Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №4

по дисциплине ЭВМ и периферийные устройства

НЕПРИВИЛЕГИРОВАННЫЕ КОМАНДЫ ПРОЦЕССОРОВ х86

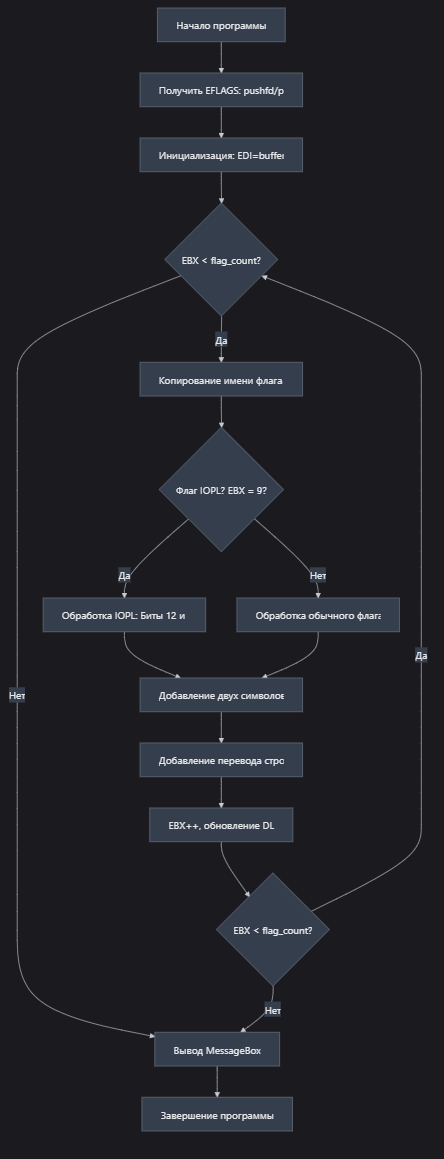
Вариант 28

Выполнил: Федоров Я.А.

студент группы ИВТ-42-23

Проверила: к.т.н.

Андреева А.А.

ГСА:

flowchart TD

A[Начало программы] --> B[Получить EFLAGS: pushfd/pop eax]

B --> C[Инициализация: EDI=buffer, EBX=0, DL=flag\_bits]

C --> D{EBX < flag\_count?}

D -->|Да| E[Копирование имени флага в буфер]

D -->|Нет| M[Вывод MessageBox]

E --> F{Флаг IOPL? EBX = 9?}

F -->|Да| G[Обработка IOPL: Биты 12 и 13]

F -->|Нет| H[Обработка обычного флага: Бит DL]

G --> I[Добавление двух символов '0'/'1']

H --> I

I --> J[Добавление перевода строки CRLF]

J --> K[EBX++, обновление DL]

K --> L{EBX < flag\_count?}

L -->|Да| D

L -->|Нет| M

M --> N[Завершение программы]

Код программы:

.386

.model flat, stdcall

includelib import32.lib

extrn MessageBoxA:near

extrn ExitProcess:near

MessageBox equ MessageBoxA

MB\_OK equ 0

.data

buffer db 512 dup(0) ; буфер для текста

title\_str db 'EFLAGS state :',0

flag\_ptrs dd offset flag\_cf, offset flag\_pf, offset flag\_af, offset flag\_zf, offset flag\_sf, offset flag\_tf, offset flag\_if, offset flag\_df, offset flag\_of, offset flag\_iopl, offset flag\_nt, offset flag\_rf, offset flag\_vm

flag\_cf db "CF",0

flag\_pf db "PF",0

flag\_af db "AF",0

flag\_zf db "ZF",0

flag\_sf db "SF",0

flag\_tf db "TF",0

flag\_if db "IF",0

flag\_df db "DF",0

flag\_of db "OF",0

flag\_iopl db "IOPL",0

flag\_nt db "NT",0

flag\_rf db "RF",0

flag\_vm db "VM",0

flag\_bits db 0,2,4,6,7,8,9,10,11,12,14,16,17

flag\_count equ 13

.code

start:

; Получаем EFLAGS

pushfd

pop eax

mov edi, offset buffer ; указатель на буфер

xor ebx, ebx ; индекс текущего флага

next\_flag:

cmp ebx, flag\_count

jge show\_message ; если все флаги обработаны, показать MessageBox

mov esi, [flag\_ptrs + ebx\*4] ; Текущий адрес имени флага

copy\_name:

mov al, [esi]

cmp al, 0

je after\_name

mov [edi], al

inc esi

inc edi

jmp copy\_name

after\_name:

mov byte ptr [edi], ':'

inc edi

mov byte ptr [edi], ' '

inc edi

movzx edx, byte ptr flag\_bits[ebx] ; edx = номер бита (0..31)

; если это IOPL печатаем 2 бита

cmp ebx, 9

jne single\_bit

;IOPL

bt eax, 13 ; старший бит

setc al

add al, '0'

mov [edi], al

inc edi

bt eax, 12 ; младший бит

setc al

add al, '0'

mov [edi], al

inc edi

jmp after\_value

single\_bit:

bt eax, edx ; CF бит eax[edx]

setc cl ; CL = 0/1

add cl, '0'

mov [edi], cl

inc edi

after\_value:

mov byte ptr [edi], 13

inc edi

mov byte ptr [edi], 10

inc edi

inc ebx ; следующий флаг

jmp next\_flag

show\_message:

; Показываем MessageBox

push 0

push offset title\_str

push offset buffer

push MB\_OK

call MessageBox

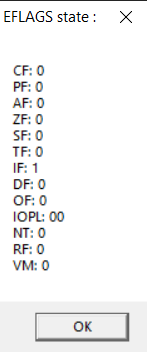
; Завершение процесса

push 0

call ExitProcess

end start

результат программы:



Вывод: Научился работать с непривилегированными командами.