

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. 8383

Ишанина Л.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

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 прерывания.

Выполнение работы.

Шаг 1. Был написан и отлажен программный модуль типа .EXE, который выполняет следующие функции:

- 1) Проверяет установлено ли пользовательское прерывание 1Ch.
- 2) Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляется 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/up. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Код устанавливаемого прерывания должен содержать функции:

- 1) Сохранения значения регистров в стеке при входе и восстановления их при выходе.
- 2) При выполнении тела процедуры накапливание общего суммарного числа прерываний и выводить на экран (использование int 10h).

Шаг 2. Запуск программы и проверка карты памяти при помощи 3 лабораторной работы. Результат запуска представлен на рис.1.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
C:\>link lab4.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB4.EXE]: lab4.exe
List File [NUL.MAP]: lab4.map
Libraries [.LIB]:

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

C:\>lr3_1.com
Available memory: 634128 bytes;
Extended memory: 15360 kilobytes;
List of MCB:
Type: 4Dh; Segment's address: 0008h; Size: 16;
Type: 4Dh; Segment's address: 0000h; Size: 64;
Type: 4Dh; Segment's address: 0040h; Size: 256;
Type: 4Dh; Segment's address: 0192h; Size: 144;
Type: 4Dh; Segment's address: 0192h; Size: 14608; LAB4
Type: 4Dh; Segment's address: 052Eh; Size: 14144;
Type: 5Ah; Segment's address: 052Eh; Size: 634128; LR3_1

C:\>
```

Рисунок 1 — Первый запуск.

Шаг 3. Повторный запуск, чтобы убедиться, что программа определяет установленный обработчик прерываний. Результат запуска представлен на рис.2.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB4.EXE]: lab4.exe
List File [NUL.MAP]: lab4.map
Libraries [.LIB]:

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

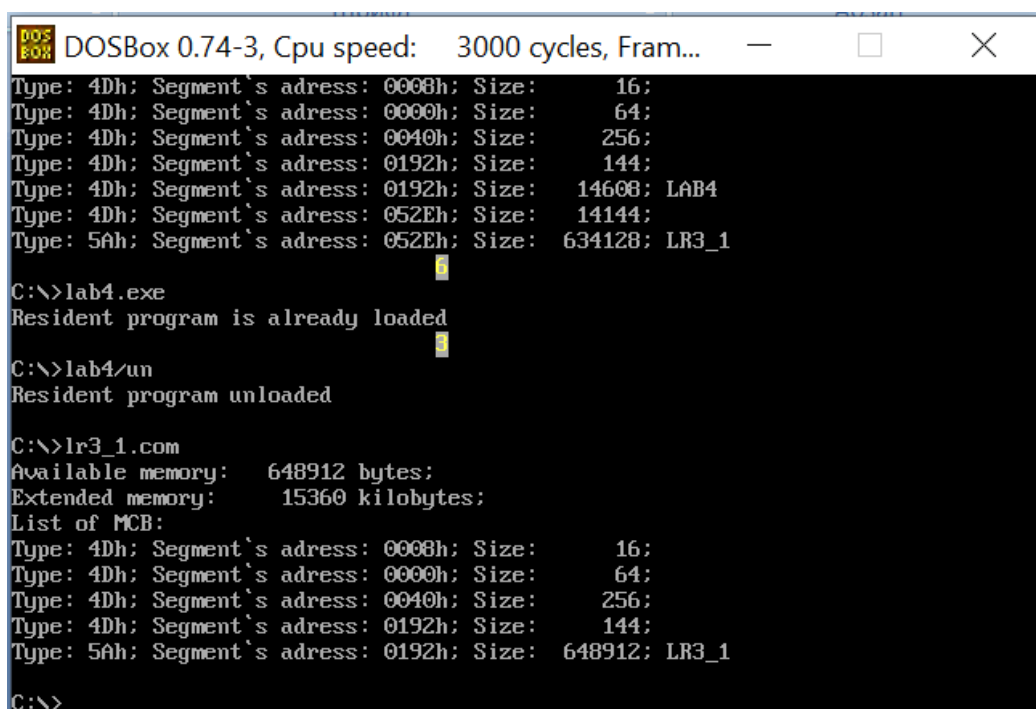
C:\>lr3_1.com
Available memory: 634128 bytes;
Extended memory: 15360 kilobytes;
List of MCB:
Type: 4Dh; Segment's address: 0008h; Size: 16;
Type: 4Dh; Segment's address: 0000h; Size: 64;
Type: 4Dh; Segment's address: 0040h; Size: 256;
Type: 4Dh; Segment's address: 0192h; Size: 144;
Type: 4Dh; Segment's address: 0192h; Size: 14608; LAB4
Type: 4Dh; Segment's address: 052Eh; Size: 14144;
Type: 5Ah; Segment's address: 052Eh; Size: 634128; LR3_1

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

C:\>
```

Рисунок 2 — Второй запуск.

Шаг 4. Запуск с ключом выгрузки. Результат запуска представлен на рис.3.

A screenshot of a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...". The window displays a memory dump at the top with segments 0008h, 0000h, 0040h, 0192h, and 052Eh. Below the dump, the command "C:\>lab4.exe" is entered, followed by the message "Resident program is already loaded". Then, "C:\>lab4/un" is entered, followed by "Resident program unloaded". Next, "C:\>lr3_1.com" is entered, showing available memory (648912 bytes) and extended memory (15360 kilobytes). A second memory dump is shown, with the last segment being 0192h of size 648912. The prompt "C:\>_" is visible at the bottom.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
Type: 4Dh; Segment's address: 0008h; Size: 16;
Type: 4Dh; Segment's address: 0000h; Size: 64;
Type: 4Dh; Segment's address: 0040h; Size: 256;
Type: 4Dh; Segment's address: 0192h; Size: 144;
Type: 4Dh; Segment's address: 0192h; Size: 14608; LAB4
Type: 4Dh; Segment's address: 052Eh; Size: 14144;
Type: 5Ah; Segment's address: 052Eh; Size: 634128; LR3_1
C:\>lab4.exe
Resident program is already loaded
C:\>lab4/un
Resident program unloaded
C:\>lr3_1.com
Available memory: 648912 bytes;
Extended memory: 15360 kilobytes;
List of MCB:
Type: 4Dh; Segment's address: 0008h; Size: 16;
Type: 4Dh; Segment's address: 0000h; Size: 64;
Type: 4Dh; Segment's address: 0040h; Size: 256;
Type: 4Dh; Segment's address: 0192h; Size: 144;
Type: 5Ah; Segment's address: 0192h; Size: 648912; LR3_1
C:\>_
```

Рисунок 3 — Третий запуск.

Шаг 5. Ответы на контрольные вопросы:

1. Как реализован механизм прерывания от часов?

Каждые 55 мс сначала сохраняется состояние регистров, затем определяется источник прерывания, определяющий в свою очередь адрес (смещение) вектора прерывания в таблице векторов прерываний (значения могут быть от 0000:0000 до 0000:03FF). Первые два байта помещаются в регистр IP, а вторые два байта – в CS. Затем управление передаётся по адресу CS:IP и происходит обработка соответствующего прерывания. После завершения обработки управление возвращается прерванной программе.

2. Какие прерывания использовались в работе?

В работе использовались пользовательские прерывания int 21h, int 10h, int 2Fh, а также аппаратное прерывание int 1Ch, возникающее каждые 55 мс по системному таймеру.

Выводы.

В ходе данной лабораторной работы я построила собственный обработчик прерывания для аппаратного прерывания 1Ch, происходящего по сигналу

системного таймера. Также я научилась оставлять программу (ту её часть, в которой содержится собственное прерывание) резидентной в памяти и проверять, была ли загружена программа в память, и выгружать её из памяти по команде, вводимой пользователем.