

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Операционные системы»
Тема: Сопряжение стандартного и пользовательского обработчиков
прерываний

Студент гр. 7383

Лосев М.Л.

Преподаватель

Ефремов М.А.

Санкт-Петербург

Постановка задачи.

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 передается стандартному прерыванию.

Сведения об используемых функциях и структурах данных.

BYTE_TO_HEX – переводит значение регистра AL в его запись в шестнадцатеричной с/с, помещает ее в AX

WRD_TO_HEX – переводит значение регистра AX в его запись в шестнадцатеричной с/с, помещает ее в память так, что DI указывает на младшую цифру.

BYTE_TO_DEC – переводит значение регистра AL в его запись в десятичной с/с, помещает результат в память так, что SI указывает на младшую цифру.

OUTPUT_PROC – вызывает прерывание DOS вывода строки.

fix_letter_case – приводит заглавную букву в CL в соответствие с состоянием клавиатуры (CapsLock и shift).

write_char_to_buffer – записывает символ из CL в буфер клавиатуры.

keyboard_handler – резидентный обработчик прерывания. Увеличивает счетчик на 1, выводит его на экран.

KEEP_HANDLER – сохраняет в памяти адрес обработчика прерывания 09h.

RECOVER_HANDLER – восстанавливает обработчик прерывания 09h, адрес которого хранится в памяти.

SET_HANDLER – устанавливает вектор прерывания 09h, равный DX:DS.

KEEP – выходит из программы, сохранив в памяти резидентный обработчик прерывания.

check_handler – проверяет, установлен ли обработчик прерывания keyboard_handler по его сигнатуре. Результат в ZF.

check_tail – проверяет, равен ли хвост командной строки “ /un”. Результат в ZF.

unload_handler – восстанавливает сохраненный в памяти вектор стандартного обработчика прерывания 09h, выгружает из памяти переменные среды резидента и самого резидента.

push_registers – макрос, который сохраняет состояние регистров AX, SS, SP в ячейках памяти, устанавливает SS, SP так, что они указывают на вершину стека резидента, и сохраняет состояние остальных регистров в стек резидента.

pop_registers – макрос, который восстанавливает значения всех регистров перед завершением выполнения резидентного обработчика прерывания.

Последовательность действий, выполняемых утилитой.

Утилита выполняет следующие действия:

- 1) Проверяет, установлено ли пользовательское прерывание с вектором 09h.
- 2) Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляет выход по функции 4Ch прерывания int 21h.
- 3) Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h
- 4) Выгружает прерывание по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Результаты.

Скриншоты с результатами представлены на рис. 1 – рис. 2.

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Users\mikha\Documents\OPERATING_SYST_2019\DOC_DIR\TASM\BIN"
Drive C is mounted as local directory C:\Users\mikha\Documents\OPERATING_SYST_2019\DOC_DIR\TASM\BIN\

Z:\>C:

C:\>lab5
Custom interrupt handler had been set
C:\>mcb
Available memory: 648016 bytes
Extended memory: 15360 kb
MCB chain:
Address Owner PSP Size Name Type
016F 0008 16 4D
0171 0000 64 4D
0176 0040 256 4D
0187 0192 144 4D
0191 0192 736 LAB5 4D
01C0 01CB 144 4D
01CA 01CB 648000 MCB 5A

C:\>
```

Рисунок 1. Установка резидентного обработчика прерывания, ввод символов и проверка размещения обработчика в памяти.

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Z:\>MOUNT C "C:\Users\mikha\Documents\OPERATING_SYST_2019\DOC_DIR\TASM\BIN"
Drive C is mounted as local directory C:\Users\mikha\Documents\OPERATING_SYST_2019\DOC_DIR\TASM\BIN\

Z:\>C:

C:\>lab5
Custom interrupt handler had been set
C:\>lab5
Custom interrupt handler is already set
C:\>lab5 /un
Handler is unloaded
C:\>mcb
Available memory: 648928 bytes
Extended memory: 15360 kb
MCB chain:
Address Owner PSP Size Name Type
016F 0008 16 4D
0171 0000 64 4D
0176 0040 256 4D
0187 0192 144 4D
0191 0192 648912 MCB 5A

C:\>
```

Рисунок2. Попытка повторной установки резидентного прерывания, его выгрузка из памяти и проверка выгрузки.

Контрольные вопросы.

1) Какого типа прерывания использовались в работе?

Обоих типов: программные (int 21h, int 16h) и аппаратные (прерывание от клавиатуры int 09h).

2) Чем отличается скан-код от кода ASCII?

Во-первых, коды таблицы ASCII не имеют привязки к таким событиям, как, например, нажатие клавиши, то есть не кодируют эти события, а только печатные и управляющие символы. А скан-коды кодируют события.

Во-вторых, скан-коды не кодируют регистр букв (для того, чтобы понять, какой символ введен нажатием клавиши, надо учитывать не только скан-код этой клавиши, но и состояние клавиатуры, например, нажат ли CapsLock), а коды таблицы ASCII кодируют регистр буквы.

Заключение.

Была реализована утилита, которая умеет создавать резидентный обработчик прерывания 09h (который получает из порта байт и в зависимости от его значения либо сам обрабатывает его, либо делегирует эту задачу стандартному обработчику прерывания 09h), проверять, установлен ли он, восстанавливать стандартный обработчик прерывания 09h и выгружать резидента из памяти.