

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

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

Студент гр.8304

Преподаватель

Самакаев Д.И.

Губкин А.Ф.

Санкт-Петербург

2020

Постановка задачи.

1. Формулировка

В лабораторной работе №4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определённые интервалы времени (примерно 18.2 раза в секунду) и при возникновении такого сигнала возникает прерывание с определённым значением вектора. Для выполнения работы необходимо написать и отладить модуль 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) Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
- 4) Выгрузка прерывания по соответствующему значению параметра в командной строке («/un»). Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождения памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Для демонстрации работы собственного прерывания код удалённой процедуры должен выполнять следующие функции:

- 1) Сохранять значения регистров в стеке при входе и восстановить их при выходе.
- 2) 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.

Функции и структуры данных:

count - запоминает количество вызванных прерываний.

flag - флаг, равный 1, если программа не является резидентной, и 0, если наоборот.

loaded - Сообщение о том, что программа только что была загружена в память.

unloaded – Сообщение о том, что программа была выгружена из памяти.

alreadyloaded - Сообщение о том, что программа уже была загружена в память ранее.

keep_1c - Переменная для хранения сегмента и смещения старого прерывания 1c.

2f. keep_2f - Переменная для хранения сегмента и смещения старого прерывания

keep_PSP - Переменная для хранения старого значения ES до того, как программа была оставлена резидентной в памяти.

outputAL - Вывод в текущее положение курсора символа из AL.

saveCurs - Запоминает в DX текущую позицию курсора.

setCurs - Устанавливает курсор в указанную в DX позицию.

My_2F - Собственный обработчик прерывания для 2F. Проверяет, была ли программа установлена резидентной в памяти.

My_1C - Собственный обработчик прерывания для 1C. Выводит в 33-ую позицию курсора количество прерываний, которые были вызваны. Прерывания генерируются системным таймером с частотой примерно 18.2 раза в секунду.

un_check - Проверяет, не указал ли пользователь флаг «/un» при вызове программы.

unload_inter - Восстанавливает сохранённые заранее обработчики прерываний и выгружает резидентную программу.

load_inter - Устанавливает новые обработчики прерывания, используя функцию 25h прерывания int 21h.

keep_inter - Запоминает старые обработчики прерывания, используя функцию 35h прерывания int 21h.

write_msg - Вывод сообщения на экран

Результат работы программы:

Запустим программу.

```
C:\>lab4.exe
Resident program is loaded
C:\>
```

Запустим программу из предыдущей работы.

```
C:\>lr3_3.com
Available memory: 633824 Bytes;
Extended memory: 15360 Kilobytes;
List of MCB:
Type of MCB: 4Dh; Segment's address: 0008h; MCB size: 16b;
Type of MCB: 4Dh; Segment's address: 0000h; MCB size: 64b;
Type of MCB: 4Dh; Segment's address: 0040h; MCB size: 256b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 144b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 14912b; LAB4
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 14144b;
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 14848b; LR3_3
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 65536b; LR3_3
Type of MCB: 5Ah; Segment's address: 0000h; MCB size: 567408b;
C:\>_
```

Программа lab4.exe остаётся резидентной. Попробуем запустить её ещё раз.

```

C:\>lr3_3.com
Available memory: 633824 Bytes;
Extended memory: 15360 Kilobytes;
List of MCB:
Type of MCB: 4Dh; Segment's address: 0008h; MCB size: 16b;
Type of MCB: 4Dh; Segment's address: 0000h; MCB size: 64b;
Type of MCB: 4Dh; Segment's address: 0040h; MCB size: 256b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 144b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 14912b; LAB4
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 14144b;
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 14848b; LR3_3
Type of MCB: 4Dh; Segment's address: 0541h; MCB size: 65536b; LR3_3
Type of MCB: 5Ah; Segment's address: 0000h; MCB size: 567408b;

C:\>lab4.exe
Resident program has already been loaded

C:\>_

```

Получили сообщение о том, что программа уже загружена. Запустим программу с командой /un.

```

C:\>lab4.exe /un
Resident program is unloaded

C:\>_

```

Получили сообщение о том, что программа выгружена. Видим, что программа действительно прекратила работу.

```

C:\>lab4.exe /un
Resident program is unloaded

C:\>lr3_3.com
Available memory: 648912 Bytes;
Extended memory: 15360 Kilobytes;
List of MCB:
Type of MCB: 4Dh; Segment's address: 0008h; MCB size: 16b;
Type of MCB: 4Dh; Segment's address: 0000h; MCB size: 64b;
Type of MCB: 4Dh; Segment's address: 0040h; MCB size: 256b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 144b;
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 848b; LR3_3
Type of MCB: 4Dh; Segment's address: 0192h; MCB size: 65536b; LR3_3
Type of MCB: 5Ah; Segment's address: 0000h; MCB size: 582496b;

C:\>

```

Ответы на контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Сохраняется состояние регистров, после чего определяется смещение вектора прерывания (первые два байта помещаются в IP, последние два – в CS),

управление передается по адресу CS:IP и обрабатывается соответствующее прерывание. После этого продолжается прерванная программа.

Вывод.

В ходе данной лабораторной работы было произведено сравнение структур COM и EXE файлов, исследованы различия в исходных текстах модулей COM и EXE, а также сравнены способы их загрузки в память.