UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIA
DIVISION DE ELECTRONICA Y COMPUTACION

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE TRADUCTORES DE LENGUAJE I (I7026)

Reporte de Práctica 7

Practica 7: " Manejo de archivos"

Alumno: Oswaldo Luna Grados
Código: 211718256

Sección: D07

Profesor: Valentín Martínez López

Fecha 7 de noviembre del 2018
firma de revisado

Objetivo: Escribir un programa que sea capaz de leer un archivo de texto del que debe determinar el número de caracteres de cada línea, el programa debe crear un archivo nuevo en el que al inicio de cada línea indique el número de caracteres que existen hasta el momento.

Puede usarse interrupción 21 o funciones de librerías.

Suponga que el archivo de texto contiene lo siguiente:

hola que tal
como has estado
esto es una prueba
para contar caracteres.
Se deberá crear un archivo de texto como el que sigue:

000 hola que tal
012 como has estado
027 esto es una prueba
045 para contar caracteres.
068
En la primera línea del archivo de salida se imprime al principio 000 por que no se han contado ningún carácter.
La primera línea tiene 12 caracteres por lo que en archivo de salida la segunda línea empieza con 012
La segunda línea tiene 15 caracteres al sumarse con los caracteres de la primera línea el resultado es 27 y se imprimen en la tercera línea y esto se repite hasta obtener la cantidad total de caracteres del archivo.

Un archivo informático está identificado por un nombre y una descripción, el cual almacena información en formato binario (es decir ceros y unos). En lenguajes de alto nivel manejan los grupos de información (archivos), escondiendo la complejidad sobre el manejo y compilación de los mismos. En lenguajes de alto nivel la manipulación de archivos se reduce a taras simples como por ejemplo, creación, lectura, escritura. En lenguaje ensamblador, la manipulación de archivo requiere de mayor detalle.

Lo primero es leer el archivo
El cual se llama file.txt y lo máximo de caracteres que lera el archivo es 110 y lo guardara en la cadena llamada leído y cerramos el archivo

```
read:
mov dx, offset filename
mov ah, 3dh
int 21h
jc error
mov handle,ax
mov bx,handle
 mov cx, 110 ;maximo de caracteres en el archivo
mov dx, offset leido
mov ah, 3fh
int 21h

                ;close file
mov bx, handle
mov ah, 3eh
int 21h
```

Hice que imprimiera en pantalla la cadena leída y guardada en leído.

```
print:
 ;imprimir el contenido de leido
 mov dx, offset leido
 mov ah, 9
 int 21h
```

Después empezamos el contar los caracteres por renglón inicializando los registros en cero luego ponemos 0 puesto que no se ha leído nada aun.

```
cont:
xor dx,dx             ;contar caracteres
xor ax,ax
xor bx,bx
xor cx,cx
mov cx, 110            ;maximo de caracteres a leer
xor si,si
xor di,di

mov string[di],'0'
inc ap1
mov di,ap1
mov string[di],' '
inc ap1
```

En recorrer lo que hace es poner en los registros di y si la posición actual de las cadenas leídas y escrita con el número de caracteres. También revisa si hay un salto de carácter o final de la cadena leída si no es ninguna de las otras va a contcarac donde se pone en contChar el número en hexadecimal de caracteres contados actualmente. Y vuelve a seguir donde se pasa el carácter actual y después incrementa la posición de cada cadena y vuelve a recorre

```
recorre:
mov si,ap
mov di,ap1
cmp leido[si],'$'
jz  fin
cmp leido[si],10
jz  suma
jnz contcarac
seguir:
mov al,leido[si]
mov string[di],al
seguir2:
inc ap
inc ap1
loop recorre

contcarac:
mov bx,ap
inc contChar
call seguir
```

Si encuentra salto de línea va a suma donde pone el valor actual de caracteres contados en ascii en la cadena escrita.

```
suma:
mov di,ap1
mov string[di],10
inc ap1
call hex2ascii              ;agrega la cantidad de caracteres
mov di,ap1
mov string[di],' '
call seguir2
```

Hex2ascii hace que devuelva el valor en decimal.

```
hex2ascii proc
    xor ax,ax
    mov al, contChar
    mov bx, 0ah
    div bl
    mov bl, al
    add al, 48
    mov di,ap1
    mov string[di],al
    inc ap1
    xor ax,ax
    mov al, bl
    mov bl, 0ah
    mul bl
    mov bx, ax
    mov al, contChar
    sub ax, bx
    add al, 48
    mov di,ap1
    mov string[di],al
    inc ap1
    ret
```

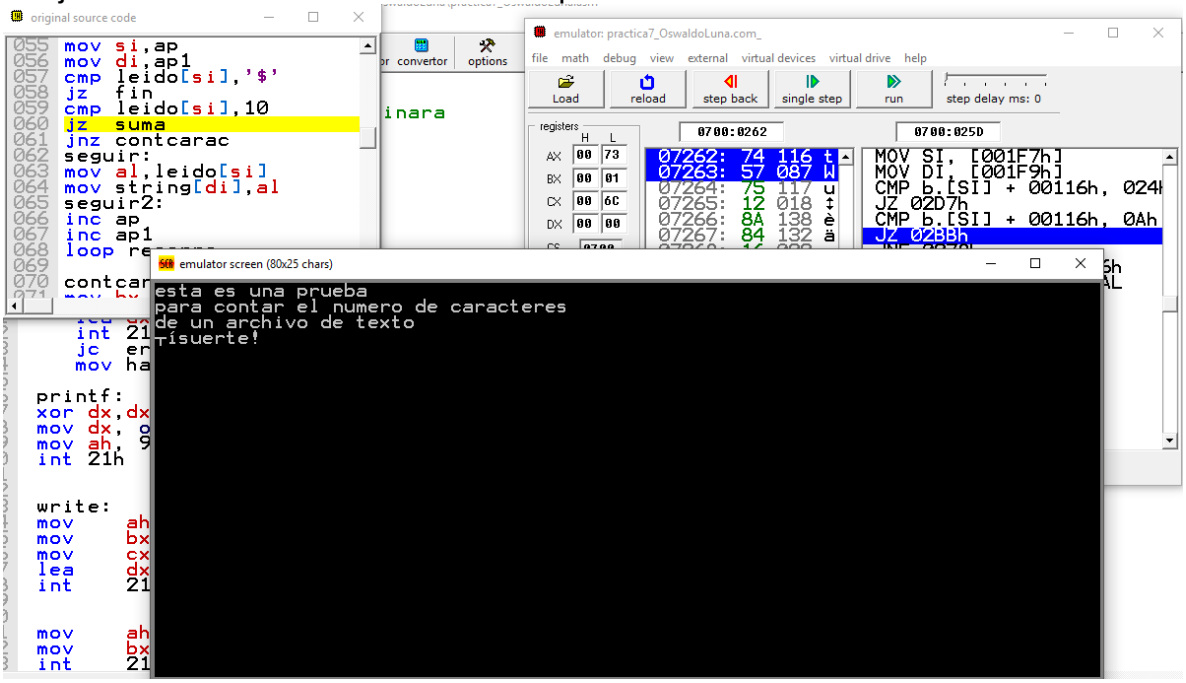
En cuanto encuentre el final de la cadena leída guardara en la cadena escrita el número de caracteres después de dar un salto de renglón.

```
fin:
; se sumara el total y terminara
mov string[di],13
inc ap1
mov di,ap1
mov string[di],10
inc ap1
mov di,ap1
call hex2ascii
mov di,ap1
```

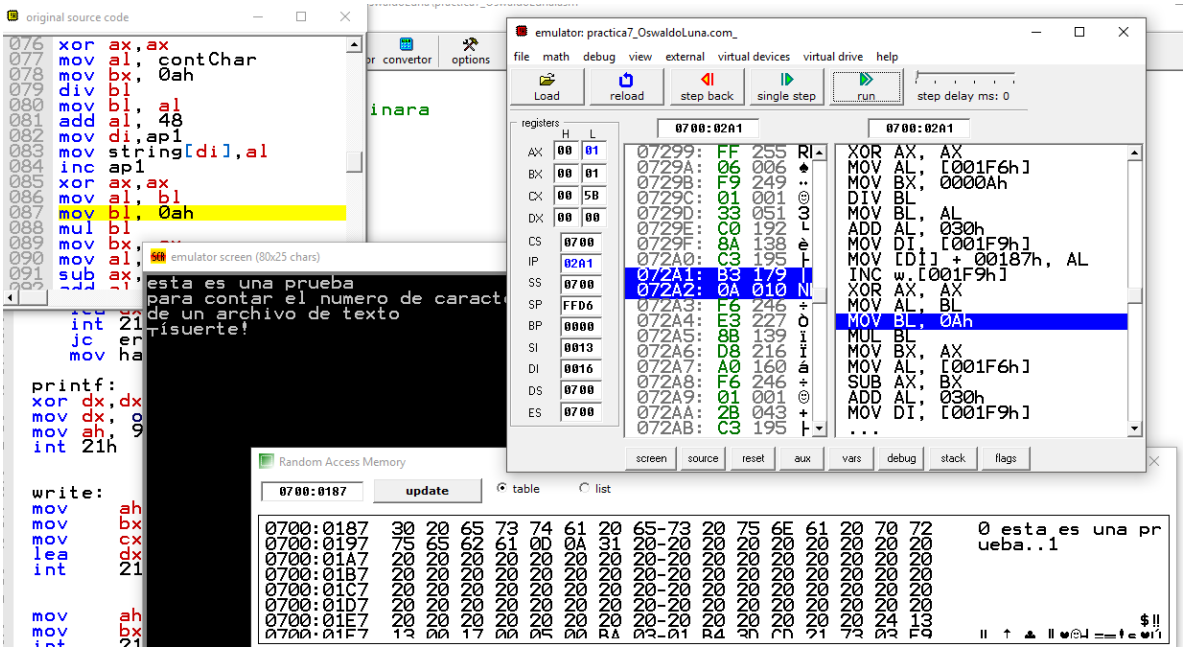
Ya que tenemos la cadena escrita con el número de caracteres por renglón creamos un nuevo archivo llamado file2.txt
Imprimimos la nueva cadena en pantalla, escribimos en el archivo la cadena escrita y cerramos el archivo.

```
; create file
create:
    mov ah,3ch
    mov cx,00000000b
    lea dx,filename2
    int 21h
    jc  error
    mov handle,ax

printf:
xor dx,dx                   ;print
mov dx, offset string
mov ah, 9
int 21h


write:
mov     ah,40h
mov     bx,handle
mov     cx,110
lea     dx,string
int     21h

            ; close file
mov     ah,3eh
mov     bx,handle
int     21h
error:

int 20h
```

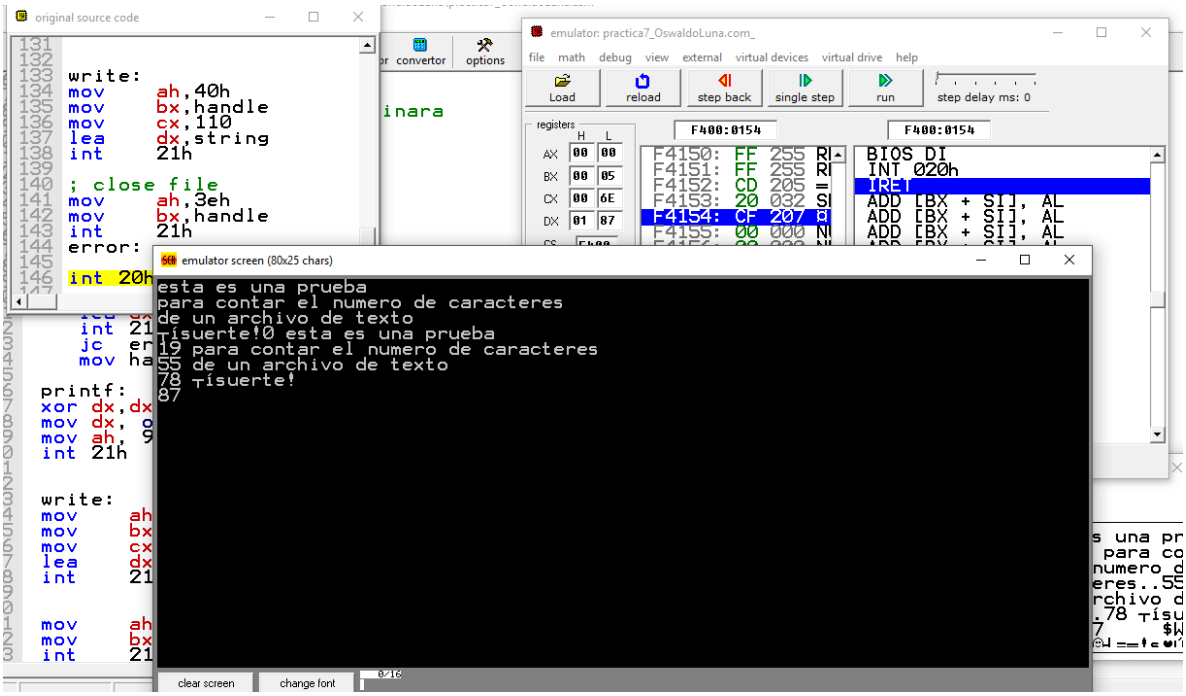
Al ejecutar muestra la cadena leída en pantalla



Después empieza a guardar en la cadena donde se escribe para crear el nuevo archivo.



Y termina guardando en un nuevo archivo e imprimiendo el resultado en pantalla



Conclusión

El manejo de archivos es la interrupción del procesador para leer en la memoria secundaria y obtener la información de ahí la cual se puede modificar eliminar copiar etc. El cual el objetivo es leer dentro del archivo y analizar la cantidad de caracteres antes que aparezca un salto de línea es decir un 13 en hexadecimal. Logrando así poder reconocer cuantos caracteres hay en cada renglón aunque a final de cuentas todo el archivo no tiene saltos de renglón sino que por ascii parece que existe el salto de renglón aunque todo está en un mismo arreglo. También se comprendió más el manejo de cadenas y como moverse en esta y agregar en esta parte por parte.