****

**Pontificia Universidad Católica Madre Y Maestra**

**Facultad De Ingeniería**

**Departamento De Ingeniería De Sistemas Y Computación**

Práctica 3: Lectura de Archivos

**Nombre**

Dhamar Martínez J.           2014-0047

**Materia**

Arquitectura Computacional

**Profesor**

Álvaro A. Reyes P.

**Fecha de Entrega**

viernes, 3 de noviembre de 2017

# Código

.model small

.stack 256

.data

;========================Variables declaradas aqu?===========================

ruta db "DhamarT3.asm",0 ;Guardar la ruta de acceso del archivo.

handler dw 0 ;Para guardar el file handle si no hay error.

Texto db 2000 dup ('$') ;Buffer donde se guarda el archivo.

leido dw 0 ;Para comprobar que se llego al final del archivo.

Msj db "ERROR, FAIL",0AH,"$" ;Declaro un mensaje de error.

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

.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 ;Función para limpiar la pantalla.

call open ;Invoca función para abrir el archivo.

call read ;Invoca función para leer e archivo.

call mostrar ;Función para mostrar el buffer en pantalla.

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

salir:

.exit

;================================Funciones aqu?==============================

    open:

    push ax dx ;Inicializo los registros.

    MOV AH,3DH   ;Inicio funcion de abrir.

    mov al,0             ;Asigna la opcion de solo lectura para abri el archivo.

    lea dx, ruta ;Recibe a ruta a leer.

    int 21h ;Corre el sistema operativo.

    mov handler,ax ;Recibe el handler si no hay errores.

jc error

    pop dx ax ;Limpia los registros

    ret

    read:

    push ax bx cx dx ;Inicializo los registros.

    mov ah, 3fH ;Inicio función de leer.

    mov bx, handler ;Recibe el handler para leer.

    lea dx, Texto ;Recibe el buffer donde se copia la información.

    mov cx, 2000 ;Número de bytes que se van a leer.

    mov leido, ax ;Tomo el handler si no hy error.

    int 21h ;Corre el sistema operativo.

    mov handler, bx ;paso el nuevo handler a mi registro handler.

    pop dx cx bx ax ;Limpia los registros

    ret

    mostrar:

    push ax bx cx dx es ;Inicializo los registros.

    mov di,0 ;Posición para mostrar información en la pantalla.

    mov bx,0      ;Posición para el buffer.

    mov cx,0 ;Posición en las líneas de pantalla.

    mov ax,0b800h ;Asigno la dirección de la tarjeta de VGA para imprimir el texto.

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

    again:

    mov al,Texto[bx] ;Asigno al registro el valor del buffer en la posición indicada.

    cmp al, 13 ;¿Es un enter?

    jne notenter ;Si no continua con el resto del texto.

    call cambiolinea ;Si lo es ejecuta la función cambio de línea.

    notenter:

    mov byte ptr es:[di],al ;Imprime el valor en pantalla.

    add di,2 ;Agrego 2 posiciones al índice para apuntar al siguiente valor.

    inc bx ;Apunto al siguiente valor del buffer.

    add cx,2 ;Apunto al siguiente valor de la pantalla.

    cmp bx,leido ;¿Llegue al final?

    jne again ;Si no, repite el proceso.

    pop di es dx cx bx ax ;Limpia los registros

    ret

    cambiolinea:

    add di,160 ;Cambio la línea en la pantalla.

    sub di,cx ;Le resto el valor en el que estaba para que aparezca en el inicio.

    add bx,2 ;Incremento la posición del buffer para que se salte el cariedge return.

    mov al, Texto[bx] ;Tomo la posición del buffer.

    mov cx,0 ;Reinicio la posición en la pantalla

    cmp al, 0Dh ;¿Es un enter?

    je cambiolinea ;Si lo es, repite el cambio de línea.

    ret ;Si no, retorna.

    LimpiarPantalla:

        mov ax,03H ;Ascii para fin de texto.

        int 10h ;Llama a la tarjeta de video.

        ret

error:

mov ah,09h ;Para imprimir un carácter.

        mov dx,OFFSET Msj ;Muestra el mensaje.

        int 21h ;Llama al sistema operativo.

    jmp salir ;Salir.

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

end main