

# Manual técnico

2025

Zamora Camacho Adal Mauricio

A2213332218

Mendez Guerrero Pablo Daniel

a2223330178

Ortega Resendiz Luis Fernando

a2183330150

Sanchez Morales Jesus

a2223339020

Vergara Gonzalez Magnus Henrich

a2143222011

---

Sistemas operativos

6 J

28/11/2025

# Manual técnico - Simulador de Memoria Virtual

## Estructura general

- Todo en C++17, puro código de consola - Un solo ejecutable + config.ini - 1200 pasos de simulación fija

```
## Configuración (config.ini)
```

```
ini
```

```
ram_size_kb 2048           ; 2048 KB de RAM → 32 marcos de 64 KB
swap_size_kb 4096          ; 4096 KB de swap → 64 marcos
page_size_kb 64            ; tamaño de página
replacement_algorithm clock ; fifo / lru / clock
enable_tlb true             ; true o false tlb_size 16 ; entradas de TLB
```

## Estructuras de datos principales

- RAM → dos vectores globales:
  - ram\_owner[i] → PID del proceso que ocupa el marco i (-1 = libre)
  - ram\_page\_in\_ram[i] → número de página lógica que está en ese marco
- Swap → un vector swap\_owner[i] (solo dice quién tiene el slot)
- Cada proceso tiene:
  - vector<PageEntry> page\_table → una entrada por página lógica
    - present (bool)
    - frame (marco en RAM o -1)
    - swap\_frame (slot en swap o -1)
- Procesos activos → unordered\_map<int, Process\*> active
- TLB → vector<TLBEntry> tlb con política FIFO simple

## Algoritmos de reemplazo

Todos heredan de ReplacementAlgorithm:

- FIFO.cpp → cola normal, la víctima es la más antigua
- LRU.cpp → lista doblemente enlazada + unordered\_map para acceso O(1)
- Clock.cpp → puntero que da vueltas y bit de referencia

El MemoryManager crea el que diga el config.ini con un make\_unique.

## Flujo de un acceso a memoria

1. Busca en TLB → si está → hit y actualiza algoritmo

2. Si no está en TLB → mira el bit present de la tabla de páginas
3. Si present = false → page fault → llama a algo->selectVictim()
4. Si hay que sacar una página → swap out (si hay espacio) y luego swap in la nueva
5. Actualiza TLB y bit de referencia (si corresponde)

### Salida por consola

- Mapa RAM cada 100 pasos
- Tabla de páginas completa de un proceso aleatorio cada 300 pasos
- Estadísticas finales con % de fallos y TLB hits

Todo impreso con caracteres simples (-, , =) para que no se rompa en CMD viejo.

### Compilación (por si alguien pregunta)

g++ -std=c++17 src/\*.cpp -o simulador.exe

En VS Code solo F5.

Eso es básicamente todo lo que hay que saber del código.