



Laboratorio 1

Integrantes:

Benjamin Pavez

Diego Cisternas

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

```
-----
Nombre del Archivo: casa.txt
-----
Orientacion: horizontal
Tamaño: 50 x 50
ESTADO: Palabra encontrada en la fila 2.
Tiempo que transcurrido: 0.000002
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: 0.003939
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: 0.005077
UBICACION: /home/benja_p/Semestre_6/Sistemas_Operativos/Laboratorio_1/vertical/200x200
-----
```

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.

```
155     if(strcasecmp(orientation, "horizontal") == 0) {
156         //Horizontal
157         char cadenafinal[500]; // Define un buffer para cadenafinal
158         while(fgets(bufer, sizeof(bufer), archivo)){
159             quitaespacios(bufer, cadenafinal);
160             if(strstr(cadenafinal, nom_sopa) != NULL){
161                 //Se calcula el tiempo de ejecucion
162                 clock_t end = clock();
163                 //calcula el tiempo transcurrido encontrando la diferencia (end - begin) y
164                 //dividiendo la diferencia por CLOCKS_PER_SEC para convertir a segundos
165                 time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
166                 //Verificación, RECORDAR ELIMINAR AL ENVIAR
167                 printf("ESTADO: Palabra encontrada en la fila %d.\n",fila+2);
168                 printf("Tiempo que transcurrido: %f\n",time_spent);
169                 printf("-----\n");
170                 printf("\n");
171                 printf("\n");
172                 found = true;
173                 break;
174             }
175             fila++;
176             memset(cadenafinal, 0, 500); //limpiar para que no lo una con la basura
177         }
178         memset(cadenafinal, 0, 500); //limpiar para que no lo una con la basura
179         fclose(archivo);
```

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 reemplaza con el fin de reducir los tiempos con las sopas de mayor dimensión.

```
142
143 //OPTIMIZACION
144 /*
145 La funcion reemplaza la funcion strstr pa
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     Retorna 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 *h = cadena1;
161         const char *n = cadena2;
162         while (*n && (*h == *n)) {
163             h++;
164             n++;
165         }
166         if (*n == '\0') {
167             return (char *)cadena1;
168         }
169         cadena1++;
170     }
171     return NULL;
172 }
173
```

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	0.000001
casa.txt	horizontal	0.000002
banco.txt	horizontal	0.000223
Carne.txt	vertical	0.001304
gato.txt	vertical	0.000118
perro.txt	horizontal	0.000011
viktor.txt	horizontal	0.000007
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.