**PONTIFICIA UNIVERSIDAD CATOLICA MADRE Y MAESTRA**

****

**Nombre:**

Félix Alejandro Guzmán           2014 - 0565

**Materia:**

Arquitectura Computacional

**Profesor:**

Alvaro A. Reyes P.

**Practica sobre:**

Práctica #5

**Fecha de Entrega:**

viernes, 24 de noviembre de 2017

# Código

section .bss

opción resb 1 ;Para guardar Archivo el resultado del menú

handler resb 4 ;Para guardar Archivo el handler

find resb 1 ;Para guardar Archivo la find a reemplazar

longF resw 1 ;Para guardar el tamaño de la find

replace resb 1 ;Para guardar Archivo la find por la cual reemplazar

longR resw 1 ;Para guardar el tamaño de la find

input resb 7900 ;Para guardar Archivo el archivo

length resw 1 ;Para guardar Archivo el tamaño del archivo

output resb 10000 ;Para guardar el archivo a reemplazar

tam resw 1 ;Para guardar el tamaño del archivo a reemplazar

section .data

mensaje: db 'E to exit, R to replace: ',0 ;String

mensaje\_size: equ $-mensaje ;Size del string

mensajeFind: db 'find a buscar: ',0 ;String

mensajeFind\_size: equ $-mensajeFind ;Size del string

mensajeReplace: db 'find a reemplazar: ',0 ;String

mensajeReplace\_size: equ $-mensajeReplace ;Size del string

mensajesalto: db 0ah,0 ;String

mensajesalto\_size: equ $-mensajesalto ;Size del string

filename db "./Archivo.txt", 0 ;Ruta del archivo donde Guardar Archivo

file db '/home/felixal/Documents/Arq/tarea2.asm',0 ;Dirección del archivo

global \_start

\_start:

mov eax,5 ;Abrir el archivo

mov ebx,file ;Nombre del archivo

mov ecx,0 ;Para solo lectura

int 80h

mov [handler],eax ;GuardarArchivo el handler si todo es correcto

mov eax,3 ;Leer del archivo

mov ebx,[handler] ;Tomo el handler

mov ecx,input ;Coloco el archivo en el input

mov edx,length ;Leo la cantidad de bytes

int 80h

mov eax,4 ;Escribir en pantalla

mov ebx,1

mov ecx,input

mov edx,length

int 80h

call Next

call GuardarArchivo

Display:

mov eax,4     ;Escribir en pantalla

mov ebx,1

mov ecx,mensaje

mov edx,mensaje\_size

int 80h

mov eax,3     ;Leer de pantalla

mov ebx,0

mov ecx,opcion

mov edx,1

int 80h

mov cl,52h ;Tecla R

mov al,45h ;Tecla E

mov edx,opcion ;La tecla opcionada

mov dl,[edx] ;Acceder al registro

cmp dl,cl ;La tecla es igual que R

je Replace ;Función de reemplazar

cmp dl,al ;La tecla es igual que E

jne Display ;Repite el menú

Fin:

call Next

mov eax,6 ;Cierra el archivo

mov ebx,[handler] ;El handler del archivo

int 80h

mov eax,1 ;Cierra el programa

mov ebx,0

int 80h

section .text

longitud:

mov eax, ebx ;Tomo el valor del ebx

lop:

cmp byte [eax], 0 ;Es el valor que leo nulo?

jz salte

inc eax ;Incremento el puntero

jmp lop

salte:

sub eax, ebx ;Resto el puntero inicial de la posición actual

dec eax ;Decremento el valor para reflejar el valor real

ret

GuardarArchivo:

mov eax, 8 ;Para abrir el archivo

mov ebx, filename ;El nombre del archivo

mov ecx, 0777 ;Permisos de accesibilidad del archivo

int 0x80

mov ebx, eax ;Guardo el registro resultante para continuar

mov eax, 4 ;Imprime el input en el archivo

mov ecx, input

mov edx, length

int 0x80

mov eax, 6 ;Cerrar el archivo

mov eax, 1

int 0x80

ret

Next:

mov eax,4     ;Imprimir en pantalla

mov ebx,1

mov ecx,mensajesalto

mov edx,mensajesalto\_size

int 80h

ret

Replace:

mov eax,3     ;Pausa para continuar

mov ebx,0

mov ecx,find

mov edx,1

int 80h

mov eax,4    ;Imprimir en pantalla

mov ebx,1

mov ecx,mensajeFind

mov edx,mensajeFind\_size

int 80h

mov eax,3     ;Leer de pantalla la find a buscar

mov ebx,0

mov ecx,find

mov edx,1

int 80h

mov eax,4    ;Escribir en pantala

mov ebx,1

mov ecx,mensajeReplace

mov edx,mensajeReplace\_size

int 80h

mov eax,3     ;Leer de pantalla la find a reemplazar

mov ebx,0

mov ecx,replace

mov edx,1

int 80h

mov ebx,find ;Calcular el length de la find, paso la find por parametro

call longitud ;Función ara calcular la longitud

mov [longF],eax ;Guardo el resultado en la variable correspondiente

dec word [longF] ;Actualizo el valor

xor ebx,ebx ;Limpio las variables

mov ebx,replace ;Calcular el length de la find, paso la find por parametro

call length ;Función ara calcular la longitud

mov [longR],eax ;Guardo el resultado en la variable correspondiente

dec word [longR] ;Actualizo el valor

xor ebx,ebx ;Limpio las variables

mov ebx,input ;Calcular el length de la find, paso la find por parametro

call length ;Función ara calcular la longitud

mov [length],eax ;Guardo el resultado en la variable correspondiente

dec word [length] ;Actualizo el valor

mov eax, input ;Paso el buffer viejo a registro para referenciarlo

mov ebx, find ;Paso la find a registro para referenciarlo

mov ecx, output ;Paso el buffer nuevo a registro para referenciarlo

mov sp, [length] ;Paso el tamaño de buffer viejo a registro para referenciarlo

xor dx,dx ;Inicializo el puntero del tamaño de la find

xor cx,cx ;Inicializo el puntero del tamaño del buffer

ciclo:

mov al, [eax] ;Tomo referencia del valor del buffer viejo

mov dl, [ebx] ;Tomo referencia del valor de la find

mov cl, [ecx] ;Tomo referencia del valor del buffer nuevo

mov [ecx], al ;Muevo el valor del buffer viejo al nuevo

cmp cx, [longF] ;Es el puntero de la find igual que la longitud de la misma?

je cont ;Si es así replace la find

cmp al,dl ;Es el valor actual igual que el valor de la find?

jz aumento ;Si es así aumento los registros

jne restart ;Si no reinicio los registros

cmp sp, dx ;Es el puntero del buffer igual que la longitud del mismo?

call GuardarArchivo ;Termino

aumento:

inc eax ;Aumento el buffer viejo

inc ebx ;Aumento la find

inc ecx ;Aumento el buffer nuevo

inc dx ;Aumento el puntero del buffer viejo

inc cx ;Aumento el puntero de la find

inc cl ;Aumento el puntero del buffer nuevo

jmp ciclo ;Vuelvo al loop principal

restart:

mov ebx, find ;Reinicio el registro de la find

xor cx, cx ;Reinicio el puntero de la find

call ciclo ;Vuelvo al loop principal

cont:

sub cl, [longF] ;Coloco el puntero del buffer nuevo en la posición correcta

mov edx, replace ;Inicializo el registro para recorrer el replace

xor di,di ;Inicializa el puntero del replace

sigue:

mov bx,[edx] ;Coloco el valor del buffer en el registro

cmp di,[longR] ;Me encuentro en el final del replace?

je ciclo ;Vuelvo al loop principal

mov [ecx],bx ;Muevo el valor del replace al nuevo buffer

inc di ;Incremento el puntero del replace

inc cl ;Incremento el puntero del buffer nuevo

call sigue ;Repito ;Fin