

```

1: ;=====
2: ;=====
3: ; PROGRAMA                : pe21.asm
4: ; FUNCION                  : Programa que crea y lee un archivo.
5: ; REALIZADO POR           : Prof. Juan Ju?rez Fuentes
6: ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7: ; FECHA                    : 20240528101832
8: ;=====
9: ; DECLARACION DE CONSTANTES
10: ;=====
11: CR            EQU 13      ; Constante CR (Retorno de carro)
12: LF            EQU 10      ; Constante LF (Salto de l?nea)
13: TECLADO_8    EQU 8
14: IMPRIMIR_9   EQU 9
15: FIN_4C00H    EQU 4C00H
16: DOS_21H      EQU 21H
17: BIOS_10H     EQU 10H
18: TAB          EQU 09H
19: BELL         EQU 07H
20: ;=====
21: ; DECLARACION DEL SEGMENTO DE DATOS
22: ;=====
23: DATOS SEGMENT
24:     RENGLON   DB 2
25:     COLUMNA   DB 18
26: ;-----
27:     MENSAJE   DB 'PROGRAMA QUE CREA Y LEE UN ARCHIVO DE DATOS',CR,LF,'$'
28:     filename  db 'output.txt', 0
29:     filehandle dw ?
30:     msg       db 'Hola, Mundo; PROGRAMA DOS!', 0Dh, 0Ah, '$'
31:     length    equ $ - msg
32:     buffer    db 100h dup(?)
33:
34: DATOS ENDS
35: ;=====
36: ; DECLARACION DEL SEGMENTO DE DATOS
37: ;=====
38: PILA SEGMENT STACK
39:     db 32 DUP('0')
40: PILA ENDS
41: ;=====
42: ; DECLARACION DEL SEGMENTO DE CODIGO
43: ;=====
44: CODIGO SEGMENT
45: ;-----
46:     MAIN PROC FAR ; Inicio procedimiento MAIN
47: ;-----
48:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
49: ;-----
50:     CALL ENCABEZADO
51: ;-----
52:     ; Inicializaci?n del segmento de datos
53:     MOV AX,DATOS
54:     MOV DS,AX
55:
56:     ; Crear o abrir archivo (funci?n 3Ch)
57:     MOV AH, 3Ch      ; Funci?n 3Ch - Crear archivo con atributos
58:     MOV CX, 0        ; Atributos de archivo (0 para archivo normal)
59:     LEA DX, filename ; Direcci?n del nombre del archivo
60:     INT 21h          ; Llamada a la interrupci?n DOS
61:     JC error         ; Saltar si hay error
62:     MOV filehandle, AX ; Guardar el manejador de archivo
63:
64:     ; Escribir en el archivo (funci?n 40h)
65:     MOV AH, 40h      ; Funci?n 40h - Escribir en archivo o dispositivo

```

```

66:         MOV BX, filehandle ; Manejador del archivo
67:         LEA DX, msg         ; Direcci?n del mensaje
68:         MOV CX, length      ; Longitud del mensaje
69:         INT 21h             ; Llamada a la interrupci?n DOS
70:         JC error            ; Saltar si hay error
71:
72:         ; Cerrar el archivo (funci?n 3Eh)
73:         MOV AH, 3Eh         ; Funci?n 3Eh - Cerrar archivo
74:         MOV BX, filehandle ; Manejador del archivo
75:         INT 21h             ; Llamada a la interrupci?n DOS
76:         JC error            ; Saltar si hay error
77:
78:         ; Abrir el archivo para lectura (funci?n 3Dh)
79:         MOV AH, 3Dh         ; Funci?n 3Dh - Abrir archivo
80:         MOV AL, 0           ; Modo de acceso: lectura
81:         LEA DX, filename    ; Direcci?n del nombre del archivo
82:         INT 21h             ; Llamada a la interrupci?n DOS
83:         JC error            ; Saltar si hay error
84:         MOV filehandle, AX ; Guardar el manejador de archivo
85:
86:         ; Leer del archivo (funci?n 3Fh)
87:         MOV AH, 3Fh         ; Funci?n 3Fh - Leer archivo o dispositivo
88:         MOV BX, filehandle ; Manejador del archivo
89:         LEA DX, buffer      ; Buffer donde almacenar los datos le?dos
90:         MOV CX, 100h        ; N?mero de bytes a leer
91:         INT 21h             ; Llamada a la interrupci?n DOS
92:         JC error            ; Saltar si hay error
93:
94:         ; Posiciona el cursor para salida a pantalla.
95:         MOV AH, 02          ; Servicio para Fija posicion del Cursor
96:         MOV BH, 00          ; Numero de pagina o pantalla
97:         MOV DH, 4           ; Renglon
98:         MOV DL, 8           ; Columna
99:         INT BIOS_10H
100:
101:        ; Mostrar el contenido le?do en la pantalla (funci?n 40h)
102:        MOV AH, 40h         ; Funci?n 40h - Escribir en archivo o dispositivo
103:        MOV BX, 1           ; Manejador de salida est?ndar (1 = pantalla)
104:        LEA DX, buffer      ; Direcci?n del buffer
105:        INT 21h             ; Llamada a la interrupci?n DOS
106:        JC error            ; Saltar si hay error
107:
108:        ; Cerrar el archivo (funci?n 3Eh)
109:        MOV AH, 3Eh         ; Funci?n 3Eh - Cerrar archivo
110:        MOV BX, filehandle ; Manejador del archivo
111:        INT 21h             ; Llamada a la interrupci?n DOS
112:        JC error            ; Saltar si hay error
113:
114:        ; Terminar el programa (funci?n 4Ch)
115:        MOV AH, 4Ch
116:        XOR AL, AL
117:        INT 21h
118:
119:    error:
120:        ; Manejo de errores (simplemente terminar el programa)
121:        MOV AH, 4Ch
122:        XOR AL, AL
123:        INT 21h
124:
125:        ;-----
126:        ;CALL FIN
127:        ;-----
128:    MAIN ENDP          ; Fin del procedimiento MAIN
129:        ;-----
130:    IMPRIME_TEXTO PROC NEAR

```

```
131:         MOV AH,IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
132:         INT DOS_21H      ; Llamar a la interrupci?n del DOS
133:         RET
134: IMPRIME_TEXTO ENDP
135: ;-----
136: FIN PROC NEAR
137:     MOV AX,FIN_4C00H ; Fin del programa
138:     INT DOS_21H
139:     RET
140: FIN ENDP
141: ;-----
142: LIMPIA_PANTALLA PROC NEAR
143:     MOV AX,0600H ; Establecer modo texto y CLS
144:     MOV BH,1FH ; Colores de fondo (negro=0) y texto (verde=A).
145:     MOV CX,0000H ; Desde el inicio de pantalla
146:     MOV DX,484FH ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
147:     INT BIOS_10H ; Interrupci?n que llama al BIOS
148:     RET
149: LIMPIA_PANTALLA ENDP
150: ;-----
151: POSICIONA_CURSOR PROC NEAR
152:     MOV AH,02 ; Servicio para Fija posicion del Cursor
153:     MOV BH,00 ; Numero de pagina o pantalla
154:     MOV DH,REGLON
155:     MOV DL,COLUMN
156:     INT BIOS_10H
157:     RET
158: POSICIONA_CURSOR ENDP
159: ;-----
160: ENCABEZADO PROC NEAR
161:     CALL LIMPIA_PANTALLA
162:     ;-----
163:     MOV AX,DATOS ; AX=Direcci?n del segmento de datos
164:     MOV DS,AX ; DS=AX. Indicar el segmento de datos
165:     ;-----
166:     CALL POSICIONA_CURSOR ; Llama a procedimiento par posionar cursor.
167:     ;-----
168:     LEA DX,MENSAJE ; DS:DX = direcci?n de MENSAJE
169:     CALL IMPRIME_TEXTO
170:     ;-----
171:     RET
172: ENCABEZADO ENDP
173: ;=====
174: CODIGO ENDS
175: ;=====
176: END MAIN
177: ;=====
178: ;=====
179: ;=====
```