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

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Организация ЭВМ и систем»

Тема: Разработка собственного прерывания

Студент гр. 9383 _____ Камзолов Н.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Познакомиться с процессами прерывания. Разработать собственный процесс прерывания.

Краткие 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. 12

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
```

Текст задания(Вариант 4).

Написать собственное прерывание.

60h - прерывание пользователя - должно генерироваться в программе;

Печать собственного сообщения на экран.

Ход работы.

Написана программа, которая переопределяет прерывание 60h так, чтобы оно выводило на экран “Hello world”. По окончании программы происходи

Выводы.

Произошло ознакомление с процессами прерывания. Разработан собственный процесс прерывания.

ПРИЛОЖЕНИЕ А
КОД ПРОГРАММЫ

Название файла: Source.asm

```
DATA SEGMENT
        KEEP_CS DW 0 ; для хранения сегмента
        KEEP_IP DW 0 ; и смещения вектора прерывания
        HELLO   DB 'Hello Worlds!  $';,10,13,'$'
DATA ENDS


AStack    SEGMENT  STACK
          DW 12 DUP(?)    ; ▒▒▒▒▒ 12 ▒▒ ▒▒▒▒▒▒
AStack    ENDS

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

SUBR_INT PROC FAR
        PUSH AX ; сохранение изменяемых регистров
        PUSH DX;

        MOV   DX, OFFSET HELLO
        MOV   AH,9
        int   21h  ; Вызов функции DOS по прерыванию

        POP DX;
        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
          sub   AX,AX    ; > для последующего восстановления по
          push  AX       ;/  команде ret, завершающей процедуру.
          mov   AX,DATA             ; Загрузка сегментного
          mov   DS,AX               ; регистра данных.

          MOV AH, 35H ; функция получения вектора
          MOV AL, 60H ; номер вектора
          INT  21H  ;  возвращает  текущее  значение  вектора
прерывания
          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, 60H ; номер вектора
          INT 21H ; меняем прерывание
          POP DS

          int 60H; вызов измененного 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

        RET
Main      ENDP
CODE      ENDS
          END Main
```