

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                : pe21.asm
4:  ; FUNCION                  : Programa que crea 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                    : 20240522165800
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 UN ARCHIVO DE DATOS',CR,LF,'$'
28:     filename db 'output.txt', 0
29:     filehandle dw ?
30:     msg      db 'Hola, Mundo!', 0Dh, 0Ah, '$'
31:     length   equ $ - msg
32:
33: DATOS ENDS
34: ;=====
35: ; DECLARACION DEL SEGMENTO DE DATOS
36: ;=====
37: PILA SEGMENT STACK
38:     db 32 DUP('0')
39: PILA ENDS
40: ;=====
41: ; DECLARACION DEL SEGMENTO DE CODIGO
42: ;=====
43: CODIGO SEGMENT
44:     ;-----
45:     MAIN PROC FAR ; Inicio procedimiento MAIN
46:     ;-----
47:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
48:     ;-----
49:     CALL ENCABEZADO
50:     ;-----
51:     MOV AX,DATOS
52:     MOV DS,AX
53:
54:     ; Crear o abrir archivo (funcion 3Ch)
55:     MOV AH, 3Ch      ; Funcion 3Ch - Crear archivo con atributos
56:     MOV CX, 0        ; Atributos de archivo (0 para archivo normal)
57:     LEA DX, filename ; Direccion del nombre del archivo
58:     INT 21h          ; Llamada a la interrupcion DOS
59:     JC error         ; Saltar si hay error
60:     MOV filehandle, AX ; Guardar el manejador de archivo
61:
62:     ; Escribir en el archivo (funcion 40h)
63:     MOV AH, 40h      ; Funcion 40h - Escribir en archivo o dispositivo
64:     MOV BX, filehandle ; Manejador del archivo
65:     LEA DX, msg       ; Direccion del mensaje

```

```

66:         MOV CX, length      ; Longitud del mensaje
67:         INT 21h              ; Llamada a la interrupci?n DOS
68:         JC error             ; Saltar si hay error
69:
70:         ; Cerrar el archivo (funci?n 3Eh)
71:         MOV AH, 3Eh           ; Funci?n 3Eh - Cerrar archivo
72:         MOV BX, filehandle    ; Manejador del archivo
73:         INT 21h              ; Llamada a la interrupci?n DOS
74:         JC error             ; Saltar si hay error
75:
76:         ; Terminar el programa (funci?n 4Ch)
77:         MOV AH, 4Ch
78:         XOR AL, AL
79:         INT 21h
80:
81: error:
82:         ; Manejo de errores (simplemente terminar el programa)
83:         MOV AH, 4Ch
84:         XOR AL, AL
85:         INT 21h
86:
87:         ;-----
88:         ;CALL FIN
89:         ;-----
90: MAIN ENDP          ; Fin del procedimiento MAIN
91:         ;-----
92: IMPRIME_TEXTO PROC NEAR
93:     MOV AH, IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
94:     INT DOS_21H        ; Llamar a la interrupci?n del DOS
95:     RET
96: IMPRIME_TEXTO ENDP
97:         ;-----
98: FIN PROC NEAR
99:     MOV AX, FIN_4C00H   ; Fin del programa
100:    INT DOS_21H
101:    RET
102: FIN ENDP
103:         ;-----
104: LIMPIA_PANTALLA PROC NEAR
105:     MOV AX, 0600H       ; Establecer modo texto y CLS
106:     MOV BH, 1FH         ; Colores de fondo (negro=0) y texto (verde=A).
107:     MOV CX, 0000H       ; Desde el inicio de pantalla
108:     MOV DX, 484FH       ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
109:     INT BIOS_10H        ; Interrupci?n que llama al BIOS
110:     RET
111: LIMPIA_PANTALLA ENDP
112:         ;-----
113: POSICIONA_CURSOR PROC NEAR
114:     MOV AH, 02          ; Servicio para Fija posici?n del Cursor
115:     MOV BH, 00          ; Numero de pagina o pantalla
116:     MOV DH, RENGLON
117:     MOV DL, COLUMNNA
118:     INT BIOS_10H
119:     RET
120: POSICIONA_CURSOR ENDP
121:         ;-----
122: ENCABEZADO PROC NEAR
123:     CALL LIMPIA_PANTALLA
124:     ;-----
125:     MOV AX, DATOS ; AX=Direcci?n del segmento de datos
126:     MOV DS, AX    ; DS=AX. Indicar el segmento de datos
127:     ;-----
128:     CALL POSICIONA_CURSOR ; Llama a procedimiento par posionar cursor.
129:     ;-----
130:     LEA DX, MENSAJE ; DS:DX = direcci?n de MENSAJE

```

```
131:          CALL IMPRIME_TEXTO
132:          ;-----
133:          RET
134:      ENCABEZADO ENDP
135:  ;=====
136: CODIGO ENDS
137:  ;=====
138: END MAIN
139:  ;=====
140:  ;=====
141:  ;=====
```