**Контрольная работа**

Доскоч Роман 4 курс 13 группа



Условие задачи 1.04

Получить программу, которая выводит на экран размер соответствующего ей исполнимого модуля.

Решение:

Исходный текст программы начинается с определения двух переменных данных: PROGRAM\_NAME и BYTES. PROGRAM\_NAME определяется как массив из 256 байт, все они инициализированы нулем, в то время как BYTES определяется как строка символов с надписью "байты".

.MODEL SMALL

.DATA

PROGRAM\_NAME DB 256 DUP (0)

BYTES DB " bytes$"

Затем код вводится в начальную метку, которая отмечает начало основной программы. Он настраивает сегмент ДАННЫХ и копирует значение регистра ES в регистр AX. Регистр ES — это сегментный регистр, который используется для хранения начального адреса сегмента данных.

.CODE

START:

MOV AX, @DATA

MOV DS, AX

Далее код выполняет поиск определенной последовательности байтов в сегменте ES. Он делает это путем увеличения регистра SI до тех пор, пока не найдет байт со значением 0, за которым следует байт со значением 1. Как только эти байты найдены, код копирует имя программы из сегмента ES в переменную PROGRAM\_NAME.

MOV AX, ES:[2CH]

MOV ES, AX

MOV SI, -1

SEARCH\_01:

ADD SI, 1

MOV AL, ES:[SI]

CMP AL, 0

JNE SEARCH\_01

MOV AL, ES:[SI+1]

CMP AL, 1

JNE SEARCH\_01

ADD SI, 2

MOV BX, OFFSET PROGRAM\_NAME

После копирования названия программы код выводит его на экран, используя прерывание INT 21H. Затем он использует другое прерывание, чтобы получить размер программы, и сохраняет его в регистре AX.

COPY\_NAME:

ADD SI, 1

MOV AL, ES:[SI]

MOV [BX], AL

INC BX

CMP AL, 0

JNE COPY \_NAME

MOV DX,OFFSET PROGRAM\_NAME

MOV AX,3D00H

INT 21H

Затем размер исполнимого в текст и выводит его на экран, за которым следует строка "байты".

MOV BX, AX

MOV AL, 2

XOR CX, CX

XOR DX, DX

MOV AH, 42H

INT 21H

PUSH BX

MOV CX, 0

NUM\_TO\_TEXT:

CMP AX, 0

JE PRINT

XOR DX, DX

MOV BX, 10

DIV BX

PUSH DX

INC CX

JMP NUM\_TO\_TEXT

PRINT:

POP DX

ADD DX, 30H

MOV AH, 02H

INT 21H

LOOP PRINT

POP BX

MOV AH, 3EH

INT 21H

MOV DX, OFFSET BYTES

MOV AH, 9H

INT 21H

MOV AH, 4CH

INT 21H

END START

Результат работы:

