**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе № 5**

**по дисциплине «Операционные системы»**

Тема: **Сопряжение стандартного и пользовательского обработчиков прерываний.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0382 |  | Диденко Д. В. |
| Преподаватель |  | Ефремов М. А. |

Санкт-Петербург

2022

**Цель работы.**

Исследование возможности встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры. Пользовательский обработчик прерывания получает управление по прерыванию (int 09h) при нажатии клавиши на клавиатуре. Он обрабатывает скан-код и осуществляет определенные действия, если скан-код совпадает с определенными кодами, которые он должен обрабатывать. Если скан-код не совпадает с этими кодами, то управление передаётся стандартному прерыванию

**Постановка задачи.**

**Шаг 1.** Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет такие же функции, как в программе ЛР 4, а именно:

Программа должна содержать код устанавливаемого прерывания в виде удаленной процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

* Сохранить значения регистров в стеке при входе и восстановить их при выходе.
* При выполнении тела процедуры анализируется скан-код.
* Если этот код совпадает с одним из заданных, то требуемый код записывается в буфер клавиатуры.
* Если этот код не совпадает ни с одним из заданных, то осуществляется передача управления стандартному обработчику прерывания.

**Шаг 2.** Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания 09h установлен. Работа прерывания проверяется введением различных символов, обрабатываемых установленным обработчиком и стандартным обработчиком.

**Шаг 3.** Также необходимо проверить размещение прерывания в памяти. Для этого запустите программу ЛР 3, которая отображает карту памяти в виде с писка блоков МСВ. Полученные результаты поместите в отчет.

**Шаг 4.** Запустите отлаженную программу еще раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчет.

**Шаг 5.** Запустите отлаженную программу с ключом выгрузки и убедитесь, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом освобождена. Для этого также следует запустить программу ЛР 3. Полученные результаты поместите в отчет.

**Выполнение работы.**

Были написаны строки для вывода информации:

* STR\_IST\_LOAD DB 'Iterrapt is not load', 0AH, 0DH,'$'
* STR\_ALR\_LOAD DB 'Iterrapt is already loaded', 0AH, 0DH,'$'
* STR\_SUC\_LOAD DB 'Iterrapt has been loaded', 0AH, 0DH,'$'
* STR\_IS\_UNLOAD DB 'Iterrapt is unloaded', 0AH, 0DH,'$'

Переменные для хранения флагов:

* flag db 0 – флаг удаления;
* flag\_load db 0 – флаг загрузки.

Переменные, хранящиеся в прерывании:

* PSP dw ? – сохранение адреса PSP;
* KEEP\_IP dw 0 – сохранение данных исходного прерывания;
* KEEP\_CS dw 0 – сохранение данных исходного прерывания;
* ITERRUPT\_ID dw 8f17h – уникальный идентификатор прерывания;
* STR\_ITERRUPT db SUCCESS WORK LAB! $' – строка вывода в прерывании;
* REQ\_KEY db 3Bh - скан код клавиши f1;
* STR\_INDEX db 0 – счётчик для вывода строки
* KEEP\_SS dw ? – для работы стека прерывания;
* KEEP\_SP dw ? – для работы стека прерывания;
* KEEP\_AX dw ? – для работы стека прерывания;
* ITERRUPT\_STACK dw 32 dup (?) –стек прерывания;
* END\_IT\_STACK dw ? – конец стека прерывания;

Были составлены функции (см. табл.1)

Таблица 1 – функции в программе

|  |  |
| --- | --- |
| Процедура | Описание |
| MY\_ITERRUPT | Резидентное прерывание, которое загружается в память и выполняет вывод символа от сообщения при нажатии на f1 |
| WRITE\_STRING | Вывод строки на экран |
| LOAD\_FLAG | Проверка на наличия флага “/un” |
| IS\_LOAD | Проверка на загрузку пользовательского прерывания в память |
| LOAD\_ITERRAPT | Сохранение первоначального прерывания и загрузка пользовательского прерывания в память |
| UNLOAD\_ITERRAPT | Выгрузка пользовательского прерывания из памяти, а также освобождение памяти и восстановление первоначальных прерываний |
| MAIN | Главная функция |

В результате выполнения были получены следующие значения(рис.1-3):

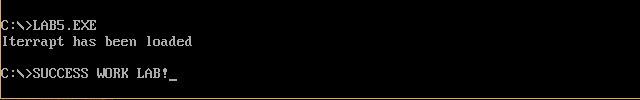


Рисунок 1 – прерывание загружено в память и пример вывода для нескольких нажатий f1

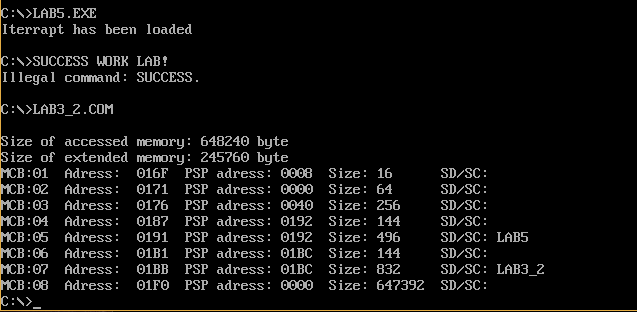


Рисунок 2 – проверка на расположение в памяти

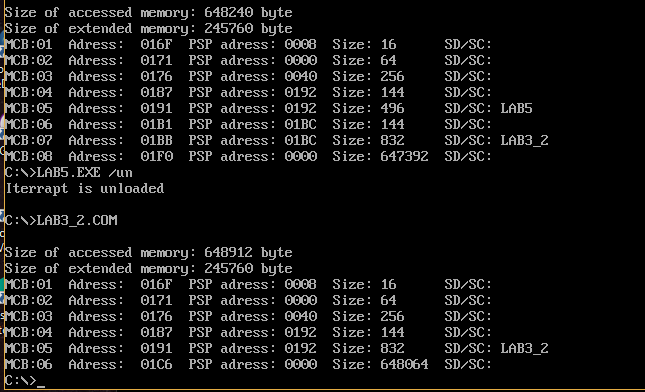


Рисунок 3 – выгрузка прерывания

**Выводы.**

В ходе лабораторной работы были исследованы возможности встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры. Загружается и выгружается резидент, а также производится проверка флагов и загрузки прерывание в память. С помощью rout при нажатии на клавишу f1 на экран посимвольно выводится строка, определённая в этом прерывании.

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какого типа прерывания использовались в работе?

В данной работе использовались прерывания функции DOS – int 21h и прерывания функции BIOS.

1. Чем отличается скан код от кода ASCII?

Скан-код – уникальное число-идентификатор клавиши, используется для определение нажатой клавиши контроллером клавиатуры. Контроллер пересылает скан-код в порт.

Код ASCII – это уникальный код для каждого символа.

Скан код характеризуете клавишу, а код ANCSII – символ.