**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

Estudiante 3 Cod XXXX

|  |  |  |  |
| --- | --- | --- | --- |
|  | Máquina 1 | Máquina 2 | Máquina 3 |
| Procesadores |  |  |  |
| Memoria RAM (GB) |  |  |  |
| Sistema Operativo |  |  |  |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @LP [ms]** |
| 0.1 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 1.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @SC [ms]** |
| 2.00 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 1.**

* Comparación de memoria y tiempo de ejecución para PROBING y CHAINING

# **Maquina 2**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @LP [ms]** |
| 0.1 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

Tabla 4. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 2.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @SC [ms]** |
| 2.00 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 2.**

* Comparación de memoria y tiempo de ejecución para PROBING y CHAINING

# **Maquina 3**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @LP [ms]** |
| 0.1 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

Tabla 6. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 3.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @SC [ms]** |
| 2.00 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

Tabla 7. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 3.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 3.**

* Comparación de memoria y tiempo de ejecución para PROBING y CHAINING

# **Preguntas de análisis**

1. ¿Por qué en la función **getTime()** se utiliza **time.perf\_counter()** en vez de otras funciones como **time.process\_time()**?

En la función getTime() se utiliza time.perf\_counter() en vez de otras funciones como time.process\_time(), debido a que esta no solo devuelve el valor flotante del tiempo en segundos sino que tiene en cuenta el tiempo durante el sueño y abarca todo el sistema. En otras palabras, tiene en cuenta el tiempo que hay en el momento en que un hilo se desocupe y entre el tiempo que se le asigna otro trabajo. Además, tiene en cuenta el tiempo de espera de un hilo, es decir, el tiempo cuando un hilo se detiene y tiene que esperar a que otro hilo cumpla con una labor para que este pueda seguir con su trabajo.

1. ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?

Las funciones start() y stop() de la librería tracemalloc son importantes debido a que permiten iniciar y finalizar el proceso para medir memoria. Si estos no se utilizaran no se podría establecer cuando comenzar o cuando se deberñia detener esta medición, por lo que seguiría por un tiempo indefinido.

1. ¿Por qué no se puede medir paralelamente el **uso de memoria** y el **tiempo de ejecución** de las operaciones?

No se puede medir paralelamente el uso de memoria y el tiempo de ejecución de las operaciones debido a que se afectan mutuamente, es decir, la medición de la memoria afecta el tiempo de ejecución de las operaciones y viceversa. Lo que causa conflictos en que los resultados sean óptimos.

1. Teniendo en cuenta cada uno de los requerimientos del reto ¿Cuántos índices implementaría en el Reto? y ¿Por qué?
2. Según los índices propuestos ¿en qué caso usaría **Linear Probing** o **Separate Chaining** en estos índices? y ¿Por qué?
3. Dado el número de elementos de los archivos del reto (large), ¿Cuál sería el factor de carga para estos índices según su mecanismo de colisión?
4. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de contenido FIFA?
5. ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar el catálogo de contenido FIFA?
6. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
7. ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
8. ¿Qué configuración de ideal ADT Map escogería para el **índice jugadores FIFA (“scorer”)**?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.