Análisis de tiempos de ejecución en sopas de letras

Nombre: Nicolas Olivos *Rol: 202073507-7*

Nombre: Diego Morales *Rol: 202073511-5*

índice

Introducción

El rendimiento y la optimización son aspectos fundamentales en el desarrollo de aplicaciones y algoritmos, donde los tiempos de ejecución de programas se convierte en un factor crítico cuando se trata de resolver problemas que involucran búsquedas y procesamiento de datos. En este informe se aborda la búsqueda de palabras en sopas de letras utilizando el lenguaje de programación C, mostrando mediante el uso de tablas de comparación, los tiempos de ejecución del código para encontrar las palabras objetivo.

Preguntas y Análisis

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

Los resultados de la siguiente tabla fueron escritos en base a los resultados de la **imagen 5:**

PALABRA	ORIENTACION	TIEMPO [S]
hola	Vertical	0.000015
Gamer	Vertical	0.000020
gato	Vertical	0.000009
Carne	Vertical	0.000023
Jamon	Vertical	0.000014
carro	Vertical	0.000014
Cobre	Horizontal	0.000013
perro	Horizontal	0.000010
tapia	Horizontal	0.000009
banco	Horizontal	0.000012
casa	Horizontal	0.000008
viktor	Horizontal	0.000011

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

La palabra que tuvo un mayor tiempo de ejecución fue la palabra "Carne" como se puede observar en la **imagen 5**, las razones de poseer un mayor tiempo de ejecución implican varios factores tales como:

- Tamaño de la matriz: Mientras mas grande sea la matriz, implica buscar una palabra en una mayor cantidad de caracteres, realizando una mayor cantidad de instrucciones afectando al tiempo de ejecución. La palabra "Carne" se encuentra en una matriz de 200 x 200, siendo este la matriz mas grande de los ejemplos.

- Orientación: La búsqueda horizontal realiza iteraciones en la misma fila de la matriz, en cambio la búsqueda vertical debe ir iterando en filas distintas lo que genera un mayor costo de ejecución. La palabra "Carne" posee una orientación de búsqueda en vertical, esto implica que los saltos de filas requieren una mayor cantidad de instrucciones.

- Cantidad de caracteres: A mayor cantidad caracteres que posea la palabra a buscar, el programa debe iterar como mínimo la cantidad de caracteres que conforma la palabra.

En base a las razones anteriores, la palabra "Carne" tiene un mayor tiempo de ejecución, ya que, su matriz es de las mas grandes, posee una búsqueda en vertical y además pertenece a una de las palabras con mas caracteres, implicando que el programa a la hora de buscar "Carne" en la sopa de letras requiera iterar una mayor cantidad de instrucciones, provocando un mayor tiempo de ejecución.

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

Promedio de Tiempos de Ejecución por Orientación:

- Orientación Vertical: (0.000015 + 0.00002 + 0.000009 + 0.000023 + 0.000014 + 0.000014) / 6 = **0.0000167** segundos en promedio.

- Orientación Horizontal: (0.000013 + 0.00001 + 0.000009 + 0.000012 + 0.000008 + 0.000011) / 6 = **0.0000117** segundos en promedio.

La orientación que tuvo un menor tiempo de ejecución fue la Horizontal, esto se debe a como se implementó el código de búsqueda de palabras en la sopa de letras, tal y como muestra en la **imagen 1.** Donde la búsqueda horizontal busca la palabra en la misma fila de la matriz, y la vertical debe cambiar de filas para encontrar la palabra objetivo, lo que significa un mayor costo en tiempos de ejecución. Complementando lo anterior con la materia aprendida y en base a la **imagen 2**, a mayor cantidad de instrucciones el tiempo de ejecución será mayor.

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

Para minimizar los tiempos de ejecución en la búsqueda de palabras, se redujeron la cantidad de instrucciones en la búsqueda horizontal y vertical, para ello analizando el código de la **imagen 1**, se puede observar que este contiene un printf y una variable llamada found (saber si encuentra o no la palabra). En un principio estos parecen ser innecesarios, al fin y al cabo, son instrucciones extras para la ejecución, por lo tanto, se decidió quitar ambas instrucciones en ambas orientaciones y a la vez crear una única función aparte para la búsqueda de palabras considerando su orientación, tal como se pude ver en la **imagen 3** con la función "encontrar_palabra". Como se modificó el código para crear una función aparte para la búsqueda, el llamado de esta última en el código main se puede observar en la **imagen 4.**

Al comparar el código inicial de la **imagen 1** con el código optimizado de la **imagen 3 y 4**, la variable found paso a ser el return de la función de búsqueda para generar los prints en el main, esta optimización tiene un impacto positivo en los tiempos de ejecución, ya que reduce la necesidad de seguir rastreando y actualizando la variable found dentro del ciclo de búsqueda que teníamos inicialmente.

La función printf puede requerir más tiempo si utilizamos formatos de cadena complicados, ya que tiene que analizar y formatear la cadena (printf de la **imagen 1** que contiene un %d y una suma) antes de imprimir. Por lo tanto, en el código optimizado (**imagen 4**) los únicos printf se encuentran después que haya terminado el tiempo de ejecución de una palabra.

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.

Los resultados de la siguiente tabla fueron escritos en base a los resultados de la **imagen 6:**

PALABRA	ORIENTACION	TIEMPO [S]
hola	Vertical	0.000005
Gamer	Vertical	0.000007
gato	Vertical	0.000001
Carne	Vertical	0.000012
Jamon	Vertical	0.000005
carro	Vertical	0.000006
Cobre	Horizontal	0.000004
perro	Horizontal	0.000001
tapia	Horizontal	0.0000001
banco	Horizontal	0.000004
casa	Horizontal	0.000001
viktor	Horizontal	0.000002

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

Con la materia que hemos visto hasta el momento en clases, sabemos que los componentes de hardware afectan al tiempo de respuesta ante la ejecución de algún programa, ahora bien, viendo las unidades que pasaremos a lo largo de “Sistemas Operativos” creemos que la unidad de Memoria caché puede solventar en cierto grado los tiempos de ejecución, dado que la memoria cache es parte del procesador facilitando el acceso entre la CPU y RAM acelerando el intercambio de datos.

Conclusión

Mediante el trabajo realizado podemos destacar la importancia del rendimiento y la optimización que conlleva un desarrollo de algoritmos y secuencias. Esto cobra cierta relevancia en tareas como la búsqueda de palabras en una sopa de letras, donde el tiempo de ejecución se convierte en un factor decisivo para el procesamiento eficiente de datos. Identificando a su vez, factores que influyen en el tiempo de ejecución de este proceso. Entre estos factores se incluyen el tamaño de la matriz, la orientación de búsqueda y la longitud de la palabra. Por lo tanto, siempre se deben tener en consideración aquellos factores que puedan afectar quizás no en gran medida al tiempo de búsqueda, pero si a gran escala con millones de datos, generando una apertura a la optimización de código y algoritmos.

Anexo

```
if (strcmp(direccion, "HORIZONTAL\n") == 0) {
    start = clock();
    for (int i = 0; i < rows; i++) {
        if (strstr(matrix[i], nombre_archivo) != NULL) {
            printf("Palabra encontrada en la fila %d\n", i + 1);
            found = 1;
            break;
        }
    }
} else if (strcmp(direccion, "VERTICAL\n") == 0) {
    start = clock();
    for (int j = 0; j < cols; j++) {
        char col[MAX_SIZE];
        for (int i = 0; i < rows; i++) {
            col[i] = matrix[i][j];
        }
        col[rows] = '\0';

        if (strstr(col, nombre_archivo) != NULL) {
            printf("Palabra encontrada en la columna %d\n", j + 1);
            found = 1;
            break;
        }
    }
}
end = clock();
```

***Imagen 1**: Extracto de código de búsqueda de palabra.*

$$T_{CPU} = \frac{\#\text{Instrucciones x CPI}}{TasaReloj}$$

***Imagen 2**: Formula del tiempo de ejecución de la CPU.*

```c
int encontrar_palabra(const char *direction, char matrix[MAX_SIZE][MAX_SIZE], int rows, int cols, const char *nombre_archivo) {
    if (strcmp(direction, "HORIZONTAL\n") == 0) {
        for (int i = 0; i < rows; i++) {
            if (strstr(matrix[i], nombre_archivo) != NULL) {
                return 1;
            }
        }
    } else if (strcmp(direction, "VERTICAL\n") == 0) {
        for (int j = 0; j < cols; j++) {
            char col[MAX_SIZE];
            for (int i = 0; i < rows; i++) {
                col[i] = matrix[i][j];
            }
            col[rows] = '\0';

            if (strstr(col, nombre_archivo) != NULL) {
                return 1;
            }
        }
    }

    return 0;
}
```

***Imagen 3**: Búsqueda de palabras con código optimizado.*

```c
start = clock();
int search = encontrar_palabra(direccion,matrix,rows,cols,nombre_archivo);
end = clock();

if (search == 1){
    printf("Palabra encontrada!\n");
}else {
    printf("Palabra no encontrada\n");
}
```

***Imagen 4**: Llamado de la función de búsqueda en el código main.*

```
Leyendo archivo: hola
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 50
Tiempo de CPU utilizado: 0.000015 segundos
tamano de la matriz 50 x 50
Carpeta '50x50' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: Gamer
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 53
Tiempo de CPU utilizado: 0.000020 segundos
tamano de la matriz 100 x 100
Carpeta '100x100' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: Cobre
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 145
Tiempo de CPU utilizado: 0.000013 segundos
tamano de la matriz 200 x 200
Carpeta '200x200' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: perro
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 25
Tiempo de CPU utilizado: 0.000010 segundos
tamano de la matriz 50 x 50
Carpeta '50x50' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: Jamon
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 33
Tiempo de CPU utilizado: 0.000014 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: viktor
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 73
Tiempo de CPU utilizado: 0.000011 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: carro
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 33
Tiempo de CPU utilizado: 0.000014 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: gato
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 8
Tiempo de CPU utilizado: 0.000009 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.

Leyendo archivo: Carne
Orientacion de la palabra: VERTICAL
Palabra encontrada en la columna 50
Tiempo de CPU utilizado: 0.000023 segundos
tamano de la matriz 200 x 200
Carpeta '200x200' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: tapia
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 10
Tiempo de CPU utilizado: 0.000009 segundos
tamano de la matriz 100 x 100
Carpeta '100x100' creada exitosamente.
Archivo movido exitosamente.

Leyendo archivo: banco
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 176
Tiempo de CPU utilizado: 0.000012 segundos
tamano de la matriz 200 x 200
La carpeta '200x200' ya existe.
Archivo movido exitosamente.

Leyendo archivo: casa
Orientacion de la palabra: HORIZONTAL
Palabra encontrada en la fila 1
Tiempo de CPU utilizado: 0.000008 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.
```

***Imagen 5**: Salida en consola del código inicial.*

```
Leyendo archivo: hola
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000005 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.

Leyendo archivo: Gamer
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000007 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: Cobre
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000004 segundos
tamano de la matriz 200 x 200
La carpeta '200x200' ya existe.
Archivo movido exitosamente.

Leyendo archivo: perro
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000001 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.

Leyendo archivo: gato
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000001 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.
```

```
Leyendo archivo: Carne
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000012 segundos
tamano de la matriz 200 x 200
La carpeta '200x200' ya existe.
Archivo movido exitosamente.

Leyendo archivo: tapia
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000000 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: banco
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000004 segundos
tamano de la matriz 200 x 200
La carpeta '200x200' ya existe.
Archivo movido exitosamente.

Leyendo archivo: casa
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000001 segundos
tamano de la matriz 50 x 50
La carpeta '50x50' ya existe.
Archivo movido exitosamente.

Leyendo archivo: Jamon
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000005 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.
```

```
Leyendo archivo: viktor
Orientacion de la palabra: HORIZONTAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000002 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.

Leyendo archivo: carro
Orientacion de la palabra: VERTICAL
Palabra encontrada!
Tiempo de CPU utilizado: 0.000006 segundos
tamano de la matriz 100 x 100
La carpeta '100x100' ya existe.
Archivo movido exitosamente.
```

***Imagen 6**: Salida en consola del código optimizado.*