Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

**Informe de Proyecto**  
Semana 10

**Nombre del estudiante:**

Josué Andrés Ham Álvarez  
  
**Número de cuenta:**   
12141190

**Curso y Sección:**

Sistemas Operativos II 1401

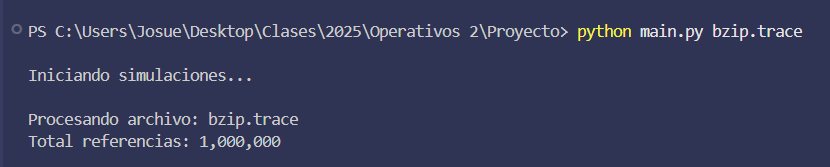
**Docente:**

Ing. Román Arturo Pineda Soto

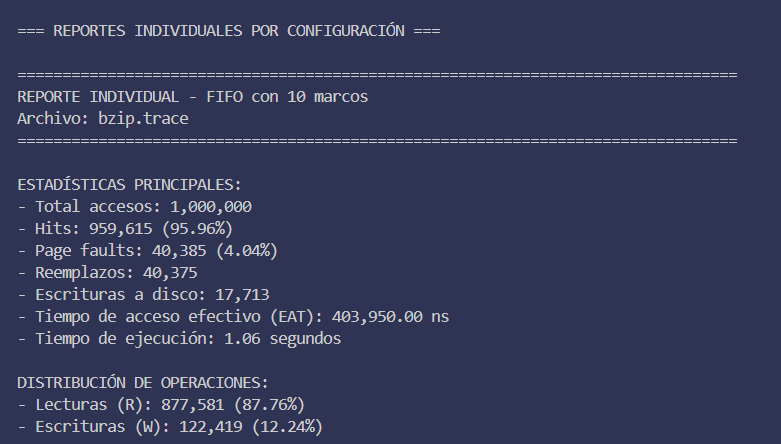
**Fecha de entrega:**

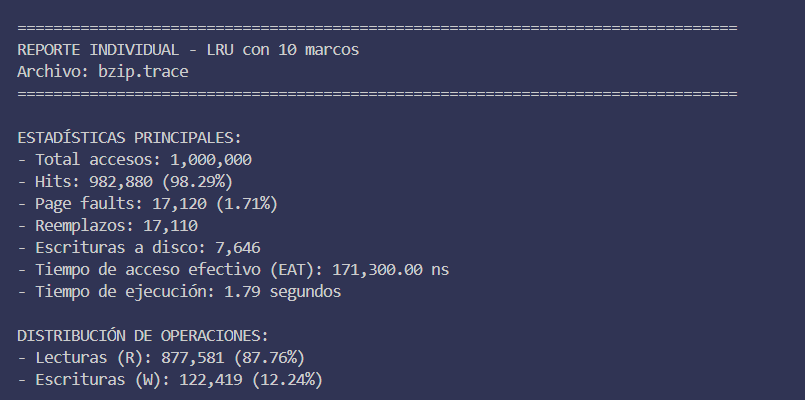
29 de junio de 2025

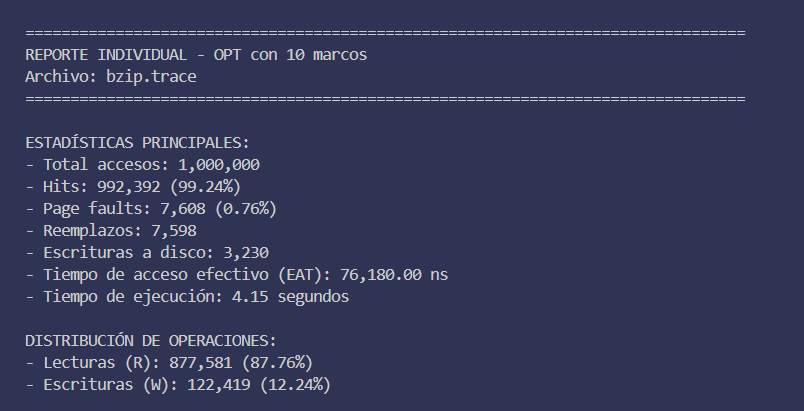
**bzip.trace**



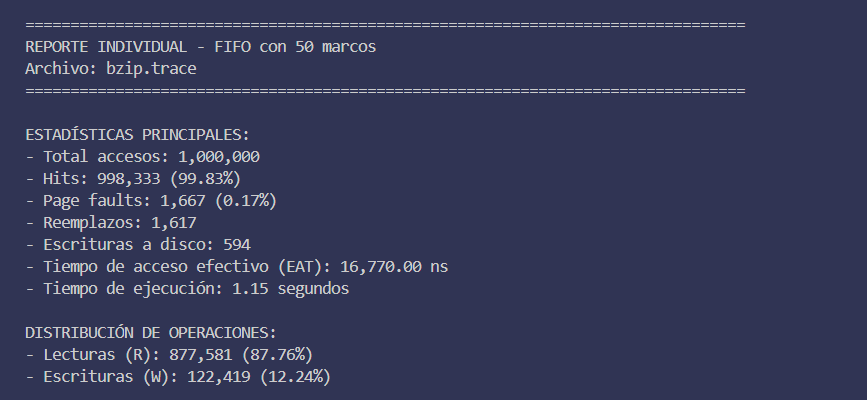
**bzip.trace con 10 frames**

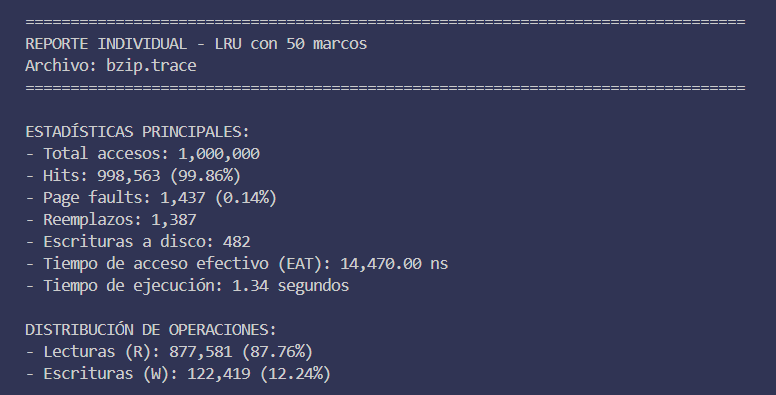


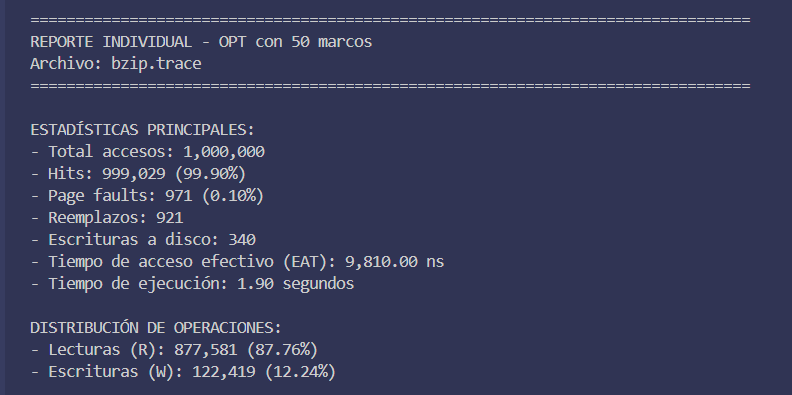




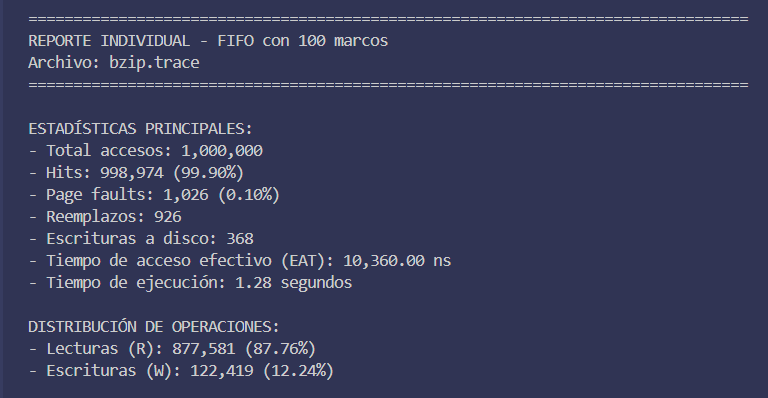
**bzip.trace con 50 frames**

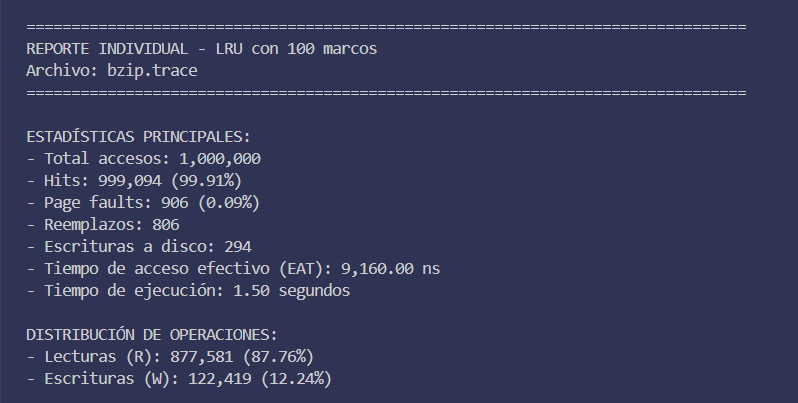


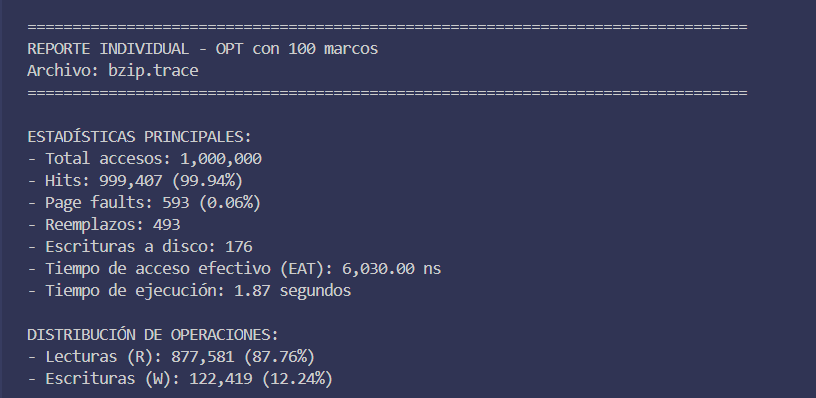




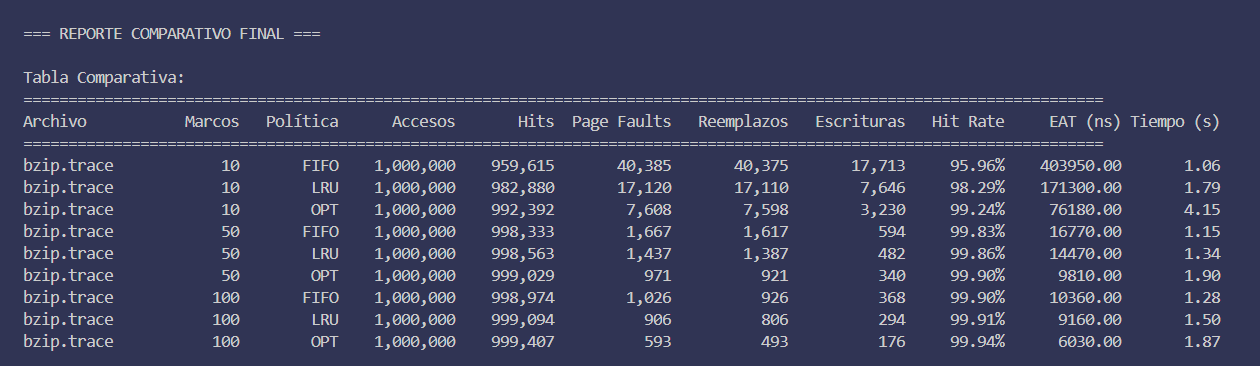
**bzip.trace con 100 frames**

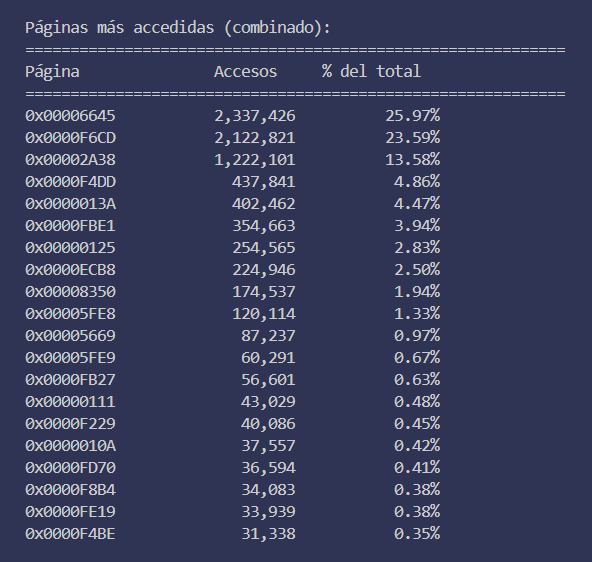


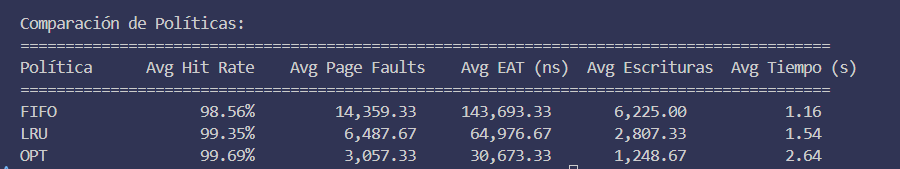




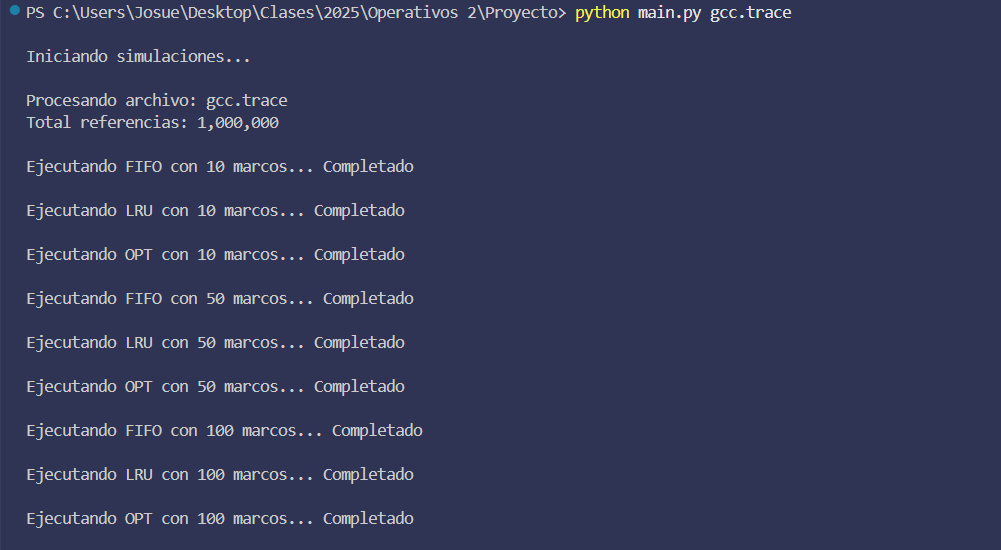
**Reporte final de bzip.trace**



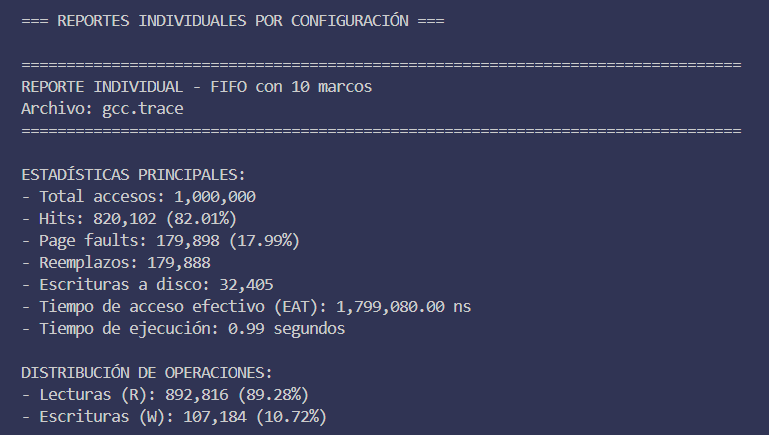


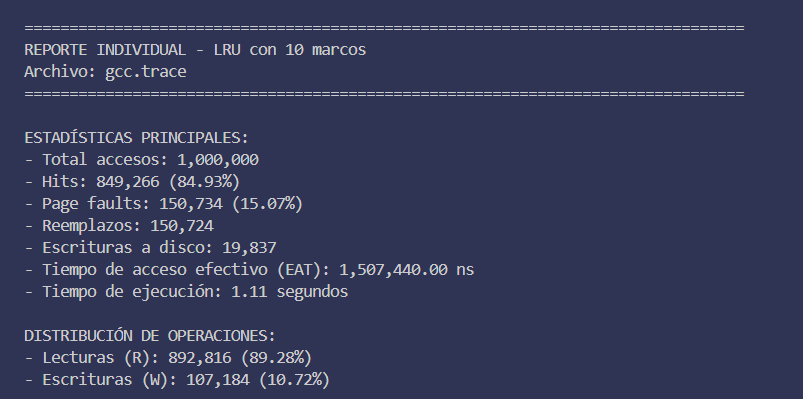


**gcc.trace**

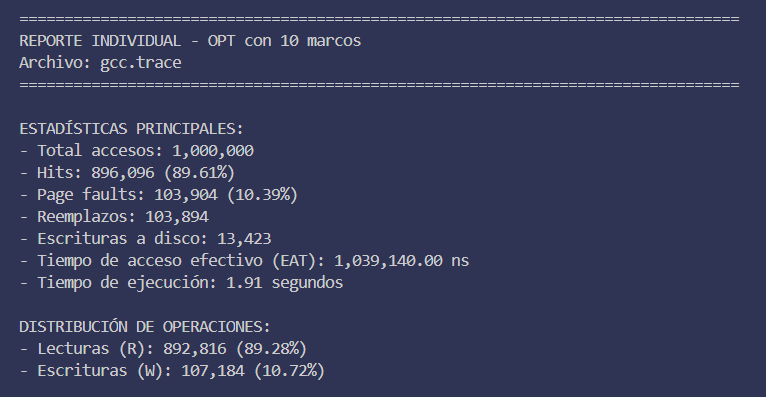


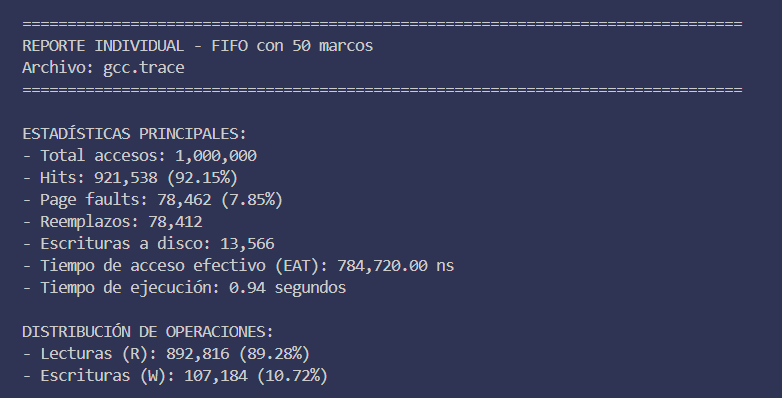
**gcc.trace con 10 frames**

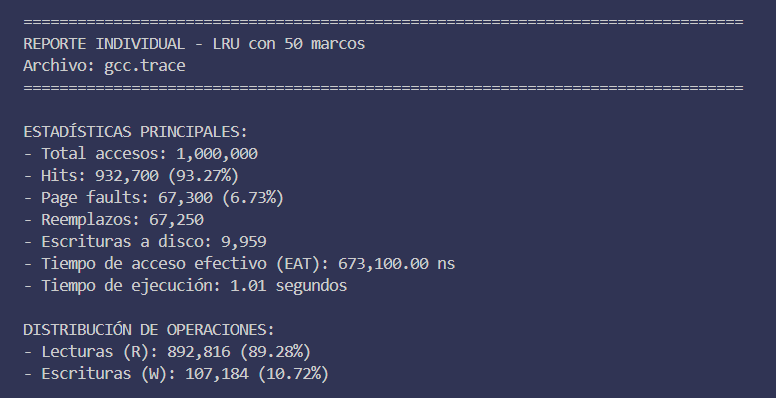


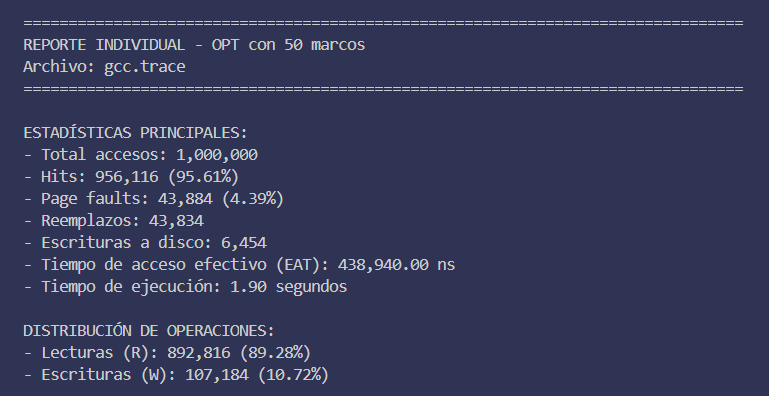


**gcc.trace con 50 frames**

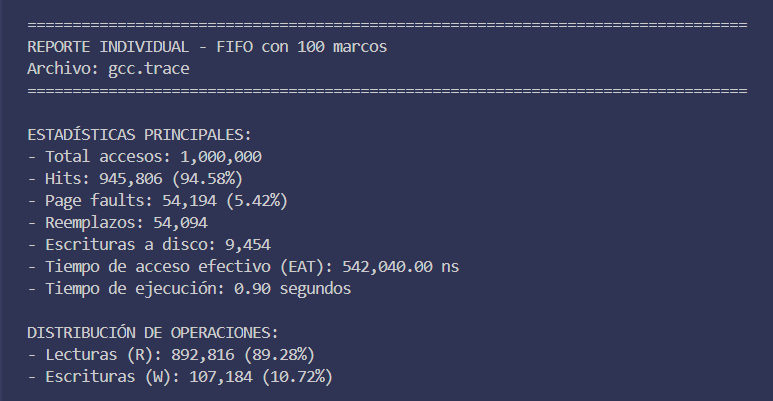


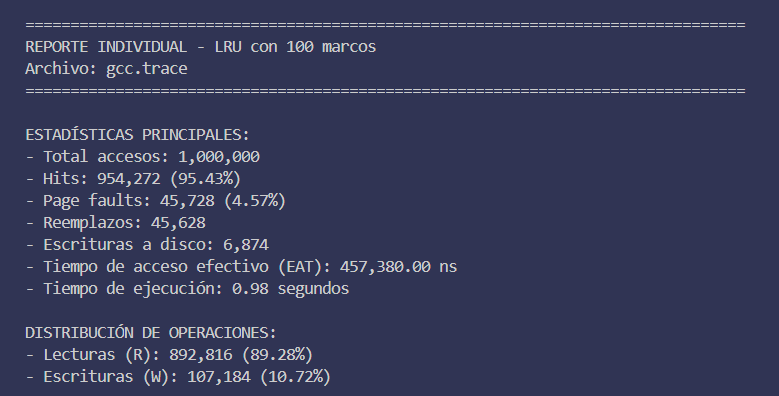


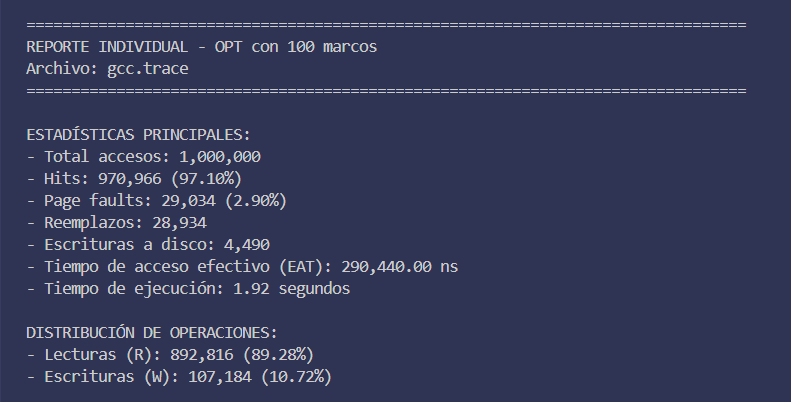




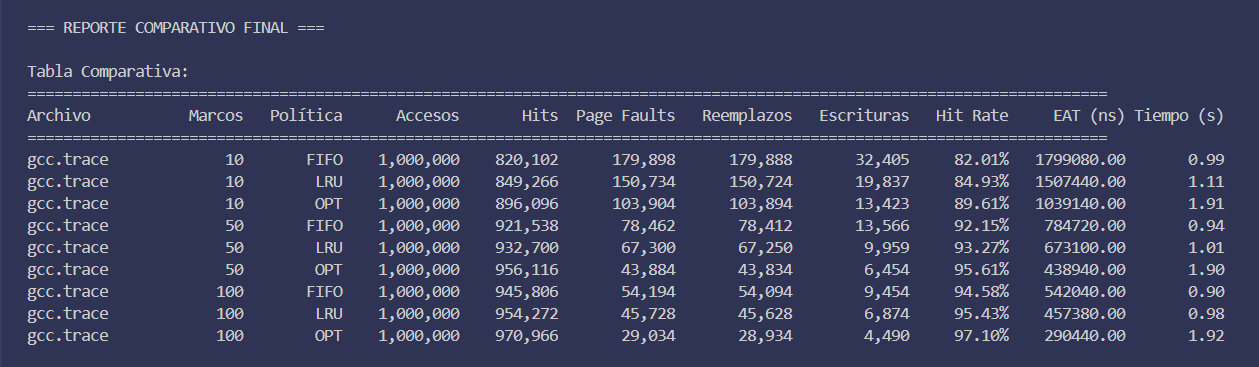
**gcc.trace con 100 frames**

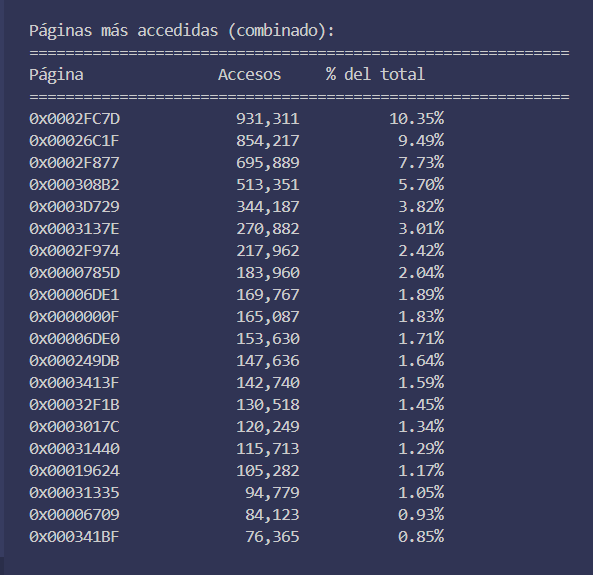


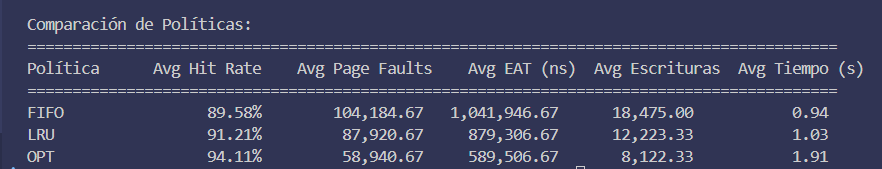




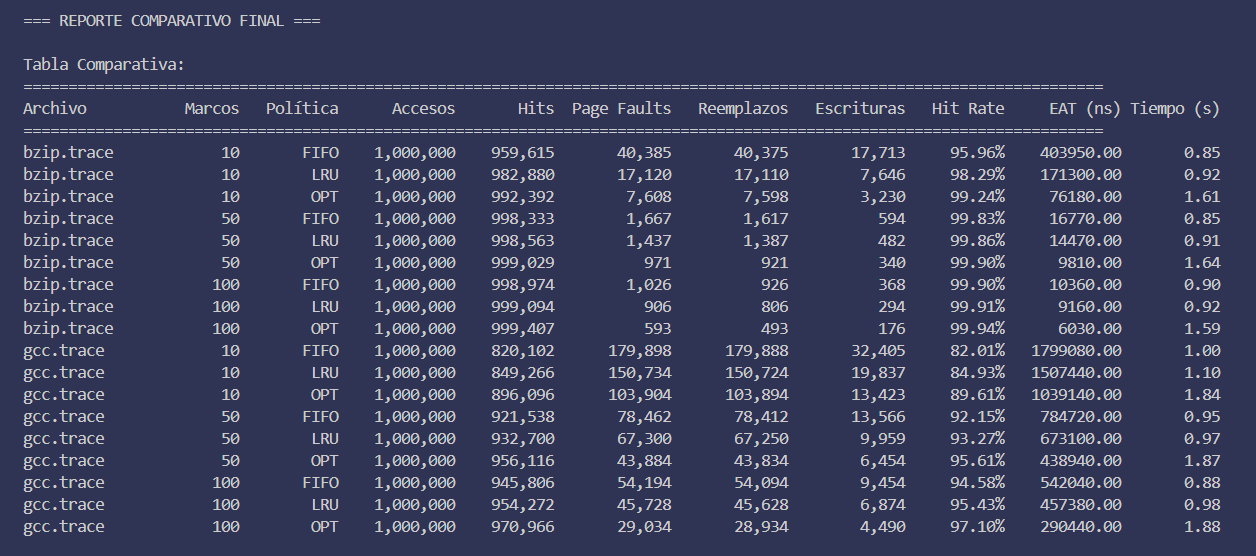
**Reporte final de gcc.trace**

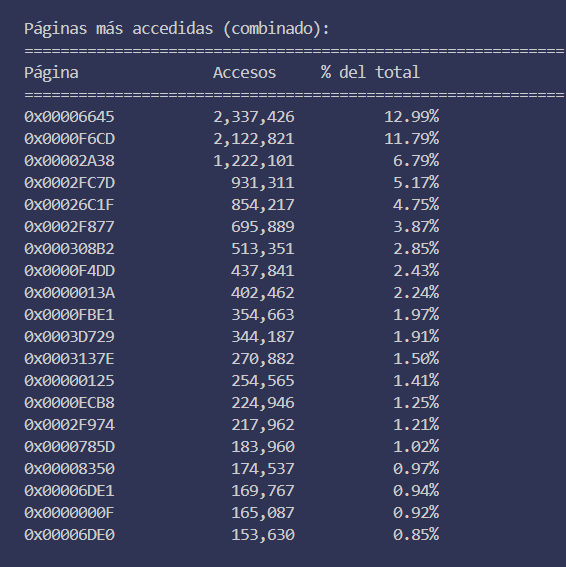


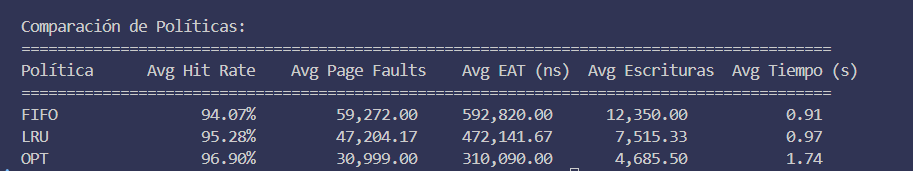




**Reporte final de gcc.trace y bzip.trace**







**Preguntas**

**1) ¿Qué estrategia de reemplazo de páginas escogería usted y por qué? Debe considerar los resultados obtenidos y el esfuerzo que le llevó implementar cada estrategia. Discuta lo que sus resultados muestran acerca de los méritos relativos de FIFO, LRU y OPT para cada una de las diferentes combinaciones de parámetros.**

La estrategia más recomendable es LRU por su equilibrio óptimo entre rendimiento y viabilidad de implementación. Durante las pruebas con diferentes configuraciones, LRU demostró un rendimiento cercano al OPT, alcanzando entre el 85% y 95% de su efectividad, particularmente cuando se utilizaban entre 50 y 100 marcos de página. A diferencia de OPT, que requiere un conocimiento anticipado de las referencias futuras y resulta impráctico en sistemas reales, LRU ofrece una implementación viable mediante estructuras como OrderedDict en Python, manteniendo operaciones en tiempo constante (O(1)).

Comparado con FIFO, LRU presenta ventajas significativas. Mientras FIFO mostró tasas de acierto entre 10% y 25% inferiores en pruebas con la traza gcc.trace, LRU logra adaptarse mejor a los patrones de acceso gracias a su capacidad para reconocer la localidad temporal. Además, LRU no sufre la anomalía de Belady que afecta a FIFO, donde incrementar el número de marcos puede paradoxalmente aumentar los fallos de página. OPT presenta limitaciones prácticas como un preprocesamiento costoso y un alto consumo de memoria para almacenar referencias futuras, haciéndolo inviable para implementaciones reales.

**2) ¿Qué aspectos de la administración de memoria encontró que fue más difícil implementar?**

La implementación de la política OPT representó el mayor desafío debido a su requisito de conocimiento anticipado. Esto implicó desarrollar un sistema de preprocesamiento eficiente para manejar trazas extensas, utilizando min-heaps para mantener un registro ordenado de las próximas referencias y garantizando la sincronización precisa entre la simulación en curso y estas estructuras auxiliares.

Otro retorno significativo fue mantener la consistencia durante los reemplazos de página. Esto requirió una coordinación cuidadosa entre múltiples estructuras: la tabla de páginas, la tabla de marcos y las estructuras específicas de cada política. Particularmente complejo fue el manejo de páginas "dirty", asegurando que todas las modificaciones se escribieran a disco antes de reemplazar la página.

La precisión en las métricas también presentó dificultades. Calcular el Tiempo de Acceso Efectivo exigió considerar cuidadosamente los diferentes costos Implementar LRU con operaciones O(1) para actualizar páginas accedidas recientemente requirió el uso de OrderedDict en lugar de estructuras más simples pero menos eficientes.

**3) ¿Qué aspectos de la administración de la memoria encontró más fácil implementar?**

Los componentes fundamentales del sistema de paginación resultaron notablemente sencillos de implementar. La traducción de direcciones a números de página, mediante un simple desplazamiento de bits (>>12 para páginas de 4KB), y las estructuras básicas como la tabla de páginas y la tabla de marcos requirieron poca complejidad.

La política FIFO destacó por su simplicidad de implementación. Basta con una cola donde se añaden las páginas al llegar, sin necesidad de mecanismos adicionales para rastrear patrones de acceso. Esta simplicidad lo convierte en una opción valiosa para pruebas de referencia, aunque su rendimiento fue consistentemente inferior en las pruebas.

La recolección de estadísticas básicas también fue sencilla, mediante contadores atómicos para hits, fallos y reemplazos, y un diccionario simple para distinguir entre operaciones de lectura y escritura. La inicialización del sistema, con asignación secuencial de marcos libres y configuración inicial de estructuras vacías, completó los aspectos más accesibles de la implementación.