Лабораторная работа №3 часть 2

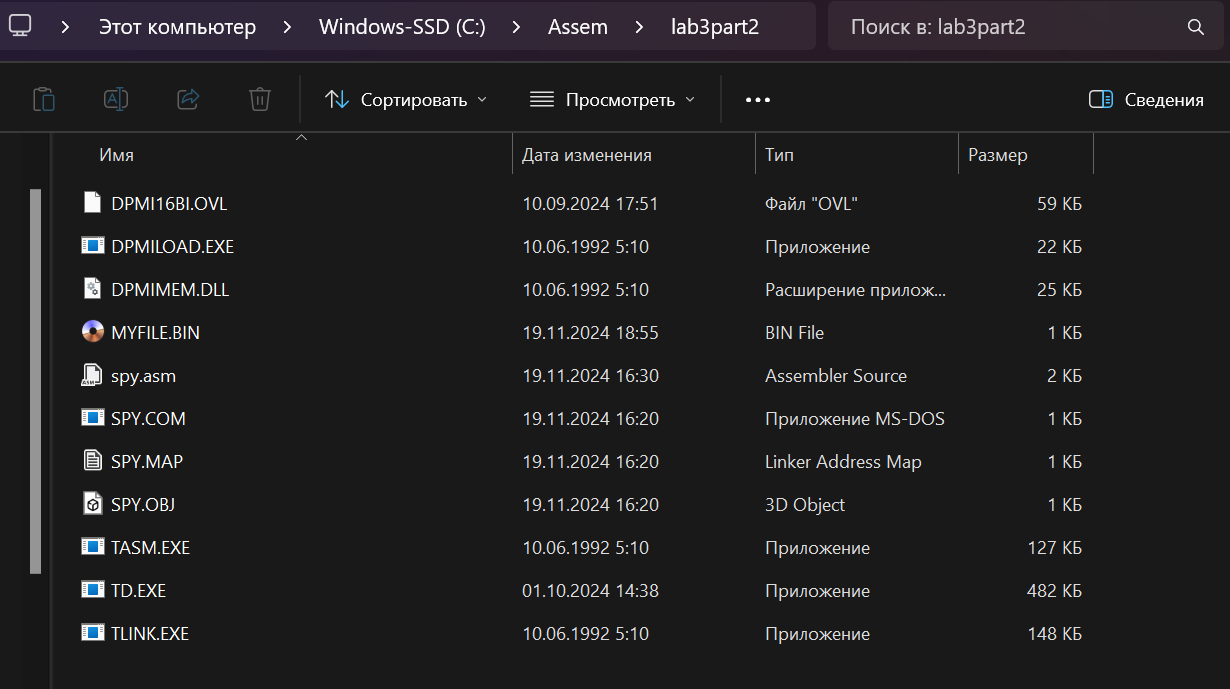
ноипШ

**Задание**

**ЛАБОРАТОРНАЯ РАБОТА №3. Часть 2. YPS!!! Получить программу, которая по информации, СОДЕРЖАЩЕЙСЯ в файле из части 1 лаб работы№3, выводит на экран СООТВЕТСТВУЮЩУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НАБРАННЫХ НА КЛАВИАТУРЕ. Решение может быть только в рамках DosBox и языка записи алгоритмов assembler/**

**Ход решения**

Для начала я перенесла все файлы из первой части лабораторной в новую папку. Для своего же удобства.



Попробуем поработать с кодом, который был нам предоставлен:

.MODEL TINY

.CODE

ORG 100H

START:

MOV DX, OFFSET FName

CALL OPEN\_FILE

CALL GET\_FILE\_LENGTH

MOV FLen, AX

PUSH BX

FIND\_ALL\_SYMBOLS:

mov ax, 4200h

POP BX

PUSH BX

xor cx, cx

mov dx, Count

int 21h

inc Count

MOV AX, Count

CMP AX, FLen

JE END\_FILE

mov ah, 3Fh

mov cx, 1

mov dx, offset Symbol

int 21h

CMP AX, 0

JE END\_FILE

MOV AL, 2Ah

CMP AL, Symbol

JE SHIFT\_START

MOV AL, 170

CMP AL, Symbol

JE SHIFT\_END

MOV CX, SymbolsLen

FIND\_SYMBOL:

MOV AX, CX

SUB AL, 1

mov bx, offset Codes

xlat

CMP AL, Symbol

JE SYMBOL\_FOUND

SUB CX, 1

CMP CX, 65535

JE SYMBOL\_NOT\_FOUND

JMP FIND\_SYMBOL

SYMBOL\_FOUND:

MOV AX, 0

CMP AX, Shift

JE WITHOUT\_SHIFT

mov bx, offset ShiftSymbols

JMP CONT

WITHOUT\_SHIFT:

mov bx, offset Symbols

CONT:

MOV AX, CX

SUB AX, 1

xlat

MOV AH, 02H

MOV DX, AX

INT 21H

JMP FIND\_ALL\_SYMBOLS

SHIFT\_START:

MOV Shift, 1

JMP FIND\_ALL\_SYMBOLS

SHIFT\_END:

MOV Shift, 0

JMP FIND\_ALL\_SYMBOLS

SYMBOL\_NOT\_FOUND:

JMP FIND\_ALL\_SYMBOLS

END\_FILE:

POP BX

CALL CLOSE\_FILE

MOV AX, 4C00H

INT 21H

FName DB 'myfile.bin', 0

Symbol DB 1 dup (?)

Codes DB 29h, 02h, 03h, 04h, 05h, 06h, 07h, 08h

db 09h, 0Ah, 0Bh, 0Ch, 0Dh, 2Bh, 10h, 11h

db 12h, 13h, 14h, 15h, 16h, 17h, 18h, 19h

db 1Ah, 1Bh, 1Eh, 1Fh, 20h, 21h, 22h, 23h

db 24h, 25h, 26h, 27h, 28h, 2Ch, 2Dh, 2Eh

db 2Fh, 30h, 31h, 32h, 33h, 34h, 35h, 39h

Symbols DB "`1234567890-=\qwertyuiop[]asdfghjkl;'zxcvbnm,./ "

SymbolsLen = $ - Symbols

ShiftSymbols db '~!@#$%^&\*()\_+|QWERTYUIOP{}ASDFGHJKL:"ZXCVBNM<>? '

Count DW 0

FLen DW 0

Shift DW 0

OPEN\_FILE PROC NEAR

MOV AX, 3D00H

INT 21H

MOV BX, AX

RET

OPEN\_FILE ENDP

CLOSE\_FILE PROC NEAR

MOV AH, 3EH

INT 21H

RET

CLOSE\_FILE ENDP

GET\_FILE\_LENGTH PROC NEAR

MOV AX, 4202H

XOR CX, CX

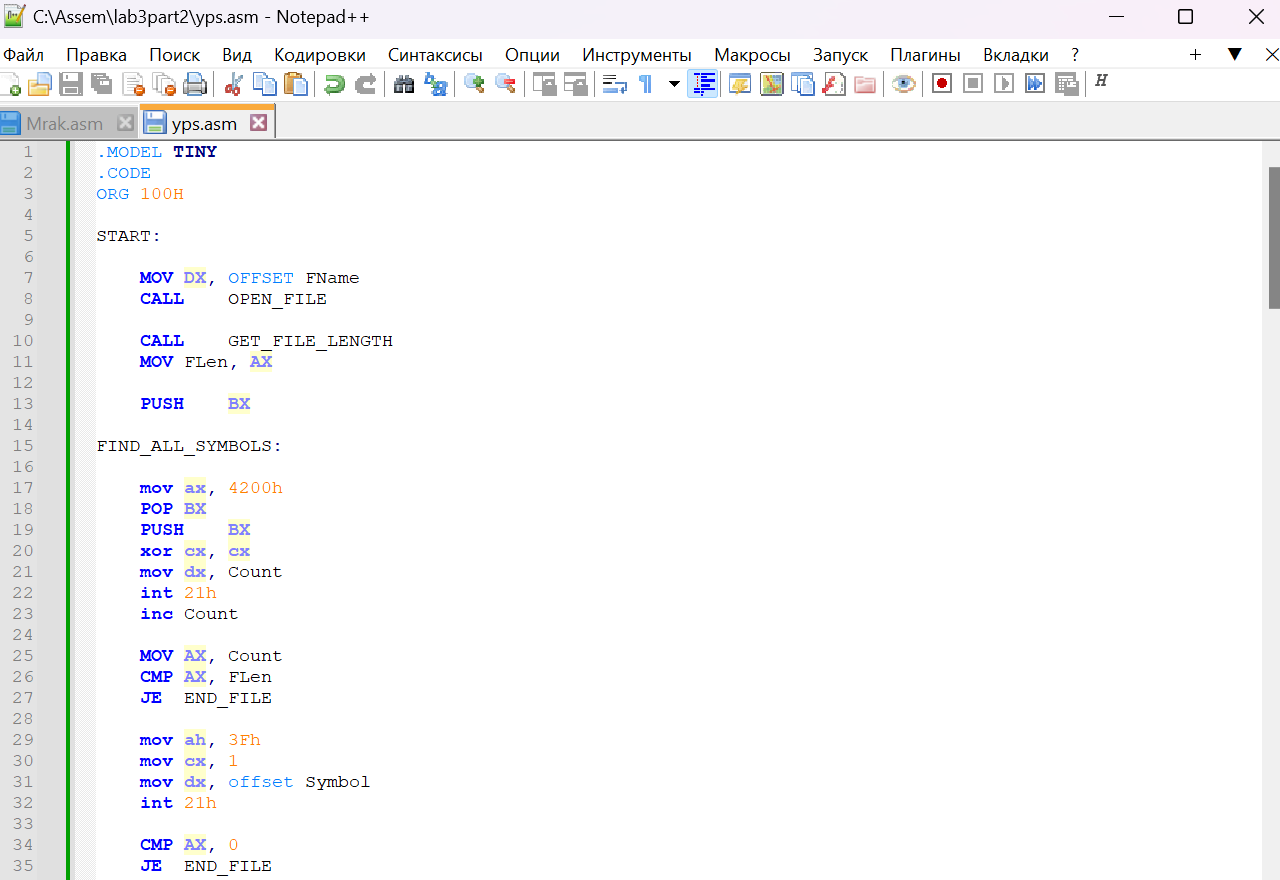
XOR DX, DX

INT 21H

RET

GET\_FILE\_LENGTH ENDP

END START

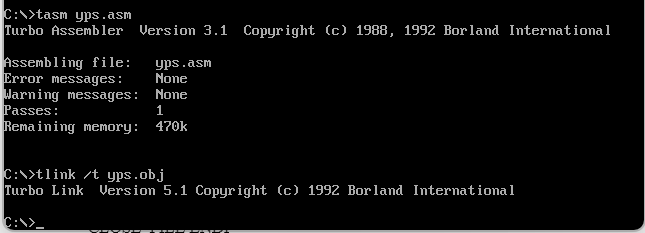


Немного сокращённый вариант пояснения:

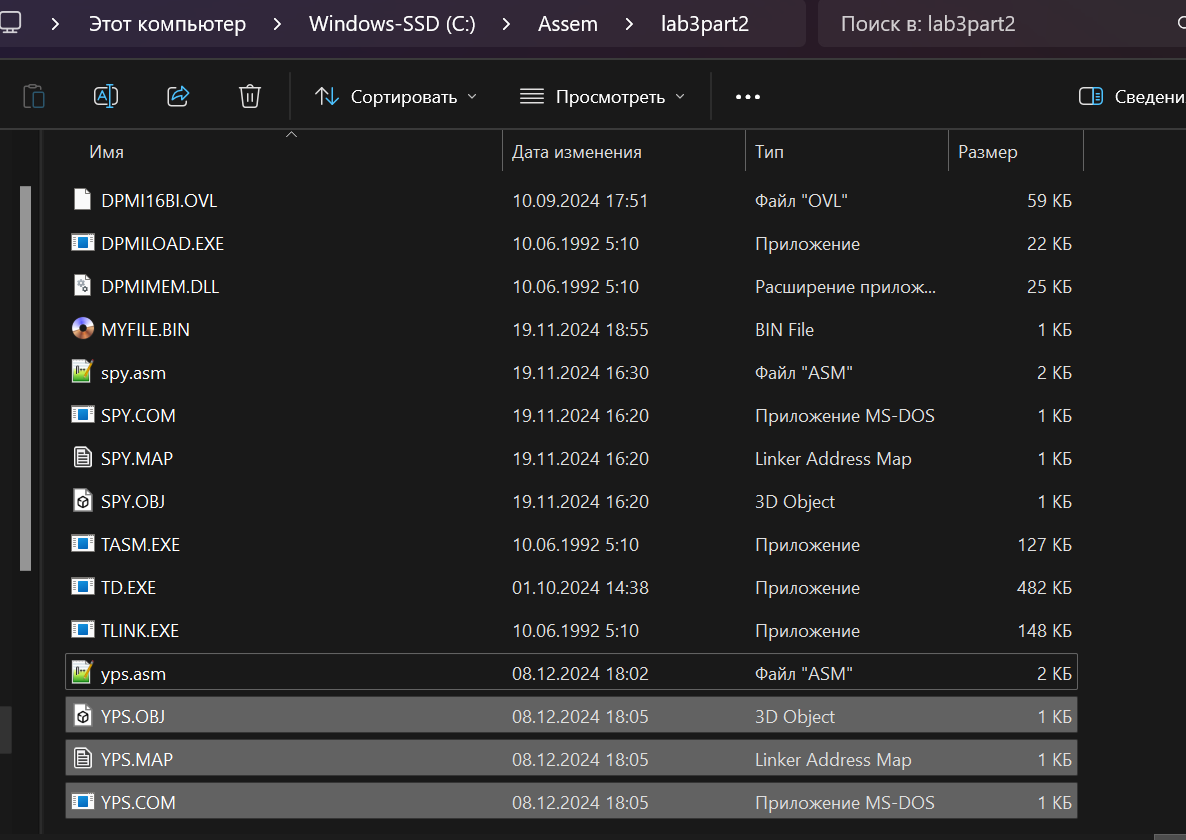
В переменных Codes, Symbols и ShiftSymbols хранятся коды клавиш, символы и символы с нажатым Shift соответственно, при этом данные в этих переменных соответствуют друг другу по индексам.

Файл читается по 1 байту, который представляет собой код клавиши. Если код найден в Codes, то по индексу беру символ из Symbols или ShiftSymbols в зависимости от состояния Shift и вывожу его на экран. Этот процесс продолжается до конца файла, с учётом проверок на Shift.

Теперь транслируем и линкуем:



Проверяем наличие файлов в папке:



Теперь проверим работоспособность. Для начала я заново запущу первый исполнимый модуль (SPY.COM) и введу, к примеру, «hello world».

Получаем примерно это:

