

Tiempo de ejecución

Integrantes	Rol	Rut	Paralelo
Elias Valle	202173537-2	20.920.259-K	200
Javiera Gutiérrez	202173626-3	21.080.000-K	200

PREGUNTAS Y ANÁLISIS

1. Cree una table con los tiempos de ejecución de cada palabra y su respectiva orientación.

Respecto al programa 'main.c':

Palabra	Orientación	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
carro	vertical	0.000090	0.000091	0.000091	0.000091
gato	vertical	0.000025	0.000038	0.000024	0.000029
carne	vertical	0.000487	0.000346	0.000345	0.000392
gamer	vertical	0.000092	0.000111	0.000092	0.000098
cobre	horizontal	0.000216	0.000228	0.000214	0.000219
viktor	horizontal	0.000120	0.000068	0.000055	0.000081
perro	horizontal	0.000012	0.000023	0.000012	0.000016
casa	horizontal	0.000003	0.000003	0.000002	0.000003
hola	vertical	0.000029	0.000040	0.000030	0.000033
banco	horizontal	0.000261	0.000270	0.000259	0.000263
jamón	vertical	0.000088	0.000102	0.000089	0.000093
tapia	horizontal	0.000009	0.000022	0.000010	0.000014

2. ¿Qué palabra tuvo un mayor tiempo de ejecución? Justifique.

La palabra 'carne' tuvo mayor tiempo de ejecución debido a su orientación vertical en la sopa de letras. Dado que el archivo.txt tiene esa orientación, el código convierte las columnas a filas al ingresar a la matriz, lo que ralentiza el ingreso de valores. Además, en el archivo 'carne.txt', la palabra está en la columna 99 de una sopa de letras 200x200, lo que retrasa aún más el proceso de búsqueda.

3. ¿Qué orientación tuvo un menor tiempo de ejecución? ¿A qué se debe esto?

La orientación con menor tiempo de ejecución fue horizontal. Esto se debe a que el código ahora no introduce valores en una matriz, sino que en un arreglo unidimensional. Esta estructura evita cálculos adicionales al acceder a los valores, como sucede en una matriz bidimensional. El código se adapta a esta mejora al usar funciones específicas para cada orientación.

4. ¿Cómo podría optimizar su código de forma que pueda minimizar sus tiempos de ejecución? Realice el código.

El código optimizado se llama 'new.c', las diferencias que tiene con 'main.c' son las siguientes:

- **horizontal()**: Se realiza la eliminación de espacios en blanco y la conversión en minúsculas mientras se lee cada línea, en lugar de hacerlo llamando otras funciones como es en el caso de 'main.c'.
- **vertical()**: Se obtiene el tamaño de la matriz (filas y columnas) en la primera iteración, lo que reduce la necesidad de una iteración adicional para determinar el tamaño.

5. Cree una tabla con los nuevos tiempos de ejecución de cada palabra y su respectiva orientación utilizando el código realizado en la pregunta anterior.

Respecto al programa 'new.c':

Palabra	Orientación	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
carro	vertical	0.000030	0.000038	0.000030	0.000033
gato	vertical	0.000006	0.000008	0.000007	0.000007
carne	vertical	0.000097	0.000077	0.000107	0.000094
gamer	vertical	0.000053	0.000038	0.000039	0.000043
cobre	horizontal	0.000193	0.000188	0.000180	0.000187
viktor	horizontal	0.000059	0.000047	0.000047	0.000051
perro	horizontal	0.000021	0.000009	0.000010	0.000013
casa	horizontal	0.000003	0.000002	0.000002	0.000002
hola	vertical	0.000036	0.000020	0.000020	0.000025
banco	horizontal	0.000237	0.000268	0.000223	0.000242
jamon	vertical	0.000041	0.000027	0.000027	0.000031
tapia	horizontal	0.000019	0.000008	0.000008	0.000012

6. ¿Qué materia de curso crees que podría ayudar a solucionar este problema? Justifique.

La materia de curso que podría ayudar a solucionar el problema es 'rendimiento'. Entender el como funciona la estructura interna de los computadores, en como el hardware interactúa con el software y como los componentes del sistema influyen en el rendimiento de los programas. Esto permitiría identificar estrategias para mejorar la eficiencia y reducir los tiempos de ejecución.

Por ejemplo, entender el CPI (Ciclos por Instrucción), este es un indicador clave que refleja como las instrucciones se ejecutan en un procesador y como afectan al tiempo de ejecución de un programa. Un bajo CPI implica que las instrucciones se ejecutan eficientemente, lo que resulta en un tiempo de ejecución más rápido. Por otro lado, un alto CPI indica que las instrucciones pueden estar tomando más ciclos de reloj para complementarse, lo que ralentiza la ejecución del programa (ver formula).

$$T_{cpu} = \frac{\#Instrucciones * CPI}{Tasa Reloj}$$

CONCLUSIÓN

El análisis de rendimiento y optimización es crucial para lograr un funcionamiento eficiente de los programas. Tomando en cuenta que la materia de curso que elegimos para la solución del problema es 'rendimiento', la evaluación en los tiempos de ejecución de un programa puede proporcionar información valiosa sobre como mejorar la eficiencia de un código. En el análisis realizado sobre el programa 'main.c', se registraron los tiempos de ejecución para diferentes palabras y orientaciones en una sopa de letras, se puede visualizar que las orientaciones tienen un impacto significativo en los tiempos de ejecución debido a la forma en que se accede y se procesa la información.

Al comparar los tiempos de ejecución, se observó que la palabra 'carne' en orientación vertical tuvo el mayor tiempo promedio de ejecución. Esto se debe a ubicación de la palabra en la sopa de letras y el tamaño de esta, lo que requiere un procesamiento más extenso.

Por otro lado, la orientación horizontal demostró tener tiempos de ejecución más bajos en comparación con la vertical. Esto se atribuye a la forma en que se manipula la información con un arreglo unidimensional cuando la orientación es horizontal.

La optimización del código se logró mediante la eliminación de operaciones redundantes y una mejor gestión del acceso a la información de los archivos texto. La implementación de estas mejoras resultó en una reducción significativa en los tiempos de ejecución.

En conclusión, mediante la aplicación de estrategias de optimización y la consideración de factores como la estructura interna y la gestión de memoria, es posible lograr mejoras en los tiempos de ejecución de los programas, lo que contribuye a la mejora global del rendimiento del programa.