UNIVERSIDAD AUTONOMA DE TAMAULIPAS

Integrantes del Equipo

- Espinosa Vázquez Cristopher

- Polanco Tijerina Jose Luis

- Ponce Medina Jesus

- Barahona Romero Billy Antonio

- Gonzalez Gonzalez Oswaldo

DOCENTE: Muñoz Quintero Dante Adolfo

Manual Tecnico

MATERIA: Sistemas Operativos

VERDAD, BELLEZA, PROBIDAD

Indice:

TÉCNICO DEL SIMULADOR DE GESTIÓN DE MEMORIA BASADO EN PAGINACIÓN Y SWAP

1. Introducción

El presente manual técnico describe la arquitectura, el diseño interno, los componentes lógicos y los algoritmos implementados en el **Simulador de Gestión de Memoria por Paginación** desarrollado como parte del curso de Sistemas Operativos.

El simulador tiene como objetivo modelar el funcionamiento real de la gestión de memoria en un sistema operativo moderno, incluyendo:

- Paginación de memoria
- Asignación de marcos en RAM
- Área de intercambio (Swap)
- Algoritmo de reemplazo de páginas **FIFO**
- Tablas de páginas por proceso
- Visualización del estado del sistema mediante una interfaz gráfica
- Simulación del entorno multiprogramado

Este documento presenta el funcionamiento interno y las decisiones técnicas del proyecto.

2. Objetivo del Sistema

El simulador busca representar de forma visual y precisa el ciclo de administración de memoria, permitiendo al usuario comprender cómo un sistema operativo:

- Traduce direcciones lógicas a físicas.
- Administra marcos disponibles.
- Maneja la ausencia de marcos libres mediante swapping.
- Actualiza tablas de páginas en tiempo real.

El comportamiento del sistema está basado en los principios teóricos vistos en clase.

3. Arquitectura General

El simulador está diseñado bajos los principios de modularidad y separación de responsabilidades. Cada archivo del proyecto implementa un componente funcional independiente, lo cual facilita la lectura, mantenimiento y escalabilidad del sistema.

Estructura del proyecto:

```
/src
   main.py
   administrador_memoria.py
   algoritmo_remplazo.py
   proceso.py
   tabla_paginas.py
   config.py
   config.ini

/docs
/tests
README.md
```

4. Descripción de Módulos

4.1 main.py – Interfaz Gráfica y Controlador Principal

Este módulo actúa como punto de entrada del sistema. Sus funciones principales incluyen:

- Gestión de la interfaz gráfica usando Tkinter.
- Recepción de entrada del usuario para:
  - Crear procesos
  - Liberar procesos
  - Visualizar tablas y memoria
- Envío de peticiones al Administrador de Memoria.
- Actualización visual de RAM, Swap y tablas de páginas.

La GUI representa gráficamente cada marco de RAM y Swap, permitiendo observar cómo el algoritmo maneja asignaciones y reemplazos.

4.2 administrador_memoria.py – Gestor de Memoria

Este módulo implementa toda la lógica interna relacionada con:

- Control de RAM y Swap mediante listas de marcos.
- Asignación secuencial de marcos a páginas de un proceso.
- Detección de falta de marcos libres.
- Invocación del algoritmo de reemplazo.
- Liberación de memoria cuando un proceso finaliza.

Responsabilidades específicas:

- asignar_proceso()

  Calcula número de páginas y asigna marcos.

- liberar_proceso()

  Libera marcos de RAM y Swap.

- realizar_swapping()

  Llama al algoritmo FIFO y mueve páginas víctimas a Swap.

- actualizar_tablas()

  Modifica la tabla de páginas tras cada operación.

La gestión se realiza sobre listas que representan marcos libres y ocupados.

4.3 proceso.py – Modelo de Proceso

Representa un proceso real del sistema operativo.

Atributos:

- pid: Identificador único.
- size_kb: Tamaño total en KB.

- num_paginas: Cantidad total de páginas requeridas.
- tabla_paginas: Estructura que almacena la traducción de páginas → marcos.

Responsabilidad:

Calcular cuántas páginas requiere un proceso y almacenar su tabla de páginas correspondiente.

4.4 tabla_paginas.py – Tabla de Páginas

Este módulo implementa la estructura que permite transformar direcciones lógicas en direcciones físicas.

Cada entrada contiene:

Campo	Descripción
Número de página	Página lógica del proceso
Marco asignado	Marco físico en RAM o Swap
Ubicación	RAM / Swap
Bit de validez	Indica si la página está presente en memoria

La tabla de páginas se actualiza dinámicamente conforme el sistema realiza swapping o libera marcos.

4.5 algoritmo_remplazo.py – Algoritmo FIFO

Implementa el algoritmo de reemplazo **First-In, First-Out**, que selecciona la página que lleva más tiempo en memoria para ser desplazada hacia Swap.

Funcionamiento general:

1. Cuando no hay marcos libres en RAM:
2. Se selecciona la página víctima mediante la cola FIFO.
3. Se mueve la página víctima a Swap.
4. Se actualizan las estructuras del proceso.
5. Se libera el marco físico para asignarlo a una nueva página.

Justificación técnica:

FIFO es simple, eficiente y suficiente para los requerimientos del proyecto.

4.6 config.ini y config.py – Configuración del Sistema

Estos archivos permiten ajustar los parámetros base del simulador:

- Tamaño total de RAM
- Tamaño de Swap
- Tamaño de página/marco

El sistema se inicializa leyendo estos valores, permitiendo condiciones de prueba personalizadas.

5. Flujo Completo del Sistem

5.1 Llegada de un Proceso

1. El usuario ingresa un tamaño en KB.
2. Se calcula cuántas páginas requiere.
3. Se intenta asignar cada página en RAM.
4. Si no hay espacio → se activa reemplazo.

5.2 Asignación de Páginas

El sistema recorre la lista de RAM en busca de marcos libres:

- Si encuentra → asigna el marco.
- Actualiza la tabla del proceso.
- Encola la página en la estructura FIFO.

5.3 Manejo de RAM Llena – Swapping

Cuando no hay marcos disponibles:

1. FIFO elige la **página víctima**.
2. La página víctima se mueve a Swap.
3. Se desasigna el marco.
4. El marco se usa para la nueva página.

5.4 Liberación de Procesos

Al eliminar un proceso:

- Todos los marcos asignados vuelven a estado libre.
- Entradas de Swap se limpian.
- Tabla de páginas queda invalida.
- Se actualiza visualmente el sistema.

6. Estructuras de Datos Clave

Módulo	Estructura	Justificación
RAM	Lista fija	Fácil indexación por marco
Swap	Lista fija	Simula disco virtual
FIFO	Cola de listas	Identifica página víctima
Tabla de páginas	Lista de diccionarios	Representación clara
Procesos	Diccionario	Acceso rápido por PID

7. Métricas del Sistemas:

- Tasa de fallos de página
- Porcentaje de uso de RAM
- Porcentaje de uso de Swap
- Total de reemplazos realizados

8. Limitaciones Actuales

- No incluye una TLB (no requerida).
- Los logs se muestran solo en consola (puedo agregarte la GUI para logs si quieres).
- Métricas aún no implementadas (opcional para versión mejorada).

Imágenes de Apoyo Técnico:

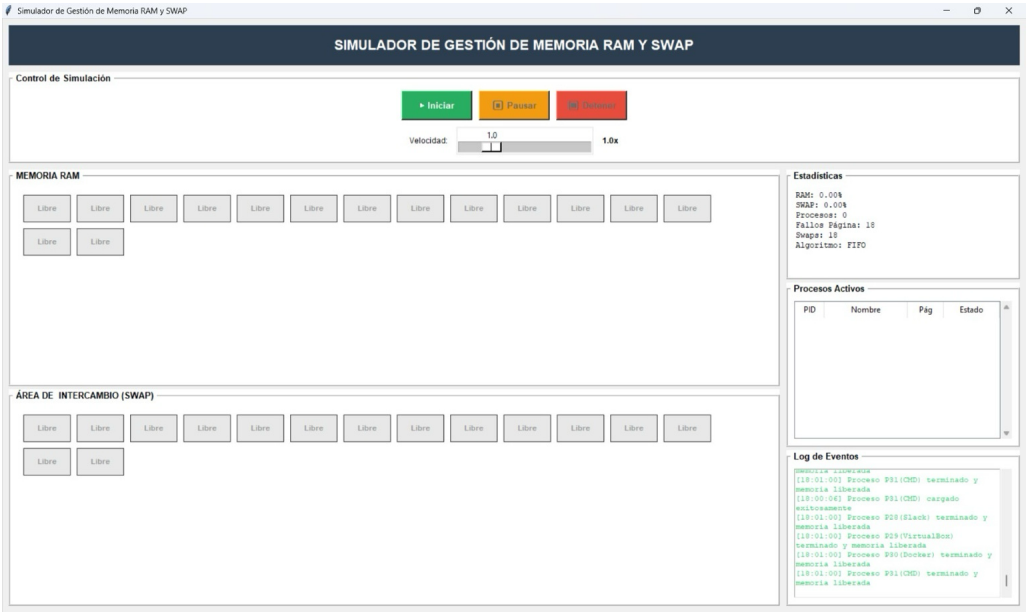

Figura 1. Estado inicial del simulador con memoria RAM y SWAP vacías.

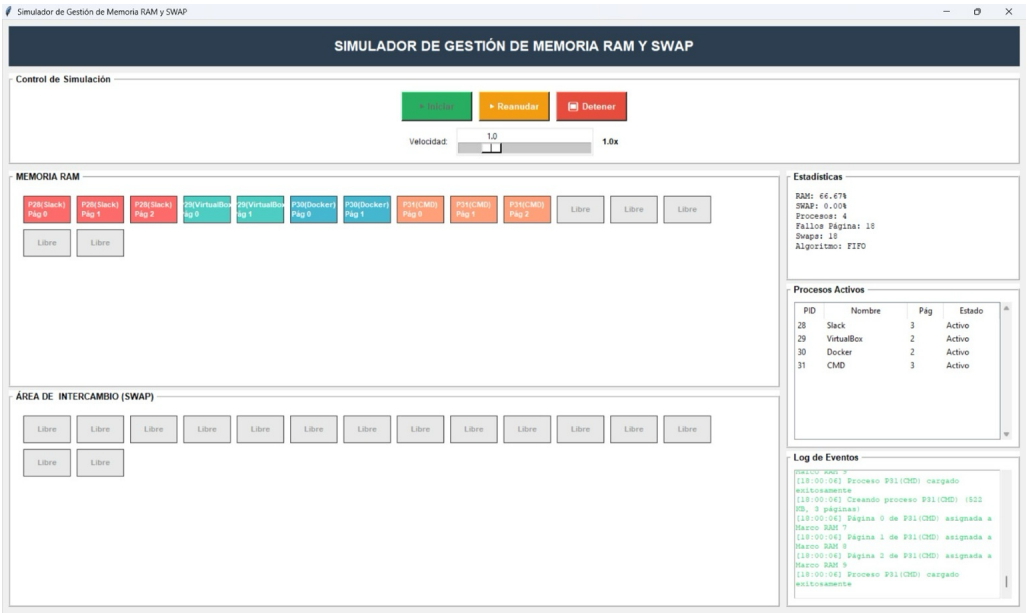

Figura 2. Asignación de páginas al crear múltiples procesos.

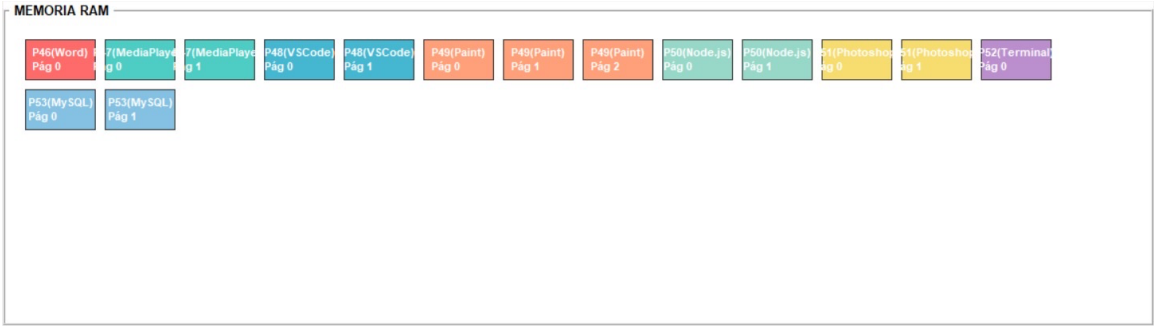

Figura 3. RAM llena mostrando múltiples procesos activos.

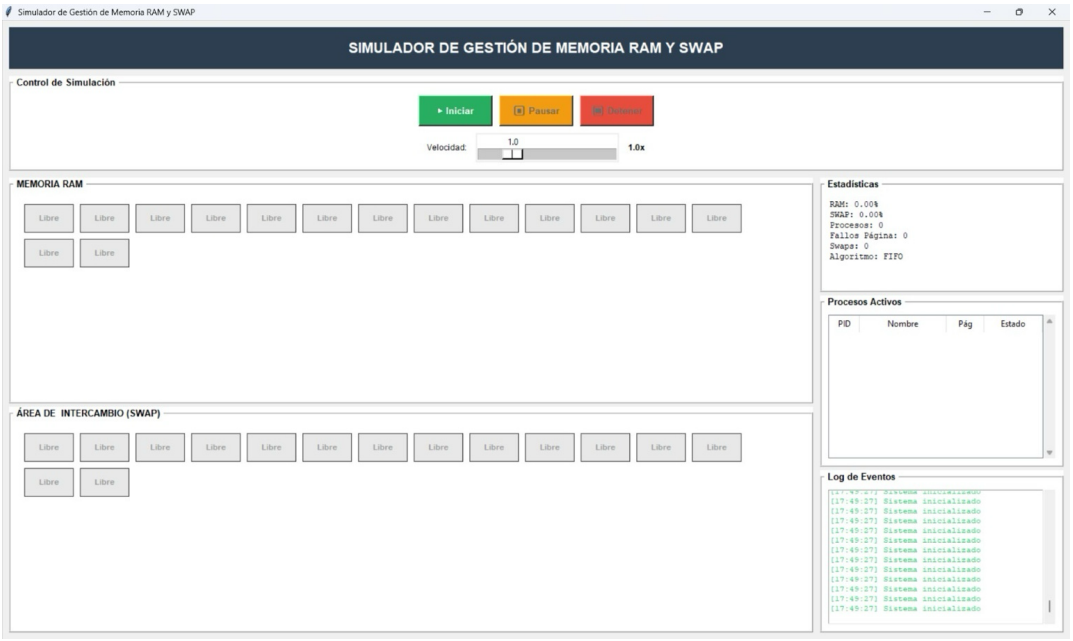

Figura 4. Estado final del sistema con la memoria liberada.

9. Conclusiones Técnicas

El simulador implementa correctamente los elementos fundamentales de la gestión de memoria mediante paginación y swapping:

- Administración de marcos de RAM
- Área de intercambio
- Algoritmo FIFO
- Tablas de páginas por proceso
- Simulación de multiprogramación
- Interfaz gráfica educativa

El diseño modular permite comprender claramente cómo trabaja la memoria virtual en un sistema operativo real.