





Содержимое LST-файла:

Turbo Assembler Version 1.01 09/30/21 21:22:59 Page 1

LAB2.ASM

1 0000 CODESG SEGMENT PARA; CODESG - название сегмента. SEGMENT - обозначение начала сегмента

2 ASSUME CS:CODESG, DS:CODESG, SS:CODESG, ES:CODESG; ASSUME позволяет использовать сегменты как

3 сегменты кода, данных или стека. CS:CODESG - сегмент кода CODESG.

4 ;DS - Сегмент данных. SS - Сегмент стека. ES - дополнительный сегмент, и может не использоваться

5 программой

6 ORG 100H ; Директива org 100H говорит компилятору что всю адресацию внутри кода нужно сместить

7 именно на эти 100 байт(нужно для того, чтобы можно было указывать реальные адреса)

8 0100 E9 008B BEGIN: JMP MAIN ; JMP MAIN выполняет безусловный переход в MAIN

9

10 0103 ?? A101 DB ?

11 0104 4C 41 4E 44 20 56 45 + A102 DB 'LAND VENTURE'

12 4E 54 55 52 45

13 0110 73 A103 DB 115

14 0111 50 A104 DB 50H

15 0112 84 A105 DB 10000100B

16 0113 03 4B 4A 48 09 48 4B + A106 DB 03,'KJH',09,'HKL',07,'SQW'

17 4C 07 53 51 57

18 011F 34 35 36 37 35 33 A107 DB '456753'

19 0125 09\*(00) A108 DB 9 DUP(0)

20 012E CDF3 B101 DW 0CDF3H

21 0130 00A9 B102 DW 10101001B

22 0132 0110r B103 DW A103

23 0134 0002 0005 0007 0009 + B104 DW 2,5,7,9,11

24 000B

25 013E 06\*(0000) B105 DW 6 DUP(0)

26 014A ???????? C101 DD ?

27 014E 00004954 C102 DD 'IT'

28 0152 00007DF0 C103 DD 32240

29 0156 0000000C C104 DD A103-A102

30 015A 0000000E 0000002B C105 DD 14,43

31 0162 ???????????????? D101 DQ ?

32 016A 0000000000005D35 D102 DQ 05D35H

33 0172 0000000000007DF0 D103 DQ 32240

34 017A ???????????????????? E101 DT ?

35 0184 0000000000000000554E E102 DT 'UN'

36

37 018E MAIN PROC NEAR

38

39 018E B4 09 mov ah,09h

40 0190 BA 0104r mov dx,offset A102

41 0193 CD 21 int 21h

42

43 0195 C3 RET; команда RET выполняет выход из программы или процедуры

44

45 0196 MAIN ENDP

46 0196 CODESG ENDS; CODESG ends - конец сегмента CODESG

47 END BEGIN

Turbo Assembler Version 1.01 09/30/21 21:22:59 Page 2

Symbol Table

Symbol Name Type Value

??DATE Text "09/30/21"

??FILENAME Text "LAB2 "

??TIME Text "21:22:59"

??VERSION Number 0101

@CPU Text 0101H

@CURSEG Text CODESG

@FILENAME Text LAB2

@WORDSIZE Text 2

A101 Byte CODESG:0103

A102 Byte CODESG:0104

A103 Byte CODESG:0110

A104 Byte CODESG:0111

A105 Byte CODESG:0112

A106 Byte CODESG:0113

A107 Byte CODESG:011F

A108 Byte CODESG:0125

B101 Word CODESG:012E

B102 Word CODESG:0130

B103 Word CODESG:0132

B104 Word CODESG:0134

B105 Word CODESG:013E

BEGIN Near CODESG:0100

C101 Dword CODESG:014A

C102 Dword CODESG:014E

C103 Dword CODESG:0152

C104 Dword CODESG:0156

C105 Dword CODESG:015A

D101 Qword CODESG:0162

D102 Qword CODESG:016A

D103 Qword CODESG:0172

E101 Tbyte CODESG:017A

E102 Tbyte CODESG:0184

MAIN Near CODESG:018E

Groups & Segments Bit Size Align Combine Class

CODESG 16 0196 Para none

**Содержимое ASM-файла(с комментариями):**

CODESG SEGMENT PARA; CODESG - название сегмента. SEGMENT - обозначение начала сегмента

ASSUME CS:CODESG, DS:CODESG, SS:CODESG, ES:CODESG; ASSUME позволяет использовать сегменты как сегменты кода, данных или стека. CS:CODESG - сегмент кода CODESG.

;DS - Сегмент данных. SS - Сегмент стека. ES - дополнительный сегмент, и может не использоваться программой

ORG 100H ; Директива org 100H говорит компилятору что всю адресацию внутри кода нужно сместить именно на эти 100 байт(нужно для того, чтобы можно было указывать реальные адреса)

BEGIN: JMP MAIN ; JMP MAIN выполняет безусловный переход в MAIN

A101 DB ?

A102 DB 'LAND VENTURE'

A103 DB 115

A104 DB 50H

A105 DB 10000100B

A106 DB 03,'KJH',09,'HKL',07,'SQW'

A107 DB '456753'

A108 DB 9 DUP(0)

B101 DW 0CDF3H

B102 DW 10101001B

B103 DW A103

B104 DW 2,5,7,9,11

B105 DW 6 DUP(0)

C101 DD ?

C102 DD 'IT'

C103 DD 32240

C104 DD A103-A102

C105 DD 14,43

D101 DQ ?

D102 DQ 05D35H

D103 DQ 32240

E101 DT ?

E102 DT 'UN'

MAIN PROC NEAR

mov ah,09h

mov dx,offset A102

int 21h

RET; команда RET выполняет выход из программы или процедуры

MAIN ENDP

CODESG ENDS; CODESG ends - конец сегмента CODESG

END BEGIN

Работа программы:

