Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей Кафедра информатики

Дисциплина: Архитектура вычислительных систем

**ОТЧЕТ**

к лабораторной работе № 1

на тему «Защищенный режим 32-разрядных процессоров»

Выполнил:

студент гр. 153503

Киселёва Е.А.

Проверил:

Калиновская А.А.

Минск 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc146631498)

[2 Краткие теоретические сведения 4](#_Toc146631499)

[3 Результаты выполнения лабораторной работы 5](#_Toc146631500)

[Выводы 6](#_Toc146631501)

[Список использованных источников 7](#_Toc146631502)

[Приложение А (обязательное) Листинг кода 8](#_Toc146631503)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является изучение особенности защищенного режима процессора, получение практических навыков по программированию переключения процессора из реального в защищенный режим и обратно.

## 2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

32-разрядные процессоры Intel 80386, 80486 и Pentium с точки зрения рассматриваемых в данном разделе вопросов имеют аналогичные средства, поэтому для краткости в тексте используется термин "процессор i386", хотя вся информация этого раздела в равной степени относится к трем моделям процессоров фирмы Intel.

Процессор i386 имеет два режима работы - реальный (real mode, R-Mode) и защищенный (protected mode, P-Mode).

Реальный режим – это режим, в котором процессор работает как быстрый процессор 8086, но позволяет пользоваться большинством своих технологий (ММХ / SSE / SSE2, 32-разрядными регистрами общего назначения, регистрами управления и отладки и пр.). После аппаратного сброса процессор переходит в этот режим и начинает выполнять программную инициализацию из BIOS-a. Реальный режим в современных процессорах предназначен для запуска компьютера и подразумевается, что операционная система будет работать в защищённом режиме (поэтому оптимизация по производительности для процессоров IA-32 производится для защищённого режима).

В реальном режиме не доступны основные достоинства процессора – виртуальная память, мультизадачность, уровни привилегий, работа с кэшами, буферами TLB, буфером ветвлений и некоторыми другими технологиями, обеспечивающими высокую производительность.

Как утверждает Intel, это "родной" (native) режим 32-разрядного процессора. В защищённый режим процессор надо переводить специальными операциями над системными регистрами и войти в этот режим процессор может только из реального режима. При работе в защищённом режиме процессор контролирует практически все действия программ и позволяет разделить операционную систему, драйвера и прикладные программы разными уровнями привилегий. Благодаря этому ОС может ограничить области памяти, выделяемой программам, запретить или разрешить ввод/вывод по любым адресам, управлять прерываниями и многое другое. При попытке программы выйти за допустимый диапазон адресов памяти, выделенной ей, либо при обращении к "запрещённым" для неё портам процессор будет генерировать исключения – специальный тип прерываний. Грамотно оперируя исключениями, операционная система может контролировать действия программ, организовать систему виртуальной памяти, мультизадачность и другие программные технологии.

## 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ     РАБОТЫ

В рамках выполнения лабораторной работы было создана приложение, которое осуществляет переключение процессора между защищенным режимом и реальным режимом. Пользователь может ввести время нахождения в защищенном режиме (Рисунок 1).



Рисунок 1 – Введение времени в защищенном режиме

После введения пользователем времени появляется основное окно программы, работающее в защищенном режиме указанное пользователем время (Рисунок 2).

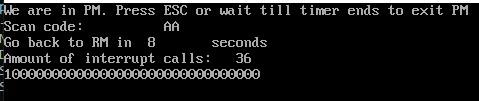


Рисунок 2 – Результат работы программы

По истечению таймера программа сообщает, что она снова работает в обычном режиме (Рисунок 3).



Рисунок 3 – Завершение работы таймера

## ВЫВОДЫ

В результате выполнения этой лабораторной работы была разработана программа с использованием DOSBox и Tasm C/C++, которая выполняет переключение процессора между защищенным режимом и реальным режимом. При этом программа отображает сообщение на экране и возвращает процессор в реальный режим после истечения времени, установленного пользователем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Юрий А. Уровни защиты. – СПб: Питер, 2000. – 158 с.
2. [Электронный ресурс]. – Режим доступа:  [https://dfe.petrsu.ru/x86/env\_1.shtml – Дата доступа 08.10.2023](https://learn.microsoft.com/ru-ru/windows/win32/procthread/process-and-thread-functions%20–%20Дата%20доступа%2008.10.2023)

**ПРИЛОЖЕНИЕ А**

## (обязательное)

## Листинг кода

**1.cpp**

TIMER\_HANDLER proc near ;Обработчик прерываний системного таймера

push ds

push es

pushad ;Занести в стек расширенные регистры общего назначения

mov ax,DATA\_DESC ;Переинициализировать сегментные регистры

mov ds,ax

inc ds:[COUNT] ;Увеличить значение счетчика

lea edi,ds:[BUFFER\_COUNT]

mov ax,ds:[COUNT]

call WORD\_TO\_DEC ;Преобразовать счётчик в строку

mov edi,538

lea esi,BUFFER\_COUNT

call BUFFER\_OUTPUT ;Вывести значение счетчика

SHOW\_TIMER:

mov al,0h ;Выбрать регистр секунд cmos

out 70h,al

in al,71h ;Прочитать значение секунд

cmp al,ds:[SECOND] ;Если секунда та же самая

je SKIP\_SECOND ;То пропустить вывод

mov ds:[SECOND],al ;Иначе записать значение новой секунды

mov al,ds:[TIME] ;Получить значение оставшегося время

cmp ds:[TIME],0 ;Если время подошло к концу

je DISABLE\_PM ;То на выход из защищенного режима

xor ah,ah

lea edi,ds:[BUFFER\_TIME]

call WORD\_TO\_DEC ;Преобразовать его в строку

mov edi,356

lea esi,BUFFER\_TIME

call BUFFER\_OUTPUT ;Вывести значение оставшегося время

dec ds:[TIME] ;Уменьшить значение оставшегося времени

lea esi,BUFFER\_TIME

call BUFFER\_CLEAR ;Очистка буфера

jmp SKIP\_SECOND ;На выход из обработки время

DISABLE\_PM: ;Выход из защищенного режима

mov al,20h

out 20h,al

db 0eah ;Дальний jmp

dd OFFSET EXIT\_FROM\_INTERRUPT ;На метку

dw CODE\_PM\_DESC ;В сегменте

SKIP\_SECOND: ;Секунда та же, не надо производить никаких действий

mov al,20h

out 20h,al ;Отпарвка сигнала контроллеру прерываний

popad

pop es

pop ds

iretd