

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и систем»
ТЕМА: Разработка собственного прерывания.

Студент гр. 9382

Герасев Г.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить команды для работы с прерываниями в ассемблере, написать собственное прерывание.

Теоретические сведения:

Прерывание - это процесс вызова процедур для выполнения некоторой задачи, обычно связанной с обслуживанием некоторых устройств (обработка сигнала таймера, нажатия клавиши и т.д.). Когда возникает прерывание, процессор прекращает выполнение текущей программы (если ее приоритет ниже) и запоминает в стеке вместе с регистром флагов адрес возврата(CS:IP) - места, с которого будет продолжена прерванная программа.

Затем в CS:IP загружается адрес программы обработки прерывания и ей передается управление. Адреса 256 программ обработки прерываний, так называемые векторы прерывания, имеют длину по 4 байта (в первых двух хранится значение IP , во вторых - CS) и хранятся в младших 1024 байтах памяти. Программа обработки прерывания должна заканчиваться инструкцией IRET (возврат из прерывания), по которой из стека восстанавливается адрес возврата и регистр флагов.

Программа обработки прерывания - это отдельная процедура, имеющая структуру:

`SUBR_INT PROC FAR`

`PUSH AX ; сохранение изменяемых регистров`

`<действия по обработке прерывания>`

POP AX ; восстановление регистров

...

MOV AL, 20H

OUT 20H,AL

IRET

SUBR_INT ENDP

Две последние строки обработчика прерывания, указанные перед командой IRET выхода из прерывания, необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Замечание: в лабораторной работе действиями по обработке прерывания может быть вывод на экран некоторого текста, вставка цикла задержки в вывод сообщения или включение звукового сигнала.

Программа, использующая новые программы обработки прерываний при своем завершении должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. В соответствии с этим, программа должна содержать следующие инструкции:

; -- в сегменте данных

KEEP_CS DW 0 ; для хранения сегмента

KEEP_IP DW 0 ; и смещения вектора прерывания

; -- в начале программы

MOV AH, 35H ; функция получения вектора

MOV AL, 1CH ; номер вектора

INT 21H

MOV KEEP_IP, BX ; запоминание смещения

MOV KEEP_CS, ES ; и сегмента вектора прерывания

Для установки адреса нового обработчика прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером.

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 60H ; номер вектора

INT 21H ; меняем прерывание

POP DS

Далее может выполняться вызов нового обработчика прерывания. В конце программы восстанавливается старый вектор прерывания

CLI

PUSH DS

MOV DX, KEEP_IP

MOV AX, KEEP_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 1CH

INT 21H ; восстанавливаем старый вектор прерывания

POP DS

STI

Задание:

10 Вариант – 4А

4 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

А - Печать сообщения на экране;

Ход работы:

При разработке программы были использованы следующие команды:

Инструкция OUT выводит данные из регистра AL или AX (ИСТОЧНИК) в порт ввода-вывода. Номер порта должен быть указан в ПРИЁМНИКЕ.

Тестирование.

Вводные данные	Результат
	A can of beer!
	A can of beer!

	A can of beer!
	A can of beer!
	A can of beer!
	A can of beer!
	A can of beer!

Выводы.

В результате выполнения лабораторной работы был разработан код, определяющий собственное прерывание. Были улучшены навыки письма в ассемблере.

Приложение.

Текст файла MAIN.LST

[illegible]


```

007E E6 20          out 20h,al
0080 CF              iret
0081                Output ENDP

0081                Main PROC FAR
0081 1E            push ds
0082 2B C0        sub ax,ax
0084 50            push ax
0085 B8 ---- R    mov ax,data
0088 8E D8        mov ds, ax

008A B8 3523      mov ax,3523h
008D CD 21        INT 21H
008F 89 1E 0002 R MOV KEEP_IP, BX ; Ð·Ð°Ð¸ÐŸÐœÐžÐ°ÐœÐšÐ¼ Ñ† ÐœÐ
µÑ† Ð¼ÐœÐžÑ†
0093 8C 06 0000 R MOV KEEP_CS, ES ; Ðž Ñ† Ð¼Ð³ÐœÐ¼ÐœÑ† Ð° Ð² Ð¼ÐºÑ†
ÐŸÑ† Ð° ÐžÑ† Ð¼Ñ† Ñ† Ñ† Ð²Ð°ÐœÐžÑ†
0097 1E            PUSH DS
0098 BA 0000 R    MOV DX, OFFSET Output ; Ñ† ÐœµÑ† Ð¼ÐœÐžÐ¼ ÐžÐ»Ñ†
Ñ†Ñ† ÐŸÑ† Ð¼ÐžÑ† Ñ† Ñ† Ð² DX
009B B8 ---- R    MOV AX, SEG Output ; Ñ† Ð¼Ð³ÐœÐ¼ÐœÑ† ÐžÑ† ÐžÑ† ÐŸÑ† µ
ÐžÑ† Ñ† Ñ†
009E 8E D8        MOV DS, AX ; ÐžÐŸÐœµÑ† Ð° Ð°Ðœ Ð² DS
00A0 B8 2508      mov ax,2508h
00A3 CD 21        INT 21H ; ÐœµÐœÑ† Ð¼Ðœ ÐžÑ† Ð¼Ñ† Ñ† Ñ† Ð²Ð°ÐœÐžÐ¼
00A5 1F            POP DS

00A6                waiting:
00A6 B4 01        mov ah,1h
00A8 CD 21        int 21h
00AA 3C 1B        cmp al,27
00AC 75 13        jne nextstep

00AE FA            CLI
00AF 1E            PUSH DS
00B0 8B 16 0002 R MOV DX, KEEP_IP
00B4 A1 0000 R    MOV AX, KEEP_CS
00B7 8E D8        MOV DS, AX
00B9 B8 2508      mov AX,2508h
00BC CD 21        INT 21H ; Ð²ÐŸÑ† Ñ† Ñ† Ñ† Ð°ÐœÐ°Ð²Ð»Ð²Ð°Ð¼Ðœ Ñ† Ñ† Ð
°Ñ† Ñ† Ð¹ Ð²Ð¼ÐºÑ† ÐŸÑ† ÐžÑ† Ð¼Ñ† Ñ† Ñ† Ð²Ð°ÐœÐžÑ†
00BE 1F            POP DS
00BF FB            STI
00C0 CB            ret

00C1                nextstep:
00C1 EB E3        jmp waiting

00C3                Main ENDP
00C3                CODE ENDS
                    END Main

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0800	PARA	STACK	
CODE	00C3	PARA	NONE	
DATA	0015	PARA	NONE	

Symbols:

N a m e	Type	Value	Attr
ISTACK	L WORD	0009	CODE Length = 001E
KEEP_CS	L WORD	0000	DATA
KEEP_IP	L WORD	0002	DATA
MAIN	F PROC	0081	CODE Length = 0042
MESSAGE	L BYTE	0004	DATA
NEXTSTEP	L NEAR	00C1	CODE
OUTPUT	F PROC	0000	CODE Length = 0081
START	L NEAR	0045	CODE
ST_AX	L WORD	0005	CODE
ST_SP	L WORD	0007	CODE
ST_SS	L WORD	0003	CODE
WAITING	L NEAR	00A6	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	MAIN	
@VERSION	TEXT	510	

94 Source Lines
94 Total Lines
20 Symbols

48000 + 459260 Bytes symbol space free

0 Warning Errors
0 Severe Errors