

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Операционные системы»
Тема: Обработка стандартных прерываний

Студентка гр. 0381

Шиняева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

Исследовать структуры обработчиков стандартных прерываний, построить обработчик прерываний сигналов таймера.

Постановка задачи

Требуется написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

- Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
- Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляет выход по функции 4Ch прерывания int 21h.
- Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
- Выгрузка прерывания 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.

Код пользовательского прерывания должен выполнять следующие функции:

- Сохранять значения регистров в стеке при входе и восстановить их при выходе.
- При выполнении тела процедуры накапливать общее суммарное число прерываний и выводить на экран. Для вывода на экран следует использовать прерывание int 10h, которое позволяет непосредственно выводить информацию на экран.

Выполнение работы

Для выполнения данной работы были реализованы следующие функции:

- `inter` – код резидентного обработчика прерывания `1Ch`;
- `load` - для загрузки обработчика прерываний;
- `unload` - для выгрузки обработчика прерываний;
- `isParam` - для проверки наличия ключа выгрузки;
- `isLoad` - для проверки установки обработчика прерываний;
- `PRINT_STR` - для вывода строки, адрес которой лежит в регистре `DX`;
- `main` - для выполнения поставленной в данной лабораторной работе задачи.

Для вывода информации на экран были созданы следующие строки:

- `ls`, хранящая в себе строку `'Interrupt loaded successfully$'`;
- `us`, хранящая в себе строку `'Interrupt loaded successfully$'`;
- `ial`, хранящая в себе строку `'Interrupt loaded successfully$'`;
- `iau`, хранящая в себе строку `'Interrupt loaded successfully$'`;

При запуске программы с помощью процедуры `isParam` проверяется наличие ключа выгрузки `'Interrupt loaded successfully$';/un'`. В случае, если его нет, переменная `flag` остаётся равной 0, в противном случае – становится равной 1. В зависимости от значения этой переменной будет происходить загрузка или выгрузка резидентной функции. В случае, если ключа нет, но программа уже загружена, что проверяется так же с помощью переменной `flag` после вызова процедуры `isLoad`, выводится сообщение о том, что обработчик уже загружен, и программа завершается. Если же обработчик не был загружен, то вызывается процедура `load`, которая устанавливает резидентную функцию и настраивает вектор прерываний. В случае попытки выгрузки вновь вызывается процедура `isLoad` и проверяется переменная `flag`. Если резидентная программа не была установлена, выводится соответствующее сообщение, в противном случае - вызывается процедура `unload`, которая восстанавливает необходимые регистры, вектор прерываний и освобождает память.

Результаты работы программы представлены на рисунках 1-4. Для проверки установки обработчика прерываний используется программа L2.COM из предыдущей лабораторной работы.

```
C:\>lab4.exe
Interrupt loaded successfully
C:\>l2.com
Available memory (bytes): 647792
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 944 SC/SD: LAB4
Address: 01CD PSP address: 01D8 Size: 144 SC/SD:
Address: 01D7 PSP address: 01D8 Size: 7552 SC/SD: L2
Address: 03B0 PSP address: 0000 Size: 640224 SC/SD: ìF■PìF.
```

Рис. 1 - Результат загрузки прерывания в память

```
C:\>lab4.exe
Interrupt already loaded
C:\>l2.com
Available memory (bytes): 647792
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 944 SC/SD: LAB4
Address: 01CD PSP address: 01D8 Size: 144 SC/SD:
Address: 01D7 PSP address: 01D8 Size: 7552 SC/SD: L2
Address: 03B0 PSP address: 0000 Size: 640224 SC/SD: ìF■PìF.
```

Рис. 2 - Результат повторной загрузки прерывания в память

```
C:\>lab4.exe /un
Interrupt unloaded successfully
C:\>l2.com
Available memory (bytes): 648912
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 6432 SC/SD: L2
Address: 0324 PSP address: 0000 Size: 642464 SC/SD: ìF■PìF.
```

Рис. 3 - Результат выгрузки прерывания из памяти

```

C:\>lab4.exe /un
Interrupt already unloaded
C:\>l2.com
Available memory (bytes): 648912
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 6432 SC/SD: L2
Address: 0324 PSP address: 0000 Size: 642464 SC/SD:

```

Рис. 4 - Результат повторной выгрузки прерывания из памяти

Контрольные вопросы

1. Как реализован механизм прерывания от часов?

Каждый такт из таймера вычитается определённое значение. Когда значение достигает 0, возникает прерывание от таймера. При возникновении прерывания процессор запоминает в стеке адрес возврата (CS:IP) и регистр флагов. Затем в CS:IP загружается адрес обработчика прерывания и выполняется его код. В конце регистры восстанавливаются, и процессор возвращается на выполнение прерванной программы.

2. Какие прерывания использовались в работе?

В данной работе использовались аппаратное прерывание 21h с вектором 1Ch, а также пользовательские прерывания 10h и 21h.

Вывод

В ходе лабораторной работы была исследована обработка стандартных прерываний, а также построен обработчик прерываний сигналов таймера, которые генерируются аппаратурой через определённые интервалы времени. Программа загружает и выгружает резидент, а также производится проверка флагов и загрузки прерывание в память.