

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

Студент гр. 0381

Кирильцев Д.А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

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 прерывания.

Постановка задачи.

Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет 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 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 /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Программа должна содержать код устанавливаемого прерывания в виде удаленной процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

1. Сохраняет стек прерванной программы (регистры SS и SP) в рабочих переменных и восстановить при выходе.
2. Организовать свой стек.
3. Сохранить значения регистров в стеке при входе и восстановить их при выходе.
4. 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.
5. Функция прерывания должна содержать только переменные, которые она использует.

Таблица 1 — Процедуры в программе.

Процедура	Описание
custom_interruption	Пользовательское прерывание
unload_key_check	Проверка наличия ключа выгрузки
unload_int	Выгрузка прерывания
printer	Вывод строки на экран
loaded_check	Процедура проверяет загружено ли прерывание в память
load_int	Загрузка прерывания в память
Main	Процедура запускающая программу

Выполнение работы.

Данные объявленные в программе:

```
int_id dw 8f17h
keep_cs dw 0
keep_ip dw 0
is_loaded db 0
psp dw 0
unload_key db 0
key db '/un', '$'
int_not_loaded db 'Interruption is not loaded.', 10, 13, '$'
int_unloaded db 'Interruption unloaded successfully.', 10, 13, '$'
int_load_success db 'Interruption loaded successfully.', 10, 13, '$'
int_in_use db 'Interruption is loaded already.', 10, 13, '$'
cnt db 'interruption count: 0000' , '$'
keep_ss dw ?
keep_sp dw ?
keep_ax dw ?
int_stack dw 32 dup (0)
int_stack_end dw ?
```

При запуске программа проверяет наличие ключа выгрузки и, если он есть, то запускается процедура выгрузки пользовательского прерывания из памяти.

Далее процедура проверяет загружено ли в память прерывание, если нет выводится сообщение, если да, то выводится сообщение об успешной выгрузке и выгружается прерывание.

Если же флага нет, то проверятся загружено ли в память прерывание, если да, то выводится соответствующее сообщение, иначе прерывание загружается в память.

Тестирование программы:

```
E:\OS\LB4>lab4
Interruption loaded successfully.      interruption count: 00252
E:\OS\LB4>
```

Рис. 1 - Загрузка прерывания.

```
E:\OS\LB4>lab4
Interruption loaded successfully.      interruption count: 01967
E:\OS\LB4>lab4
Interruption is loaded already.
E:\OS\LB4>
```

Рис. 2 - Попытка загрузки прерывания, когда оно уже загружено.

```
E:\OS\LB4>lab4 /un
Interruption unloaded successfully.
E:\OS\LB4>
```

Рис. 3 - Выгрузка прерывания из памяти

```
E:\OS\LB4>lab4 /un
Interruption unloaded successfully.
E:\OS\LB4>lab4 /un
Interruption is not loaded.
E:\OS\LB4>
```

Рис. 4 - Попытка выгрузки прерывания, когда оно уже выгружено

Контрольные вопросы

Как реализован механизм прерывания от часов?

Принимается сигнал прерывания происходит раз в 54 мс, запоминаются регистры, по номеру источника в таблице векторов и определяется смещение, запоминается адрес 2 байта в IP и 2 байта в CS.

После выполняется прерывание по сохранённому адресу и далее восстанавливается информация прерванного процесса и управление возвращается прерванной программе.

Какого типа прерывания использовались в работе?

1. Int 10h – BIOS video service
2. Int 21h – DOS functions service
3. Пользовательское прерывание с вектором 1Ch int 21h

Выводы.

Было написано прерывание и изучен механизм выгрузки его в память.