

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: Написание собственного прерывания.

Студентка гр. 9383

Пономаренко С. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2020

Цель работы.

Изучить прерывания на языке Ассемблер, написать собственное прерывание.

Задание.

Вариант 1С

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

1 - 1Ch - прерывание от часов - генерируется автоматически операционной системой 18 раз в сек;

2 - 60h - прерывание пользователя - должно генерироваться в программе;

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C ;

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

Буква определяет действия, реализуемые программой обработки прерываний:

A - Печать сообщения на экране;

B - Выдача звукового сигнала;

C - Приостановить вывод на экран (вставить цикл задержки).

Ход работы.

Процедура прерывания:

Сохраняем изменяемые регистры в стеке. Помещаем в регистры CX, DX значения, из которых формируется время задержки. Помещаем в регистр AX функцию 86H. Вызываем прерывание 15H. Восстанавливаем из стека помещенные туда значения.

Тестирование.

[C:\](#)>lb5.exe → (через 3 секунды) C:\>

Вывод.

Изучили прерывания на языке Ассемблер, написали собственное прерывание subr_int, создающее задержку.

Приложение А.
Код написанной программы.

EOF EQU '\$'

AStack SEGMENT STACK
DB 1024 DUP(?)
AStack ENDS

DATA SEGMENT
KEEP_CS DW 0 ; для хранения сегмента вектора прерывания
KEEP_IP DW 0 ; и смещения вектора прерывания
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

SUBR_INT PROC FAR
PUSH AX ; сохранение изменяемых регистров
PUSH CX ; сохранение изменяемых регистров
PUSH DX ; сохранение изменяемых регистров
MOV CX, 0033H
MOV DX, 00FFH
; 0007H и 8480H - 1 сек // 0033H и 00FFH - 3 сек // 0088H и 00FFH - 8 сек // 00FF
и 0FFF - 16 сек
MOV AH, 86H
; MOV AL, 0
INT 15H
POP DX ; восстановление регистров
POP CX ; восстановление регистров
POP AX ; восстановление регистров+
MOV AL, 20H
OUT 20H,AL
IRET
SUBR_INT ENDP

MAIN PROC FAR
PUSH DS ; сохранение адреса начала PSP в стеке для последующего
восстановления по команде get
MOV AX, DATA ; загрузка сегментного регистра данных
MOV DS, AX
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
PUSH DS
MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
MOV AX, SEG SUBR_INT ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора
MOV AL, 1CH ; номер вектора
INT 21H ; меняем прерывание
POP DS
INT 1CH
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
RET
MAIN ENDP
CODE ENDS
END MAIN
```