

ST0257 – Sistemas operativos

Proyecto 1: Key-value Store

MBA, I.S. José Luis Montoya Pareja
Departamento de Informática y Sistemas
Universidad EAFIT
Medellín, Colombia, Suramérica

RESUMEN

Este proyecto implica el diseño e implementación de un componente fundamental en numerosos sistemas distribuidos: un almacén simple Key-Value. Los estudiantes profundizarán en la serialización de datos, explorarán diversos mecanismos de almacenamiento (por ejemplo, archivos, bases de datos) y adquirirán una comprensión integral de la consistencia de datos. Para mejorar la aplicación práctica, los estudiantes utilizarán su almacén Key-Value creado para almacenar en caché datos a los que se accede con frecuencia dentro de una aplicación. Este caso de uso del mundo real, como almacenar en caché perfiles de usuario o información de productos, mejorará significativamente el rendimiento de la aplicación al reducir la necesidad de recuperar datos repetidamente de fuentes más lentas como bases de datos o API externas. Además, los estudiantes evaluarán el rendimiento de su almacén clave-valor midiendo el tiempo empleado en las operaciones clave-valor (insertar, obtener, eliminar) bajo diferentes cargas de trabajo y tamaños de datos. Esta experiencia práctica proporcionará valiosas perspectivas sobre las implicaciones del mundo real de su diseño e implementación, incluido el análisis de rendimiento y la optimización.

PALABRAS CLAVE

Almacenamiento Key-Value, Caché, Serialización, Deserialización, Persistencia, Consistencia de datos, Estructuras de datos, Algoritmos de búsqueda, Rendimiento, Optimización, Sistemas distribuidos.

CONTEXTO

En el mundo actual, los sistemas informáticos se enfrentan a la necesidad de procesar grandes volúmenes de datos de manera rápida y eficiente. Una estrategia clave para mejorar el rendimiento de las aplicaciones es utilizar un almacenamiento en caché para almacenar temporalmente datos frecuentemente accedidos.

Un almacén clave-valor es una estructura de datos fundamental que permite almacenar y recuperar información de manera eficiente utilizando pares clave-valor. Esta estructura es ampliamente utilizada en diversas aplicaciones, como servidores web, bases de datos, sistemas de recomendación, y plataformas de

comercio electrónico, para mejorar la velocidad de respuesta y reducir la carga en sistemas back-end.

En este caso, usaremos la información para mejorar la velocidad de respuesta a posibles preguntas que se puedan generar en un dataset de datos, similar a como funciona un motor de base de datos o plataformas de recomendación. La información del dataset y los archivos los pueden encontrar en el sitio <https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam>

OBJETIVOS

- ☐ **Comprender:** Conceptos de **Key-value store** y sus aplicaciones en sistemas distribuidos.
- ☐ **Aprender:** Técnicas de serialización y deserialización de datos.
- ☐ **Explorar:** Diferentes mecanismos de almacenamiento y sus ventajas e inconvenientes.
- ☐ **Adquirir Experiencia:** Manejar la consistencia y durabilidad de los datos.
- ☐ **Implementar:** Operaciones centrales del **Key-value store** (insertar, obtener, eliminar).
- ☐ **Evaluar:** Rendimiento del **Key-value store** a través de casos de uso prácticos y pruebas de rendimiento.
- ☐ **Analizar:** Factores que influyen en el rendimiento e identificar posibles optimizaciones.

ACTIVIDADES

La práctica consiste en realicen las siguientes tareas:

1. Descargar los archivos del dataset
2. Crear un programa que los lea a memoria
3. Presente un menú de opciones para que realicen las siguientes acciones:
 - a. Indicar cuales son los 10 juegos más recomendados
 - b. Indicar cuales son los 10 juegos menos recomendados
 - c. Indicar cuales son los 10 usuarios con más recomendaciones
 - d. Indicar cuales son los juegos que más recomiendan los 10 usuarios

NOMBRE

dataload – Lector de datos.

SINOPSIS

dataload -f FOLDER

DESCRIPCION

Se verifica si FOLDER es una carpeta que está en la misma ruta donde está el ejecutable. Por cada archivo con extensión .csv que encuentre en la carpeta, debe cargarlo en memoria en un ArrayList o similar.

OPCIONES

Estado de salida del proceso:

- | | |
|---|-----------------------------------|
| 0 | Si el proceso termina OK |
| 1 | Si el proceso termina con errores |

CONSIDERACIONES GENERALES

1. El desarrollo de la práctica puede ser individual o en equipos de máximo tres personas.
2. Deben proponer en qué contexto usarán el concepto de Key-value para la solución de este proyecto.
3. La entrega de la práctica se realizará entregando los fuentes en un archivo y el informe por el buzón recepción de trabajos de Eafit Interactiva (cualquier otro medio no será admitido).
4. Se debe informar al profesor a más tardar el 31 de enero a las 9:00 p.m. los integrantes del equipo vía Teams.
5. El informe final es una presentación que deberá contener una breve descripción de cómo funciona el programa, tablas o gráficos donde se muestre la ejecución de su programa en diferentes máquinas describiendo de cada una el tipo de procesador, cantidad de memoria RAM y sistema operativo que tienen instalado (puede ser en varios) y unas conclusiones que ustedes hagan sobre los datos obtenidos.
6. La práctica se puede realizar en cualquier lenguaje de programación. En el informe deben informar cual es la versión del compilador o del runtime que están usando.
7. Cada semana se sacará un espacio de 10 a 15 minutos al inicio de la clase para hablar de la práctica y resolver dudas.

8. Criterios de evaluación (ver Anexo 1)

FECHA DE ENTREGA

Viernes 21 de febrero en clase a través de Eafit Interactiva.

SUSTENTACIÓN

Viernes 21 de febrero en clase. El mecanismo de sustentación es el siguiente:

1. Al inicio de la clase, se realiza la evaluación de 15 minutos.
2. Cada equipo muestra su desarrollo ejecutando en vivo, dejando ver que el programa se ejecuta con los tres modos solicitados.
3. Debe mostrar cómo solucionó cada uno de los retos y cómo lo relacionaron con conceptos vistos en clase.
4. Mostrar los resultados y explicar las conclusiones.
5. Se realizarán preguntas por parte del docente para validar el entendimiento individual de los conceptos aplicados. Esto significa que, aunque el trabajo es en equipo, la nota del trabajo puede ser diferente para cada uno de acuerdo con la calidad de las respuestas.

Nombre de la asignatura: Sistemas Operativos

Competencia a la que aporta la asignatura: Conocer el sistema operativo del computador para un mejor desarrollo, diseño y ejecución de las aplicaciones y aplicar nuevas soluciones.

Resultado de asignatura evaluado: Lectura de archivos secuencial y en paralelo.

Evento evaluativo: Proyecto 3

% del evento evaluativo: 25%

Criterios (que tributen al RA de asignatura)	Cumple con altos estándares (4.5 -5)	Cumple a satisfacción (4 -4.4)	Cumple parcialmente (3.5-3.9)	Incumple parcialmente (2.5-3.4)	Incumple totalmente (0 -2.4)	Peso asignado al criterio sobre la calificación.
Análisis de fundamentación para la gestión de la lectura del archivo y la generación del cache. Claridad en el concepto	Entiende completamente la necesidad y plantea varias opciones de solución. Utiliza conceptos adecuadamente para la solución.	Entiende completamente la necesidad y plantea una opción de solución.	Omitió un elemento clave para el entendimiento de la necesidad	Omitió varios elementos para el entendimiento de la necesidad.	Demuestra poco o nulo entendimiento del problema.	40%
Diseño de la solución Solución óptima	El diseño tiene en cuenta los conceptos vistos en clase y argumenta la elección de su solución. La solución elegida es la óptima para el problema.	El diseño tiene en cuenta los conceptos vistos en clase y argumenta la elección de su solución.	Aunque se tuvieron en cuenta conceptos vistos en clase, no hubo argumentación correcta en la elección de la solución.	No se tuvieron en cuenta los conceptos vistos en clase.	Demuestra poco o nulo entendimiento de patrones de paralelismo al momento de explicar la solución.	40%

<p>Funcionalidad</p> <p>Calidad de la solución frente a necesidad planteada</p>	<p>El programa funciona correctamente y se entrega de la forma descrita en el documento.</p>	<p>La solución elegida no es la óptima pero el programa funciona correctamente y se entrega de la forma descrita en el documento.</p>	<p>El programa con los conceptos vistos en clase no funciona correctamente o no se entrega de la forma descrita en el documento.</p>	<p>El programa no tiene en cuenta ningún concepto visto en clase y la solución funciona correcta o parcialmente o no se entrega correctamente.</p>	<p>Se entrega la solución parcialmente o no se entrega ninguna solución o no se entrega correctamente.</p>	<p>20%</p>
---	--	---	--	--	--	------------