Для начала я ознакомился с материалом лекции 23 с названием «Как бороться с управляющими».

Рассмотрел код в конце материала:

.MODEL SMALL

.Data

CONST16 DB 16

HEXSYM DB ‘0123456789ABCDEF’

PATTERN DB ‘XX-$‘

CODSYM DB ?

.CODE

.STARTUP

MOV AX, 40h

MOV ES, AX

MOV СODSYM, 0.

MOV CX, 16

PRTABLE:

PUSH CX

MOV CX, 16

POVT :

MOV AL, CODSYM

MOV AH, 0

DIV CONST16

MOV BX, Offset HexSym

XLAT

MOV PATTERN, AL

MOV AL, AH

XLAT

MOV PATTERN + 1

MOV ah, 9

LEA DX, PATTERN

INT 21h

MOV BH, ES [ 62h ]

MOV BL, 0

MOV AH, 10

MOV AL, CODSYM

MOV CX, 1

INT 21h

MOV AH, 03

INT 21h

MOV AH, 02

ADD DL, 2

INT 21h

ADD CODSYM, 16

LOOP POVT

POP CX

INC CODSYM

LOOP PRTABLE

MOV AH, 4ch

INT 21h

END

И сразу исправил несколько моментов, а именно

MOV AL, 40h изменил на MOV AX, 40h: инструкция изменена для перемещения значения 40h в регистр AX вместо AL. Это делается для правильной установки сегментного регистра ES в следующей строке.

добавил MOV ES, AX: в этой строке для регистра ES (дополнительный сегмент) устанавливается значение AX, равное 40h. ES используется для доступа к видеопамяти для отображения символов.

INT 21h изменил на INT 10h: номер прерывания изменяется с 21h на 10h. INT 10h — видеопрерывание BIOS, используемое для отображения символов на экране.

INT 21h изменил на INT 10h: номер прерывания изменен с 21h на 10h для обеспечения единообразия.

INT 21h изменил на INT 10h: номер прерывания изменен с 21h на 10h для обеспечения единообразия.

И получил следующий код:

.Model Small

.Data

CONST16 DB 16

HEXSYM DB '0123456789ABCDEF'

PATTERN DB 'XX-$'

CODSYM DB ?

ENDL DB 0ah,0dh,'$'

.CODE

.STARTUP

MOV AL, 40h

MOV ES, AX

MOV CODSYM, 0

MOV CX, 16

PRTABLE:

PUSH CX

MOV CX, 16

POVT:

MOV AL, CODSYM

MOV AH, 0

DIV CONST16

MOV BX, Offset HEXSYM

XLAT

MOV PATTERN, AL

MOV AL, AH

XLAT

MOV PATTERN + 1, AL

MOV AH, 9

LEA DX, PATTERN

INT 21h

MOV BH, ES: [ 62h ]

MOV BL, 0

MOV AH, 10

MOV AL, CODSYM

PUSH CX

MOV CX, 1

INT 10h

MOV AH, 03

INT 10h

MOV AH, 02

ADD DL, 2

INT 10h

ADD CODSYM, 16

POP CX

LOOP POVT

MOV AH, 9

LEA DX, ENDL

INT 21h

POP CX

INC CODSYM

LOOP PRTABLE

MOV AH, 4ch

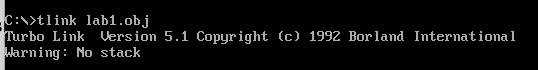
INT 21h

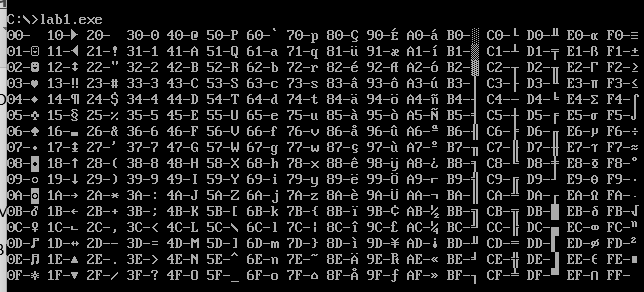
END

Полученный исходный текст сохранил в файле lab1.asm

Выполнив следующие команды:







Таким образом, вывел на экран кодировочную таблицу в таком виде, чтобы была возможность увидеть на экране и управляющие символы.

После я решил просто пропустить вывод для всех символов, кроме символов c кодами 10 и 13 кодировочной таблицы ASCII.

Для этого я сделал следующие изменения:

добавил MOV CODSYM, 10: Эта строка устанавливает начальное значение CODSYM равным 10, что соответствует символу новой строки. Начиная с этого значения, ненужные итерации в цикле будут пропущены.

добавил MOV CX, 2: эта строка устанавливает регистр CX в значение 2, указывая, что мы хотим выполнить итерацию только два раза. Это гарантирует, что будут напечатаны только символы 10 и 13.

ADD CODSYM, 3 , значение приращения изменено на 3 вместо 16. Увеличивая CODSYM на 3, мы пропускаем ненужные символы в цикле.

LOOP PRTABLE изменено: цикл выполняется только два раза, как указано значением в CX. Это предотвращает ненужные итерации.

Получил следующий исходный текст:

.Model Small

.Data

CONST16 DB 16

HEXSYM DB '0123456789ABCDEF'

PATTERN DB 'XX-$'

CODSYM DB ?

ENDL DB 0ah,0dh,'$'

.CODE

.STARTUP

MOV AL, 40h

MOV ES, AX

MOV CODSYM, 10 ; Start with character 10 (newline)

MOV CX, 2 ; Only two iterations

PRTABLE:

PUSH CX

MOV AL, CODSYM

MOV AH, 0

DIV CONST16

MOV BX, Offset HEXSYM

XLAT

MOV PATTERN, AL

MOV AL, AH

XLAT

MOV PATTERN + 1, AL

MOV AH, 9

LEA DX, PATTERN

INT 21h

MOV BH, ES: [ 62h ]

MOV BL, 0

MOV AH, 10

MOV AL, CODSYM

PUSH CX

MOV CX, 1

INT 10h

MOV AH, 03

INT 10h

MOV AH, 02

ADD DL, 2

INT 10h

ADD CODSYM, 3 ; Increment CODSYM to skip the unnecessary iterations

POP CX

LOOP PRTABLE

MOV AH, 9

LEA DX, ENDL

INT 21h

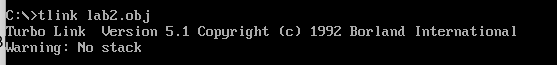
MOV AH, 4ch

INT 21h

END

Сохранил текст в файле lab2.asm, выполнил необходимые команды







И получил желаемый результат.