Proyecto Final

Juan José Álvarez Ocampo

Sara Echeverri Gomez

Nathalia Valentina Cardoza Azuaje

SI2004: Sistemas Operativos

Profesor Diego Ivan Cruz Ordiéres

Medellín, Colombia

Escuela de Ciencas Aplicadas e Ingeniería, Universidad EAFIT

23 de Octubre de 2025

**Objetivo General**

Diseñar un núcleo de sistema operativo simple, que simule los principales procesos de gestión de recursos y del sistema, con el proposito de emplear algoritmos vistos en el curso, utilizando una interfaz de línea de comandos

**Componentes**

Gestión de Memoria: Paginación por demanda. Este algoritmo solamente carga las páginas requeridas en el momento, convirtiéndolo en un algoritmo muy eficiente con los recursos

Planificación de procesos: Round Robin por prioridad; este algoritmo destruye equitativamente los recursos. Envejecimiento de procesos; para los procesos que han esperado mucho tiempo aumenta su prioridad.

Sincronización: Semáforos: este algoritmo controla el acceso a recursos con métodos Walt y Signal. Monitoreo; usan variables de condición para bloquear o despertar procesos.

Comunicación entre procesos: Colas de mensajes; transmiten datos entre procesos. Recepción con timeout; evita bloques o esperas de mensajes indefinidos.

Mecanismos básicos de entrada y salida: Simulación de operaciones con latencia; reproduce los retardos o interrupciones del hardware mediante colas de E/S.

**Stack Principal**

C++20 con CMake y CLI basada en CLI11.

Herramientas y librerías:

• Compilador: Clang ≥ 14 o GCC ≥ 11

• Build: CMake ≥ 3.20

• CLI: [CLI11]

• Pruebas: Catch2 v3

**Alcances**

El proyecto tiene como alcance el diseño e implementación de un núcleo de sistema operativo simplificado, enfocado en la simulación de los principales mecanismos internos de un Sistema Operativo. El sistema permitirá observar el comportamiento dinámico de estos componentes, analizar métricas de rendimiento.

No se busca crear un sistema operativo real ni interactuar con hardware físico, sino modelar su comportamiento de manera controlada mediante el núcleo de un Sistema Operativo.

**Supuestos**

1. La simulación se ejecuta sobre un único procesador lógico.

2. La memoria principal posee un número fijo de marcos; el tamaño de página es constante para todos los procesos.

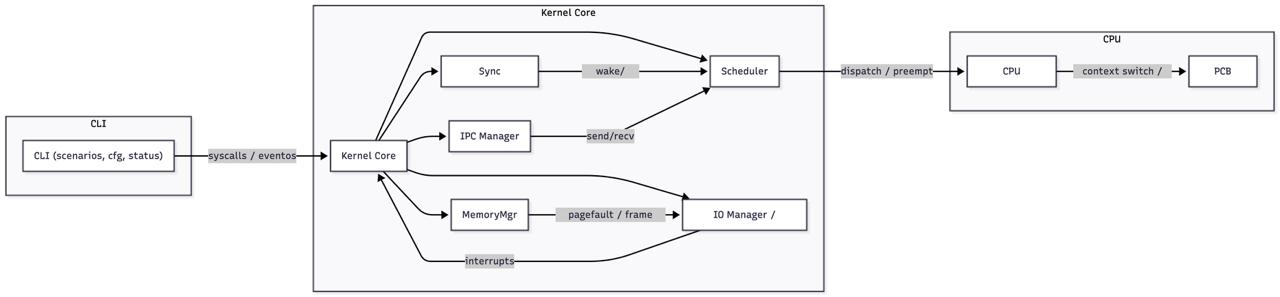
3. Los fallos de página generan una interrupción simulada, provocando la carga de la página requerida desde disco.

4. Las operaciones de entrada/salida son bloqueantes y su tiempo de respuesta está definido.

5. Los mecanismos de sincronización bloquean y despiertan procesos.

6. Las métricas de desempeño se registran en archivos de salida.

**Arquitectura**



**Plan de Prueba**

|  |  |  |
| --- | --- | --- |
| **Módulo** | **Casos de prueba** | **Resultados esperados** |
| CPU | Cambio de contexto y envejecimiento. | Planificación adecuada. |
| Memoria | Carga bajo demanda y fallos de página. | Consistencia. |
| Sincronización | Carga bajo demanda y fallos de página. | Ausencia de deadlocks. |
| IPC | Envío/recepción con y sin timeout. | Comunicación adecuada. |
| E/S | Latencia y interrupciones simuladas | Reporto a READY. |

Integración: En las pruebas de integración esperamos medir el número de fallos de páginas, evaluar los tiempos promedios de espera y, en general, observar el comportamiento del sistema.

Rendimiento: Mediremos el consumo de recursos como CPU o memoria y la latencia generada.