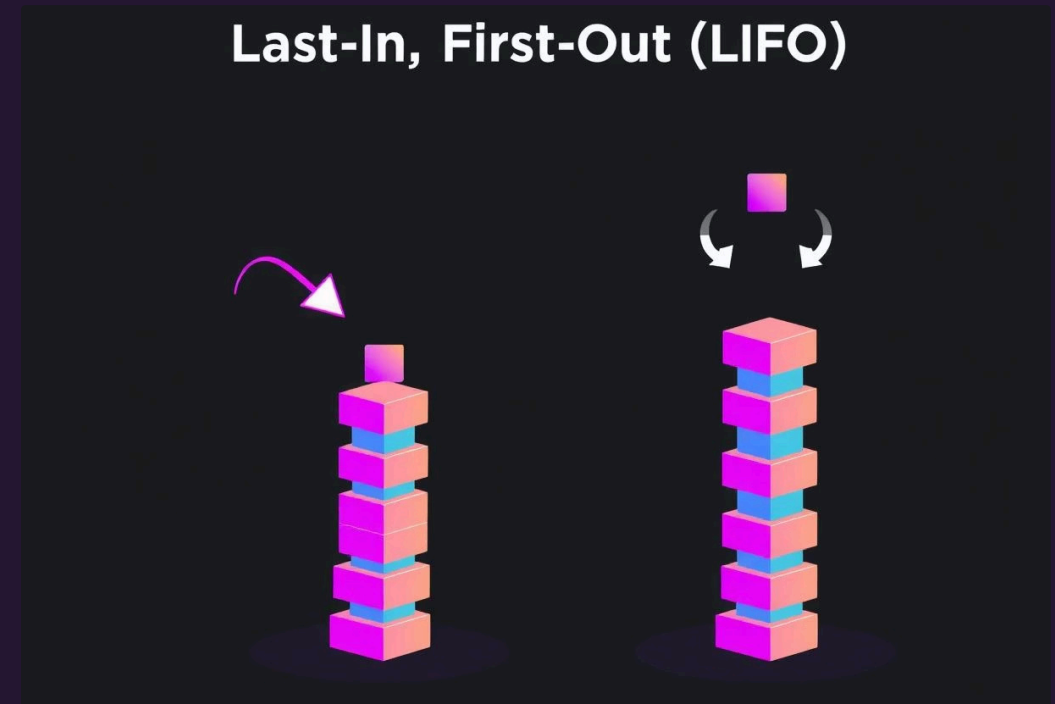
LIFO (Last-In, First-Out) es un algoritmo de reemplazo de páginas que selecciona la página que fue cargada más recientemente en la memoria RAM para ser reemplazada. Se comporta como una pila, donde el último elemento en entrar es el primero en salir.

## Ventajas

Al igual que FIFO, LIFO es un algoritmo conceptualmente simple y fácil de implementar. No requiere un seguimiento complejo del uso de las páginas, solo el orden de entrada.

## Desventajas

La principal desventaja de LIFO es su potencial ineficiencia. Las páginas cargadas recientemente suelen ser las más propensas a ser utilizadas de nuevo en un futuro cercano (principio de localidad temporal), lo que LIFO ignora, pudiendo expulsar páginas útiles prematuramente. Esto puede llevar a un alto número de fallos de página.





# Algoritmo LRU: Eficiencia Mejorada



## Menos Usado Recientemente

LRU (Least Recently Used) reemplaza la página que no ha sido referenciada por el software en el período de tiempo más largo.



## Mejor Rendimiento

En general, LRU supera a FIFO porque aprovecha la localidad temporal de las referencias.



## Mayor Complejidad

Su implementación es más costosa, ya que requiere un seguimiento constante del tiempo de último uso de cada página.



# Implementación de LRU

1

## Uso de Contadores

Cada página tiene un contador que se actualiza con la referencia del reloj lógico. La página con el valor más bajo se reemplaza.

2

## Pilas o Listas

Otra técnica utiliza una pila de páginas. La página referenciada se mueve a la parte superior. La de la parte inferior se reemplaza.

3

## Matrices de Referencia

Se puede usar una matriz de referencia de tamaño  $n \times n$  para  $n$  páginas. Los elementos registran el tiempo de último acceso.



Optimized  
performance

## Ventajas de LRU



### Menos Fallos

Reduce significativamente el número de fallos de página.



### Mejor Utilización de RAM

Mantiene las páginas más relevantes en memoria.



### Rendimiento Estable

Proporciona un rendimiento más predecible y consistente.





# Conclusión y Reflexiones

## Impacto en Sistemas

La elección del algoritmo de reemplazo de páginas afecta directamente la eficiencia del sistema operativo.

## Equilibrio Coste/Beneficio

LRU ofrece un rendimiento superior a expensas de una mayor complejidad de implementación.

## Evolución Continua

Los sistemas modernos utilizan variaciones de LRU para optimizar la gestión de memoria virtual.