**PONTIFICIA UNIVERSIDAD CATOLICA MADRE Y MAESTRA**

****

**Nombre:**

Félix Alejandro Guzmán           2014 - 0565

**Materia:**

Arquitectura Computacional

**Profesor:**

Alvaro A. Reyes P.

**Practica sobre:**

Práctica #3

**Fecha de Entrega:**

viernes, 3 de noviembre de 2017

# Código

.model small

.stack 256

.data

;========================Variables declaradas aquí?===========================

RUTA DB "tarea3.asm",0                  ;ruta de archivo a leer

handler dw 0                            ;manejador del archivo

BufLeido db 2000 dup ('$')              ;buffer del archivo

leido dw 0

Error db "No se pudo leer el archivo",0AH,"$"

;============================================================================

.code

main:

mov ax,@data

mov ds,ax

mov es, ax ;set segment register

and sp, not 3 ;align stack to avoid AC fault

;====================================C?digo==================================

call LimpiarPantalla

call OPEN

call read

call mostrar

mov ah,02 ;para imprimir en pantalla un carácter.

mov dl, 0ah ;cambio de línea.

call Fin

;============================================================================

Fin:

.exit

;================================Funciones aquí==============================

OPEN:

push ax dx                              ;guardar registros

MOV AH,3DH                              ;abrir archivo

mov al,0                                ;asigno 0 para solo leer el archivo

lea dx, RUTA                            ;ruta del archivo

int 21h                                 ;llamar al sistema operativo

mov handler,ax                          ;recibir el manejador de archivo

jc Mensaje

pop dx ax                               ;saco los registros guardados

ret

read:

push ax bx cx dx                        ;guardar registros

mov ah, 3fH                             ;empiezo a leer el archivo

mov bx, handler                         ;recibo el manejador de archivo

lea dx, BufLeido                        ;buffer del archivo

mov cx, 2000                            ;tamaño de bytes a leer

mov leido, ax                           ;tomo el manejador de archivo

int 21h                                 ;llamada al sistema operativo

mov handler, bx

pop dx cx bx ax                         ;saco los registros guardados

ret

mostrar:

push ax bx cx dx es                     ;guardar registros

mov di,0                               ;índice para mostrar archivo en la pantalla

mov bx,0                                ;índice para el buffer

mov cx,0                                ;índice de línea en pantalla

mov ax,0b800h                           ;dirección de la memoria de video

mov es,ax                               ;asigno la dirección al segmento de registro

again:

mov al,BufLeido[bx]                     ;muevo el carácter del buffer al registro

cmp al, 13                              ;reviso si el carácter es un enter

jne notenter                            ;si no es enter sigue

call cambiolinea                        ;si es enter calcula la nueva línea

notenter:

mov byte ptr es:[di],al                 ;imprimo el carácter en pantalla

add di,2                                ;sumo 2 para el siguiente carácter sin alterarle nada

inc bx                                  ;aumento el buffer en 1

add cx,2                                ;aumento el índice de pantalla en 2

cmp bx,leido                            ;reviso si llegue al final del buffer

jne again                               ;repito hasta que termine de leer del buffer

pop di es dx cx bx ax                   ;saco los registros guardados

ret

cambiolinea:

add di,160                              ;sumo 160 para cambiar de línea

sub di,cx                               ;le resto a la siguiente línea el índice de pantalla

add bx,2                                ;aumento el índice del buffer para quitar el enter

mov al, BufLeido[bx]                    ;muevo el siguiente carácter al registro

mov cx,0                                ;reinicio el índice de pantalla

cmp al, 13                              ;reviso si el carácter es un enter

je cambiolinea                          ;si es enter, calcula la nueva línea

ret

LimpiarPantalla proc

    mov ax,03H                          ;limpiar pantalla

    int 10h                             ;llamada al sistema operativo

    ret

endp

Mensaje:

    mov ah,9                            ;para imprimir caracteres en pantalla

    mov dx,OFFSET Error                 ;muestro el mensaje en pantalla

    int 21h                             ;llamo al sistema operativo

jmp Fin

;============================================================================

end main