

Manual Técnico

El programa cuenta con 2 archivos distintos, el primero se llama *pra4.asm* y es el que lleva el código fuente de la práctica, el segundo se llama *macro4.asm* y es el encargado de las macros repetitivas del código fuente del juego.

Pra4.asm:

En el segmento de datos, se encuentran todas las variables y etiquetas utilizadas para la creación correcta del programa, para un mejor control cada una va etiquetada con su finalidad en el programa.

```
7 ;===== SEGMENTO DE DATOS =====
8 .data
9
10
11 ;----- MENSAJES DEL MENU PRINCIPAL -----
12 saludo1 db 0ah, 0dh, ' UNIVERSIDAD DE SAN CARLOS DE GUATEMALA ', '$'
13 saludo2 db 0ah, 0dh, ' ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1 B', '$'
14 saludo3 db 0ah, 0dh, ' Adriana Marie Gomez Davila 201504236', '$'
15 saludo4 db 0ah, 0dh, ' Practica 4', '$'
16 saludo5 db 0ah, 0dh, '>> INGRESE x SI DESEA CERRAR EL PROGRAMA', '$'
17 imprimir db 0ah, 0dh, ' ', '$'
18 err db 0ah, 0dh, '*** ERROR COMANDO NO IDENTIFICADO ***', '$'
19
20
21
22 ;----- MENSAJES Y VARIABLES PARA ABRIR RUTA DE ARCHIVO -----
23 bufferentrada db 500 dup('$')
24 handlerentrada dw ?
25 bufferinformacion db 500 dup('$')
26 ingresa ruta db 0ah, 0dh, 'Ingrese una ruta de archivo', 0ah, 0dh, 'Ejemplo: entrada.txt', '$'
27 enc2 db 0ah, 0dh, '1.) Mostrar informacion', 0ah, 0dh, '2.) Cerrar archivo', '$'
28 op db 0ah, 0dh, 'Elija una opcion:', '$'
29 err1 db 0ah, 0dh, 'Error al abrir el archivo puede que no exista', '$'
30 err2 db 0ah, 0dh, 'Error al cerrar el archivo', '$'
31 err3 db 0ah, 0dh, 'Error al escribir en el archivo', '$'
32 err4 db 0ah, 0dh, 'Error al crear en el archivo', '$'
33 err5 db 0ah, 0dh, 'Error al leer en el archivo', '$'
34 E1 db 0ah, 0dh, '*****
35 E2 db 0ah, 0dh, '***** REPORTE *****
36 E3 db 0ah, 0dh, '-----> CONTADORES:', '$'
37
38
39
40
41 ;----- VARIABLES DEL MODO MEMORIA PARA PINTAR PALABRAS -----
42 fila db 0
43 columna db 0
```

En el segmento de código, se encuentra el flujo de la practica, como principal el despliegue del menú, dependiendo de la opción elegida del menú, saltara hasta esa etiqueta que contendrá el flujo de código que corresponde.

```

90
91 ;===== SEGMENTO DE CODIGO =====
92 .code
93 main proc
94
95     Inicio:
96         xor si, si
97         xor di, di
98         print imprimir
99         print saludo1
100        print saludo2
101        print saludo3
102        print saludo4
103        print imprimir
104        print saludo5
105        print imprimir
106
107        ObtenerTexto arregloComando
108        cmp arregloComando[0], 120 ;si es x
109        je salir
110        cmp arregloComando[0], 99 ;si es c
111        je Ce
112        cmp arregloComando[0], 97 ;si es a
113        je AbrirArchivo
114        cmp arregloComando[0], 112 ;si es p
115        je Pe
116        cmp arregloComando[0], 114 ;si es r
117        je Reporte
118        jmp Error
119
120        Pe:
121        cmp arregloComando[5], 100 ;si es d
122        je CalculaMediaDip
123        cmp arregloComando[5], 104 ;si es h
124        je CalculaMediaHia
125        cmp arregloComando[5], 116 ;si es t
126        je CalculaMediaTrip

```

El método para la creación de reportes, el cual consiste en llamar a la macro y hacer uso de ella, mandándole etiquetas ya definidas en el segmento de datos antes descrito.

```
137  
138 ;----- COMANDO "REPORTE" -----  
139 Reporte:  
140     print imprimir  
141     print ingreseruta  
142     print imprimir  
143     limpiar bufferentrada, SIZEOF bufferentrada, 24h  
144     obtenerRuta bufferentrada  
145     crear bufferentrada, handlerentrada  
146     jmp EscribirArchivo  
147  
148 EscribirArchivo:  
149     print imprimir  
150     limpiar bufferInformacion, SIZEOF bufferInformacion, 24h  
151     escribir handlerentrada, E1, SIZEOF E1  
152     escribir handlerentrada, E2, SIZEOF E2  
153     escribir handlerentrada, E3, SIZEOF E3  
154  
155     escribir handlerentrada, m11, SIZEOF m11  
156     Limpiar Cantidades, SIZEOF Cantidades, 24h  
157     IntToString palabrasTotales, Cantidades  
158     escribir handlerentrada, Cantidades, SIZEOF Cantidades  
159  
160     escribir handlerentrada, t1, SIZEOF t1  
161     Limpiar Cantidades, SIZEOF Cantidades, 24h  
162     IntToString contadorDip, Cantidades  
163     escribir handlerentrada, Cantidades, SIZEOF Cantidades  
164  
165     escribir handlerentrada, t2, SIZEOF t2  
166     Limpiar Cantidades, SIZEOF Cantidades, 24h  
167     IntToString contadorHia, Cantidades  
168     escribir handlerentrada, Cantidades, SIZEOF Cantidades  
169  
170     escribir handlerentrada, t3, SIZEOF t3  
171     Limpiar Cantidades, SIZEOF Cantidades, 24h  
172     IntToString contadorTrip, Cantidades  
173     escribir handlerentrada, Cantidades, SIZEOF Cantidades
```

El método para contar la proporción (media) de cada tipo de palabra, lo realizan 3 métodos distintos, los cuales lo que hacen es una operación “ $(100 * \text{palabrasTotales}) / \text{contadorPalabraIndicada}$ ”.

```
183 ;----- COMANDO "PROP_DIPTONGO|HIATO|TRIPTONGO" -----
184     CalculaMediaDip: ;media = (100 * palabrasTotales) / contadorDip
185         Limpiar Cantidades, SIZEOF Cantidades, 24h
186         IntToString palabrasTotales, Cantidades
187         print m11
188         print Cantidades
189         print imprimirter
190         xor ax, ax
191         xor bx, bx
192         mov al, Cantidades ;al = 77
193         mov bl, 100
194         MUL bl ;al = (100 * Cantidades)
195         mov Cantidades, al
196         print m1
197         print Cantidades
198         print imprimirter
199         jmp Inicio
200
201     CalculaMediaHia:
202         Limpiar Cantidades, SIZEOF Cantidades, 24h
203         IntToString palabrasTotales, Cantidades
204         print m11
205         print Cantidades
206         print imprimirter
207         xor ax, ax
208         xor bx, bx
209         mov al, Cantidades ;al = 77
210         mov bl, 100
211         MUL bl ;al = (100 * Cantidades)
212         mov Cantidades, al
213         print m2
214         print Cantidades
215         print imprimirter
216         jmp Inicio
217
218     CalculaMediaTrip:
219         Limpiar Cantidades, SIZEOF Cantidades, 24h
```

El método para contar el tipo de palabra indicada, lo realizan 3 métodos distintos según sea el caso indicado. Lo que hacen en general, es limpiar una variable tipo dw para poder mandarle el contador del tipo que deseemos y este casteo se guarda en la variable tipo dw para luego imprimirla en pantalla.

```
301
302 ;----- COMANDO "CONTAR_DIPTONGO|HIATO|TRIPTONGO" -----
303
304     comContar:
305         cmp arregloComando[7], 100 ;si es d
306         je contarDip
307         cmp arregloComando[7], 104 ;si es h
308         je contarHia
309         cmp arregloComando[7], 116 ;si es t
310         je contarTrip
311         jmp Error
312
313     contarDip:
314         Limpiar Cantidades, SIZEOF Cantidades, 24h
315         IntToString contadorDip, Cantidades
316         print t1
317         print Cantidades
318         print imprimir
319         jmp Inicio
320
321     contarHia:
322         Limpiar Cantidades, SIZEOF Cantidades, 24h
323         IntToString contadorHia, Cantidades
324         print t2
325         print Cantidades
326         print imprimir
327         jmp Inicio
328
329     contarTrip:
330         Limpiar Cantidades, SIZEOF Cantidades, 24h
331         IntToString contadorTrip, Cantidades
332         print t3
333         print Cantidades
334         print imprimir
335         jmp Inicio
```

Para colorear las palabras del archivo abierto, necesitamos ingresar al modo video, cada línea tiene escrito que hace en el programa. Luego en el ciclo1, hacemos las comparaciones de los tipos de palabra para poder mandarle a las macros el color que deben de pintar según sea el caso.

```

338
339
340 ;----- COMANDO "COLOREAR" -----
341 comColorear:
342     print imprimir
343     ;con esto entramos al modo video
344     mov ah, 0
345     mov al, 03h
346     int 10h
347
348     imprimir bufferInformacion
349     imprimir imprimir
350     imprimir imprimir
351
352     ;con esto le decimos que imprima en la posicion 3 despues
353     mov ah, 03h
354     mov bh, 00h
355     int 10h ;dh guarda el valor de la ult.pos. de fila y dl guarda el valor de la ult.pos. de columna
356
357     mov fila, dh
358     mov columna, dl
359     mov si, 0
360     mov di, 0
361
362
363
364 ;----- POSICIONAR EL CURSOR EN LA MATRIZ -----
365 ciclo1:
366     posicionarCursor fila, columna
367
368
369
370     ;hacemos un if, para verificar si en la 3era posicion es una vocal, es un triptongo
371     cmp bufferInformacion[si + 2], 105 ;letra i
372     je esTrip

```

```

435 esDip:
436     ;----> para diptongos
437     esDiptongo bufferInformacion[si], bufferInformacion[si + 1]
438     cmp al, 0
439     je letra
440     ;pintamos el diptongo
441     imprimirVideo bufferInformacion[si], 0010b ;verde
442     inc columna ;aumenta la posicion del cursor
443     inc si
444     posicionarCursor fila, columna
445     imprimirVideo bufferInformacion[si], 0010b ;verde
446     jmp siguiente
447
448 esHia:
449     ;----> para hiatos
450     esHiato bufferInformacion[si], bufferInformacion[si + 1]
451     cmp auxHiato, 0
452     je letra
453     ;pintamos el hiato
454     imprimirVideo bufferInformacion[si], 0100b ;rojo
455     inc columna ;aumenta la posicion del cursor
456     inc si
457     posicionarCursor fila, columna
458     imprimirVideo bufferInformacion[si], 0100b ;rojo
459     jmp siguiente
460
461 esTrip:
462     ;----> para triptongos
463     esTriptongo bufferInformacion[si], bufferInformacion[si + 1], bufferInformacion[si + 2]
464     cmp auxTriptongo, 0
465     je letra
466     ;pintamos el triptongo
467     imprimirVideo bufferInformacion[si], 1110b ;amarillo
468     inc columna ;aumenta la posicion del cursor
469     inc si
470
471     posicionarCursor fila, columna

```

Macro4.asm:

Este archivo es muy importante, ya que sin el, no funcionaria el programa.

Se divide en varias macros, todas son importantes.

Las primeras tres macros son las encargadas de recibir cadenas, imprimir cadenas y para limpiar pantalla.

```
1 ;----- MACRO PARA IMPRIMIR -----
2
3 print macro buffer
4     mov ax, @data
5     mov ds, ax
6     mov ah, 09h
7     mov dx, offset buffer
8     int 21h
9
10    endm
11
12
13 ;----- MACRO PARA LEER CARACTER OBTENIDO -----
14
15 getChar macro
16     mov ah, 01h
17     int 21h
18     ;al registro AL se le manda el codigo ascii recibido, en codigo hexadecimal
19
20    endm
21
22
23 ;----- MACRO PARA LIMPIAR PANTALLA -----
24
25 LimpiarConsola macro
26     mov ax, 0600h
27     mov bh, 89h
28     mov cx, 0000h
29     mov dx, 184Fh
30     int 10h
31
32    endm
```

Las 3 macros que le siguen, son necesarias para imprimir en pantalla el modo video.

```
54
55 ;----- MACRO PARA ESCRIBIR UN VALOR DIRECTO -----
56 escribirChar macro char
57     mov ah, 02h
58     mov dl, char
59     int 21h
60 endm
61
62
63
64 ;----- MACRO PARA POSICIONAR CURSOR -----
65 posicionarCursor macro x, y
66     mov ah, 02h
67     mov dh, x
68     mov dl, y
69     mov bh, 0
70     int 10h
71 endm
72
73
74
75 ;----- MACRO PARA IMPRIMIR MODO VIDEO -----
76 imprimirVideo macro caracter, color
77     mov ah, 09h
78     mov al, caracter ;al guarda el valor que vamos a escribir
79     mov bh, 0 ;posicion actual es la 0
80     mov bl, color ;pinta rojo (b de binario)
81
82     mov cx, 1
83     int 10h
84 endm
85
```

Las macros que detectan el tipo de palabra que está leyendo, son esDiptongo, esHiato, esTriptongo. En estas macros se verifica que las palabras cumplan con las reglas establecidas de la practica para identificar de que tipo son.

```
88
89 ;----- MACRO PARA DIPTONGOS -----
90 esDiptongo macro caracter1, caracter2
91 LOCAL salida, esA, esE, esO, Diptongo, esII, esU
92 ;en el registro al va a tener un 1 si es diptongo o un 0 si no lo es
93     mov al, 0
94
95     esA:
96         cmp caracter1, 97 ;letra a
97         jne esE
98
99         cmp caracter2, 105 ;letra i
100        je Diptongo
101        cmp caracter2, 117 ;letra u
102        je Diptongo
103        jmp salida
104
105     esE:
106         cmp caracter1, 101 ;letra e
107         jne esO
108
109         cmp caracter2, 105 ;letra i
110        je Diptongo
111        cmp caracter2, 117 ;letra u
112        je Diptongo
113        jmp salida
114
115     esO:
116         cmp caracter1, 111 ;letra o
117         jne esII
118
119         cmp caracter2, 105 ;letra i
120        je Diptongo
121        cmp caracter2, 117 ;letra u
122        je Diptongo
123        jmp salida
```