Министерство образования Республики Беларусь

Учреждение Образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №8

по дисциплине «Программирование на языке ассемблера»

на тему «Обработчик прерываний. Резидентные программы»

вариант №1

|  |  |  |
| --- | --- | --- |
| Выполнил  студент гр. 250501  Лукьянов Е.О. |  | Проверил  Туровец Н.О. |

Минск 2023

**Цель работы:** Получить понятие об обработчиках прерываний и изучить особенности создания резидентных программ.

**Теоретические сведения**

Для выполнения работы требуется рассмотреть следующие элементы языка ассемблера и операционной системы:

1. Обработчики прерываний.

Для обработки событий, особенно происходящих асинхронно по отношению к выполнению программы, лучше всего подходит механизм прерываний – особых событий в системе, которые требуют немедленной обработки.

С каждым прерыванием связано уникальное число – номер прерывания, соответствующий определенному событию, а также соответствующий обработчик этого события. Для организации связи адреса обработчика прерывания с номером прерывания используется таблица векторов прерываний – блок памяти в диапазоне адресов от 0000:0000 до 0000:03FFh, состоящий из 256 элементов – дальних адресов обработчиков прерываний. В первом слове элемента таблицы записано смещение, а во втором – сегмент адреса обработчика прерывания. Вектор прерывания с номером 0 находится по адресу 0000:0000, с номером 1 – по адресу 0000:0004 и т. д. Инициализация таблицы векторов прерываний выполняется сначала BIOS перед началом загрузки операционной системы, а затем DOS (см. таблицу 9).

Аппаратные прерывания (обозначаются IRQ0-IRQ15) вырабатываются устройствами компьютера, как правило, при завершении ими операций обмена данными или при изменении состояния и обрабатываются программируемым контроллером прерываний.

Программы могут сами вызывать прерывания с помощью команды INT – программные прерывания. Программные прерывания удобно использовать для организации доступа к отдельным, общим для всех программ функциям.

1. Резидентные программы.

Резидентной программой называют программу, остающуюся в памяти после того, как она вернула управление операционной системе, и выполняющую обработку данных при обращении к ней.

Резидентную программу можно разделить на две части: инсталлятор и обработчик.

Инсталлятор обычно выполняет следующие действия:

-- проверяет можно ли установить обработчик;

-- устанавливает обработчик на определенный вектор прерывания; -- завершается, оставляя обработчик резидентным.

При повторном запуске инсталлятора, он может быть также использован для выгрузки резидентной части из памяти с восстановлением старых обработчиков.

Для установки вектора обработчика рекомендуется использовать следующие функции DOS:

-- Функция DOS 25h (INT 21h) – установить адрес обработчика прерывания:

Ввод: АН = 25h,

AL = номер прерывания,

DS = сегментный адрес обработчика,

DX = смещение обработчика в сегменте.

-- Функция DOS 35h (INT 21h) – получить адрес обработчика прерывания:

Ввод: АН = 35h,

AL = номер прерывания.

Вывод: ES = сегментный адрес обработчика прерывания,

BX = смещение обработчика прерывания.

-- Функция DOS 31h (INT 21h) – оставить программу резидентной:

Ввод: АН = 31h,

AL = код возврата,

DX = размер резидента в 16-байтных параграфах (больше 06h), считая от начала PSP.

Обработчик резидентной программы – процедура, выполняющая необходимые действия. Эта процедура обычно заменяет стандартный обработчик программного или аппаратного прерывания.

Принято разделять резидентные программы на активные и пассивные, в зависимости от того, перехватывают ли они прерывания от внешних устройств или получают управление, только если программа специально вызовет команду INT с нужным номером прерывания и параметрами. Кроме этого, обработчик прерывания может быть заменен и обычной (нерезидентной) программой для поддержки специфичных действий (по окончанию такой программы должен быть восстановлен старый обработчик прерывания).

При вызове команды INT выполняются следующие действия:

1) В стеке сохраняются регистры FLAGS, CS, IP; сбрасываются флаги IF и TF.

2) В регистр CS из таблицы векторов прерываний заносится значение сегмента обработчика прерывания, а в регистр IP – смещение обработчика прерывания.

3) Выполняется передача управления на обработчик программного прерывания (CS:IP).

При вызове команды IRET выполняются следующие действия:

1) Из стека восстанавливаются регистры IP, CS, FLAGS.

2) Передается управление в прерванную программу, на команду, находящуюся непосредственно за командой программного прерывания INT.

Для сохранения возможностей стандартного обработчика можно воспользоваться следующими вариантами формирования своего обработчика (адрес старого обработчика должен быть сохранен до установки нового обработчика):

-- вызов стандартного обработчика в начале функции обработчика (ниже приведенные команды эквивалентны команде INT):

pushf

call old\_handler

-- вызов стандартного обработчика в конце функции обработчика:

jmp cs:old\_handler

При этом образуются цепочки обработчиков прерываний, когда один вектор прерывания имеет несколько обработчиков, вызываемых друг за другом.

Для обработчиков аппаратных прерываний рекомендуется обязательно вызывать стандартный обработчик, т.к. кроме возврата из аппаратного прерывания требуется выполнить ряд действий с аппаратурой, в том числе и с контроллером прерываний. Ниже приведенный фрагмент кода предназначен для указания контроллеру прерываний, что обработка аппаратного прерывания завершена:

MOV AL,20h

OUT 20h,AL

Во время работы обработчика прерываний, может произойти повторный вызов этого же прерывания, при этом возможны ошибки обработки. Чтобы защитить обработчик от таких ситуаций (или сделать так чтобы он являлся повторно входимым), можно временно запретить прерывания в критических участках кода:

cli ; запретить прерывания

mov al,byte ptr counter

cmp al,100

jb counter\_ok

sub al,100

mov byte ptr counter,al

counter\_ok:

sti ; разрешить прерывания

Следует помнить, что, пока прерывания запрещены, система не отслеживает изменения часов реального времени, не получает данных с клавиатуры, так что прерывания надо обязательно, при первой возможности, разрешать.

В DOS обработчик INT 21h не является повторно входимым. В отличие от прерываний BIOS, обработчики которых используют стек прерванной программы, обработчик системных функций DOS записывает в SS:SP адрес дна одного из трех внутренних стеков DOS. Если функция была прервана аппаратным прерыванием, обработчик которого вызвал другую функцию DOS, она будет пользоваться тем же стеком, затирая все, что туда поместила прерванная функция. Когда управление вернется в прерванную функцию, в стеке окажется мусор и произойдет ошибка. Лучший выход – вообще не использовать прерывания DOS из обработчиков прерываний, но если от этого нельзя отказаться, то нужно принять необходимые меры предосторожности.

Если прерывание произошло в тот момент, когда не выполнялось никаких системных функций DOS, ими можно пользоваться. Чтобы определить, занята DOS или нет, надо сначала, до установки собственных обработчиков, определить адрес флага занятости DOS:

-- Функция DOS 34h (INT 21h) – определить адрес флага занятости DOS:

Ввод: АН = 34h

Вывод: ES:BX = адрес однобайтного флага занятости DOS,

ES:BX – 1 = адрес однобайтного флага критической ошибки DOS.

Теперь обработчик прерывания может проверять состояние этих флагов и, если оба флага равны нулю, разрешается свободно пользоваться функциями DOS.

Если флаг критической ошибки не нуль, никакими функциями DOS пользоваться нельзя.

Если флаг занятости DOS не нуль, можно пользоваться только функциями 01h – 0Ch, а чтобы воспользоваться какой-нибудь другой функцией, придется отложить действия до тех пор, пока DOS не освободится. Чтобы это сделать, надо сохранить номер функции и параметры в каких-нибудь переменных в памяти и установить обработчик прерывания 08h или 1Ch. Этот обработчик будет при каждом вызове проверять флаги занятости и, если DOS освободилась, вызовет функцию с номером и параметрами, оставленными в переменных в памяти. Кроме того, участок программы после проверки флага занятости – критический, и прерывания на этом участке должны быть запрещены.

Не все функции DOS завершаются быстро – функция чтения символа с клавиатуры может оставаться в таком состоянии до нажатия клавиши, и все это время флаг занятости DOS будет установлен в 1. В DOS предусмотрена и такая ситуация. Все функции ввода символов с ожиданием вызывают INT 28h в том же цикле, в котором они опрашивают клавиатуру, так что, если установить обработчик прерывания 28h, из него можно вызывать все функции DOS, кроме 01h – 0Ch.

Функции BIOS также часто оказываются не повторно входимыми (к ним следует отнести обработчики программных прерываний 05, 08, 09, 0Bh, 0Ch, 0Dh, 0Eh, 10h, 13h, 14h, 16h, 17h). Так как нет флага занятости BIOS, можно создать его самим:

int10\_handler proc far

inc cs:byte ptr int10\_busy ; увеличить флаг занятости

pushf ; передать управление старому

; обработчику INT 10h,

call cs:dword ptr old\_int10 ; эмулируя команду INT,

dec cs:byte ptr int10\_busy ; уменьшить флаг занятости

iret

int10\_busy db 0

int10\_handler endp

Теперь обработчики аппаратных прерываний могут пользоваться командой INT 10h, только тогда когда флаг занятости int10\_busy равен нулю, и это не приведет к ошибкам, если не найдется чужой обработчик прерывания, который тоже будет обращаться к INT 10h и не будет ничего знать о нашем флаге занятости.

**Код программы (.exe)**

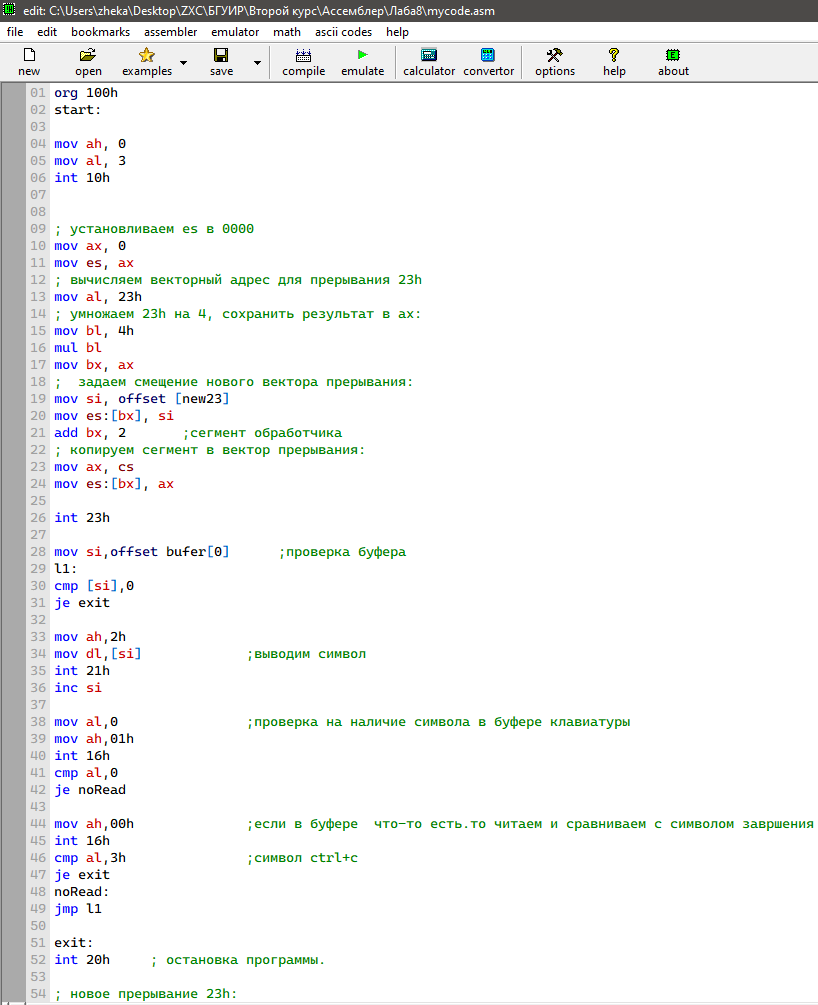
****

Рисунок 1 – Код программы

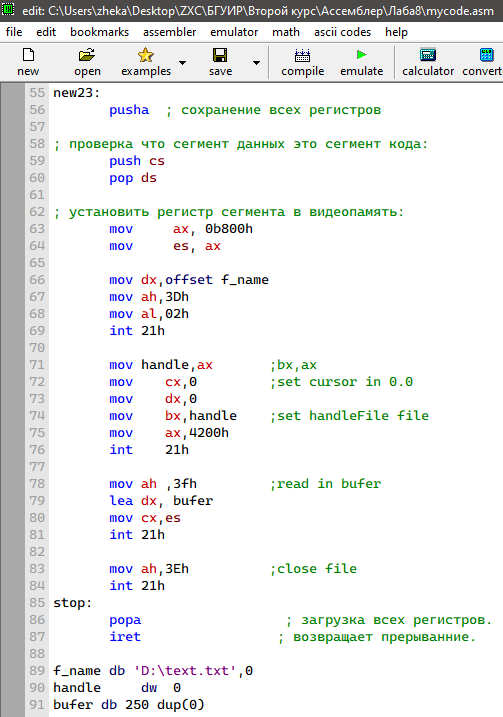


Рисунок 2 – Код программы

**Вывод программы**

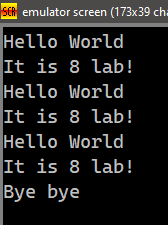
****

Рисунок 3 – Результат работы программы