Análisis Laboratorio 1

Integrantes:	Rol:
Benjamin Pavez	202173628-K
Diego Cisternas	202173521-6

Grupo: 8

Fecha: 20/08/2023

Preguntas y Respuestas

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

   Respuesta:

Archivo	Orientación	Tiempo[s]
carro.txt	vertical	0.001554
hola.txt	Vertical	0.000128
Cobre.txt	horizontal	0.002713
Jamon.txt	vertical	0.000802
casa.txt	horizontal	0.000002
banco.txt	horizontal	0.003939
Carne.txt	vertical	0.005077
gato.txt	vertical	0.000411
perro.txt	horizontal	0.000008
viktor.txt	horizontal	0.000533
tapia.txt	horizontal	0.000007
Gamer.txt	vertical	0.001390

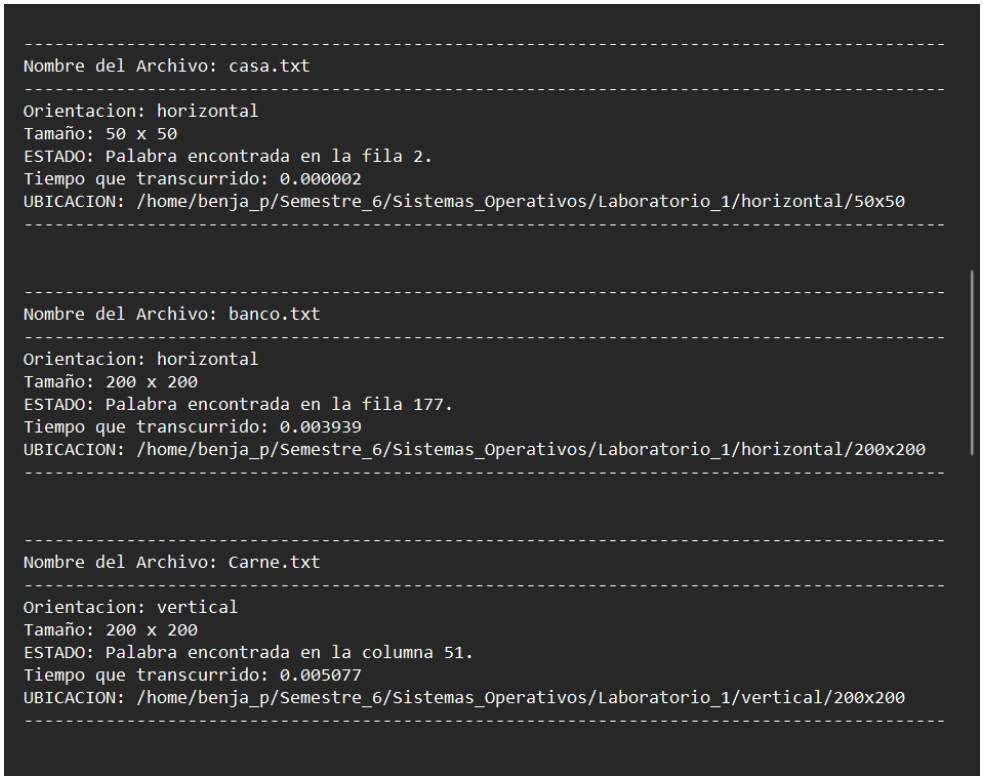

Figura 1: Captura de archivo resultante para pruebas de eficiencia (es posible que al ejecutar cambien los valores, los mostrados son realizados en la primera ejecución del programa)

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

Respuesta: La palabra con mayor tiempo de ejecución fue carne, ya que su tiempo fue de 0.005077 segundos como muestra la Figura 2, esto se debe a que, al estar la palabra en vertical, implicó la necesidad de realizar la matriz traspuesta a la sopa de letras, con el fin de manipularla como una sopa horizontal, siendo mas sencillo verificar si la palabra esta o no en cada fila de la sopa, además su dimensión de 200 x 200 aumenta el tiempo para calcular la traspuesta.

```
--------------------------------------------------------------------------------------------------
Nombre del Archivo: Carne.txt
--------------------------------------------------------------------------------------------------
Orientacion: vertical
Tamaño: 200 x 200
ESTADO: Palabra encontrada en la columna 51.
Tiempo que transcurrido: 0.005077
UBICACION: /home/benja_p/Semestre_6/Sistemas_Operativos/Laboratorio_1/vertical/200x200
--------------------------------------------------------------------------------------------------
```

Figura 2 Captura de archivo resultante para pruebas de tiempo que muestra la palabra que más se tardó en encontrar en la sopa de letras.

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

Respuesta: La orientación que tuvo menor tiempo fue la horizontal, puesto que cuando se recorre cada sopa, se obtiene cada línea de esta, lo que facilita el buscar la palabra, puesto que solo basta con verificar que la subcadena este dentro de cada línea de la sopa, reduciendo el tiempo y costo de la ejecución ya que como se mencionó en la pregunta anterior, para verificar que cada la palabra este en una sopa vertical hay que invertir la sopa para manejarla como una horizontal.

```
206     if(strcasecmp(orientation, "horizontal") == 0) {
207         //Horizontal
208         tam = (tam+1)/2; //Dimension real
209         char cadenafinal[500];
210 
211         while(fgets(bufer, sizeof(bufer), archivo)){
212             quitaespacios(bufer, cadenafinal);
213             //Para probar la optimizacion, reemplace strstr(cadenafinal, nom_sopa) != NULL por strstrmodificada(cadenafinal, nom_sopa) != NULL
214             if(strstr(cadenafinal, nom_sopa) != NULL){
215                 //Se calcula el tiempo de ejecucion
216                 clock_t end = clock();
217                 time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
218 
219                 //Verificacion de la busqueda
220                 printf("ESTADO: Palabra encontrada en la fila %d.\n",fila+2);
221                 printf("Tiempo que transcurrio: %f\n",time_spent);
222                 found = true;
223                 moverArchivo(nom_archivo,orientation ,tam);
224                 printf("--------------------------------------------------------------------------------------------------\n");
225                 printf("\n");
226                 printf("\n");
227                 break;
228             }
229             fila++;
230         }
231 
232         if(found == false){
233             clock_t end = clock();
234             time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
235 
236             //Verificacion de la busqueda
237             printf("ESTADO: Palabra NO encontrada.\n");
238             printf("Tiempo que transcurrio: %f\n",time_spent);
239             printf("--------------------------------------------------------------------------------------------------\n");
240             printf("\n");
241             printf("\n");
242         }
```

Figura 3: Captura del código que muestra lo simplificado que es buscar una palabra en horizontal.

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

   Respuesta: Para optimizar el código con el fin de reducir los tiempos de ejecución se podría reemplazar la función strstr( ), ya que esta con cadenas muy grandes el rendimiento baja por lo que se realizó otra función que la remplaza con el fin de tratar de reducir los tiempos con las sopas de mayor dimensión.

```
143  //OPTIMIZACION
144  /*
145  La funcion remplaza la funcion strstr para tratar de reducir los tiempos que tarda encontrar la palabra
146
147  Parametros :
148     const char *cadena1 : puntero tipo char que apunta a la linea de la sopa de letras
149     const char *cadena2 : puntero tipo char que apunta al nombre a buscar en la linea
150
151  Retorno :
152     Retona NULL si no esta o la linea donde aparecio la palabra
153
154  */
155  char *strstrmodificada(const char *cadena1, const char *cadena2) {
156      if (*cadena2 == '\0') {
157          return (char *)cadena1;
158      }
159      while(*cadena1) {
160          const char *carac1 = cadena1;
161          const char *carac2 = cadena2;
162          while(*carac2 && (*carac1 == *carac2)) {
163              carac1++;
164              carac2++;
165          }
166          if(*carac2 == '\0') {
167              return (char *)cadena1;
168          }
169          cadena1++;
170      }
171      return NULL;
172  }
```

Figura 4: Captura del código que muestra la optimización realizada a la función strstr, que consiste en otra función que realiza lo mismo pero que trata de reducir los tiempos en buscar la 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.

Respuesta:

Archivo	Orientación	Tiempo[s]
carro.txt	vertical	0.000347
hola.txt	Vertical	0.000118
Cobre.txt	horizontal	0.000182
Jamon.txt	vertical	0.000345
casa.txt	horizontal	0.000001
banco.txt	horizontal	0.000223
Carne.txt	vertical	0.001304
gato.txt	vertical	0.000118
perro.txt	horizontal	0.000011
viktor.txt	horizontal	0.000058
tapia.txt	horizontal	0.000007
Gamer.txt	vertical	0.000346

```
--------------------------------------------------------------------------------------------
Nombre del Archivo: casa.txt
--------------------------------------------------------------------------------------------
Orientacion: horizontal
Tamaño: 50 x 50
ESTADO: Palabra encontrada en la fila 2.
Tiempo que transcurrido: 0.000001
UBICACION: /home/benja_p/Semestre_6/Sistemas_Operativos/Laboratorio_1/horizontal/50x50
--------------------------------------------------------------------------------------------


--------------------------------------------------------------------------------------------
Nombre del Archivo: banco.txt
--------------------------------------------------------------------------------------------
Orientacion: horizontal
Tamaño: 200 x 200
ESTADO: Palabra encontrada en la fila 177.
Tiempo que transcurrido: 0.000223
UBICACION: /home/benja_p/Semestre_6/Sistemas_Operativos/Laboratorio_1/horizontal/200x200
--------------------------------------------------------------------------------------------


--------------------------------------------------------------------------------------------
Nombre del Archivo: Carne.txt
--------------------------------------------------------------------------------------------
Orientacion: vertical
Tamaño: 200 x 200
ESTADO: Palabra encontrada en la columna 51.
Tiempo que transcurrido: 0.001304
UBICACION: /home/benja_p/Semestre_6/Sistemas_Operativos/Laboratorio_1/vertical/200x200
--------------------------------------------------------------------------------------------
```

Figura 5: Captura de archivo resultante para pruebas de eficiencia con la optimización (es posible que al ejecutar cambien los valores, los mostrados son realizados en la primera ejecución del programa)

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

Respuesta: Como grupo nosotros creemos que la materia que nos podría ayudar es la del rendimiento, ya que hay que considerar que las pruebas de la tarea se llevaron a cabo en un maquina virtual, por lo que Linux contaba con un hardware limitado al que poseía el equipo, es por ello que vemos necesario entender que al limitar el procesador este tarda más tiempo en realizar las tareas, por otra parte creemos que nos seria de utilidad la unidad 8, ya que nos explicaría la gestión de la memoria principal, tratando de optimizar la creación y liberación de arrays del heap para que se reduzca el tiempo de la ejecución.

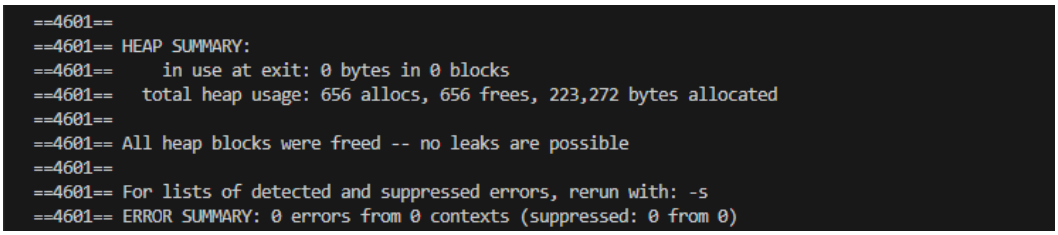

Figura 6: Captura de terminal al ejecutar valgrind al programa, en el se puede apreciar que no hay fugas de memoria ni algún error

Conclusión

En resumen, a lo largo de este informe hemos logrado comprender la relevancia de contar con un código optimizado en comparación a uno que no lo esté. También hemos observado cómo este aspecto llega a tener un impacto significativo en la resolución del problema presentado, llegando a reducir varios segundos en la búsqueda de la solución. Asimismo, hemos notado su influencia en la utilización de la CPU, ya que los testeos del código no optimizado requieren un mayor uso del procesador cuando el código optimizado no.

Estos resultados nos indican que al desarrollar programas no basta con resolver un problema en sí, sino que también debemos asegurar que la solución sea eficiente, aún más cuando los recursos del hardware son limitados.