

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

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

Студент гр. 8382

Щеглов А.С.

Преподаватель

Ефремов. М.А.

Санкт-Петербург

2020

Цель работы.

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерываний хранит адрес подпрограммы обработчика прерываний. При возникновении прерывания, аппаратура компьютера передает управление по соответствующему адресу вектора прерывания. Обработчик прерываний получает управление и выполняет соответствующие действия. В лабораторной работе №4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определенные интервалы времени и, при возникновении такого сигнала, возникает прерывание с определенным значением вектора. Таким образом, управление будет передано функции, чья точка входа записана в соответствующий вектор.

Выполнение работы.

В процессе выполнения лабораторной работы был написан и отлажен программный модель типа .exe, выполняющий следующие функции:

1. Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
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. Выгрузка прерывания по соответствующему значения параметра командной строки "/u". Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Результат работы программы в различных состояниях показан на рисунке 1-4. Состояние памяти при работе с обработчиком прерывания показано на рисунках 5-7.

```

C:\>lr4.exe
Loaded
timer = 00170
C:\>_

```

Рисунок 1. Запуск программы с установкой пользовательского прерывания

```

C:\>lr4.exe
Already loaded
timer = 00711
C:\>

```

Рисунок 2. Попытка повторной установки пользовательского прерывания.

```

C:\>lr4.exe /u
Unloaded
C:\>

```

Рисунок 3. Запуск программы с выгрузкой пользовательского прерывания.

```

C:\>lr4.exe
Loaded
timer = 00117
C:\>

```

Рисунок 4. Попытка повторной выгрузки пользовательского прерывания.

```

C:\>lr3_1.com
Available memory - 648912 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size - 16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE Size - 64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size - 256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size - 144 Bytes. Last 8 bytes -
MCB Type - 5A Sector - 0192 Size - 648912 Bytes. Last 8 bytes - LR3_1
C:\>_

```

Рисунок 5. Состояние памяти до загрузки прерывания.

```

C:\>lr3_1.com
Available memory - 647760 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size - 16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE Size - 64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size - 256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size - 144 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size - 976 Bytes. Last 8 bytes - LR4
MCB Type - 4D Sector - 01DA Size - 144 Bytes. Last 8 bytes -
MCB Type - 5A Sector - timer = 00281 647760 Bytes. Last 8 bytes - LR3_1
C:\>

```

Рисунок 6. Состояние памяти после загрузки прерывания.

```
C:\>lr4.exe /u
Unloaded

C:\>lr3_1.com
Available memory - 648912 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE  Size -      64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size -      256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -      144 Bytes. Last 8 bytes -
MCB Type - 5A Sector - 0192 Size - 648912 Bytes. Last 8 bytes - LR3_1
C:\>
```

Рисунок 7. Состояние памяти после выгрузки прерывания.

Контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Функция 1Ch прерывания int 21h берет по каждому тикку аппаратных часов (каждые 55 миллисекунд, приблизительно 18.2 раз в секунду) и сохраняет состояние регистров, определяет смещение прерывания в таблице векторов прерываний и помещает этот адрес в CS:IP. Далее обрабатывает само прерывание и после завершения работы управление возвращается прерванной программе.

2. Какого типа прерывания использовались в работе?

Функции 1Ch, 09h, 35h, 4Ch программного прерывания int 21h, а также некоторые функции прерывания int 10h.

Выводы.

В ходе работы был построен обработчик прерываний сигналов таймера. Также получены навыки работы написания резидентного обработчика прерывания.