section .bss

seleccion resb 1 ;Para guardar el resultado del menú

descriptor resb 4 ;Para guardar el handler

palabra resb 10 ;Para guardar la palabra a reemplazar

longP resw 1 ;Para guardar el tamaño de la palabra

reemplazo resb 10 ;Para guardar la palabra por la cual reemplazar

longR resw 1 ;Para guardar el tamaño del reemplazo

buffer resb 8144 ;Para guardar el archivo

len resw 1 ;Para guardar el tamaño del archivo

newBuffer 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

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

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

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

mensajeReplace: db 'Palabra 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 "./output.txt", 0 ;Ruta del archivo donde guardar

global \_start

\_start:

mov eax,5 ;Abrir el archivo

mov ebx,file ;Nombre del archivo

mov ecx,0 ;Para solo lectura

int 80h

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

mov eax,3 ;Leer del archivo

mov ebx,[descriptor] ;Tomo el handler

mov ecx,buffer ;Coloco el archivo en el buffer

mov edx,len ;Leo la cantidad de bytes

int 80h

mov eax,4 ;Escribir en pantalla el buffer leído

mov ebx,1

mov ecx,buffer

mov edx,len

int 80h

call SaltoLinea

menu:

mov eax,4     ;Escribir en pantalla mensaje de menú

mov ebx,1

mov ecx,mensaje

mov edx,mensaje\_size

int 80h

mov eax,3     ;Leer de pantalla

mov ebx,0

mov ecx,seleccion

mov edx,1

int 80h

mov cl,52h ;Tecla R a un registro

mov al,45h ;Tecla E a un registro

mov edx,seleccion ;La tecla seleccionada a un registro

mov dl,[edx] ;Acceder al registro

cmp dl,cl ;La tecla es igual que R

je MensajeR ;Función de reemplazar

cmp dl,al ;La tecla es igual que E

jne menu ;Si no, repite el menú

Salir:

call SaltoLinea

mov eax,6 ;Cierra el archivo

mov ebx,[descriptor] ;El handler del archivo

int 80h

mov eax,1 ;Cierra el programa

mov ebx,0

int 80h

section .text

length:

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

Guardar:

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 nuevo buffer en el archivo

mov ecx, newBuffer

mov edx, tam

int 0x80

call Salir

SaltoLinea:

mov eax,4     ;Imprimir en pantalla un salto de linea

mov ebx,1

mov ecx,mensajesalto

mov edx,mensajesalto\_size

int 80h

ret

MensajeR:

mov eax,3     ;Pausa para continuar

mov ebx,0

mov ecx,palabra

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 palabra a buscar

mov ebx,0

mov ecx,palabra

mov edx,10

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 palabra a reemplazar

mov ebx,0

mov ecx,reemplazo

mov edx,10

int 80h

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

call length ;Función ara calcular la longitud

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

dec word [longP] ;Actualizo el valor

xor ebx,ebx ;Limpio las variables

mov ebx,reemplazo ;Calcular el length de la palabra, paso la palabra 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,buffer ;Calcular el length de la palabra, paso la palabra por parametro

call length ;Función ara calcular la longitud

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

dec word [len] ;Actualizo el valor

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

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

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

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

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

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 palabra

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

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

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

je cont ;Si es asi reemplazo la palabra

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

jz aumento ;Si es asi 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 Guardar ;Termino

aumento:

inc eax ;Aumento el buffer viejo

inc ebx ;Aumento la palabra

inc ecx ;Aumento el buffer nuevo

inc dx ;Aumento el puntero del buffer viejo

inc cx ;Aumento el puntero de la palabra

inc cl ;Aumento el puntero del buffer nuevo

jmp ciclo ;Vuelvo al loop principal

restart:

mov ebx, palabra ;Reinicio el registro de la palabra

xor cx, cx ;Reinicio el puntero de la palabra

call ciclo ;Vuelvo al loop principal

cont:

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

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

xor di,di ;Inicializa el puntero del reemplazo

sigue:

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

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

je ciclo ;Vuelvo al loop principal

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

inc di ;Incremento el puntero del reemplazo

inc cl ;Incremento el puntero del buffer nuevo

call sigue ;Repito