

Лабораторная работа № 7

Использование программных прерываний при работе с ассемблером

Цель настоящей работы - обучение использованию программных прерываний при программировании на языке ассемблера в операционной системе MS DOS, а также изучение различных способов работы с клавиатурой и экраном.

ОСНОВНЫЕ СВЕДЕНИЯ

Программные прерывания и подпрограммы обработки прерываний.

Операционная система MS-DOS предоставляет программисту, работающему на языке ассемблера, большой набор подпрограмм, выполняющих различные полезные действия. Все эти подпрограммы оформлены как подпрограммы обработки прерываний, и для каждой подпрограммы в руководстве по MS-DOS указан номер соответствующего ей прерывания. Различают аппаратные и программные прерывания. Программа обработки аппаратного прерывания вызывается, если процессор получает сигнал запроса прерывания от одного из системных устройств. Эта подпрограмма должна содержать, кроме собственно команд, предназначенных для выполнения некоторых действий по обработке прерывания, определенных программистом, специальные команды, управляющие контроллерами прерываний. В связи с этим, подпрограммы-обработчики аппаратных прерываний редко непосредственно используются программистами. Подпрограмма обработки программного прерывания вызывается по команде INT n процессора (n - номер прерывания в системной таблице векторов прерываний). Сокращенно принято сами подпрограммы обработки прерываний называть "прерываниями", хотя это не правильно. Все программные прерывания подразделяются на прерывания BIOS (подсистемы управления вводом/выводом), прерывания ОС и прерывания пользователя. Подпрограммы обработки прерываний BIOS обычно находятся в ПЗУ, они поставляются разработчиками аппаратуры для обеспечения единого интерфейса ОС с различными типами оборудования. Основу системы прерываний ОС MS DOS составляет прерывание с номером 21h, называемое прерыванием MS DOS. Это прерывание обеспечивает API прикладных программ (Application Programm Interface - интерфейс прикладных программ) в среде ОС MS DOS. В зависимости от значения, содержащегося при вызове прерывания в регистре AH, MS-DOS выполняет при этом одну из нескольких десятков подпрограмм, которые принято называть функциями DOS. Для каждого прерывания и каждой функции DOS в руководстве описан набор входных и выходных параметров, передаваемых через регистры, а также перечень возможных ошибок.

В данной работе будут описаны прерывания и функции DOS и BIOS, относящиеся к работе с клавиатурой и экраном.

Программирование клавиатуры

От контроллера на клавиатуру поступают команды управления клавиатурой, а из клавиатуры коды подтверждения и коды сканирования.

За исключением клавиши "PAUSE" все остальные клавиши 101-клавишной клавиатуры работают на нажатие/отжатие. При нажатии генерируется код нажатия, а при удержании клавиши он повторяется через определенные интервалы времени, а

при отпускании генерируется код отжатия. Код нажатия и код отжатия вместе называются кодами сканирования клавиатуры (scan-коды). Код отжатия является двухбайтовым: первый байт равен F0h , а второй байт совпадает с кодом нажатия.

Имеется три уровня программирования подсистемы клавиатуры:

1) на физическом уровне (программирование портов 60h и 64h контроллера клавиатуры);

2) с помощью BIOS (INT 09h , INT 16h);

3) с помощью прерывания DOS (INT 21h).

Для выполнения ввода данных с клавиатуры программисту в большинстве случаев достаточно использовать описанные ниже функции DOS. В более сложных случаях используется прерывание BIOS INT 16h, дающее некоторые дополнительные возможности. Максимум возможностей программист может получить, перехватывая аппаратное прерывание от клавиатуры 09h и работая непосредственно с портами ввода/вывода.

Использование BIOS для работы с клавиатурой

Прерывание INT 09h является аппаратным прерыванием, которое вызывается по линии запроса прерывания IRQ1 всякий раз, когда заполняется выходной буфер контроллера клавиатуры, т.е. когда поступает байт от клавиатуры. Обработчик данного прерывания считывает scan-код из порта 60h, преобразует scan-код в код доступный BIOS или выполняет специальную функцию (вызов некоторых других прерываний, например INT 05h после нажатия Prt Scr и др.). Обработчик прерывания помещает для большинства клавиш соответствующий им в BIOS двухбайтовый код в специальную область памяти - буфер клавиатуры). Коды некоторых клавиш обработчик не помещает в буфер, информация о их нажатии или отжатии записывается в байты состояния клавиатуры.

Рассмотрим подробнее способ формирования двухбайтового кода, помещаемого INT 09h в буфер клавиатуры. Scan-код, получаемый обработчиком прерывания с клавиатуры, соответствует некоторой клавише или комбинации клавиш. BIOS использует для кодирования информации стандарт кодирования ASCII , применяемый для всех устройств, использующих посимвольный обмен. Наряду с обычными символами, применяются управляющие символы (например , CR - возврат каретки , LF - переход на следующую строку и т. п.). Каждому ASCII-коду BIOS ставит в соответствие некоторый scan-код, принимаемый с клавиатуры. Если нажата клавиша или комбинация клавиш, которой соответствует ASCII-код, обработчик прерывания INT 09h помещает в буфер scan-код клавиши и соответствующий ASCII-код. Однако имеется значительное количество комбинаций клавиш (и соответственно scan-кодов), не имеющих соответствующих ASCII эквивалентов. К ним относятся клавиши управления курсором, функциональные клавиши и др. Для подобных клавиш в буфер помещается специальный байт расширенного кода, а второй байт равен нулю. Для ряда клавиш ("серые" клавиши управления курсором , Delete , End и др.), продублированных на 101-клавишной клавиатуре, и их комбинаций с другими клавишами, используется расширенный код их "оригиналов", а второй байт для таких кодов равен E0h.

При нажатии комбинации клавиш Ctrl-Break прерывание INT 09h устанавливает признак запроса по Ctrl-Break (байт 471h) и вызывает прерывание BIOS INT 1Bh , которое обычно перехватывается MS DOS. Это прерывание устанавливает флаг Ctrl-C , после чего управление передается INT 23h , которое проверяет этот флаг, и в случае его установки вызывает досрочное прекращение

выполнения программы. Некоторые функции MS DOS проверяют нажатие Ctrl-C и вызывают INT 23h. Таким образом, Ctrl-Break является флагом BIOS, а Ctrl-C - флагом MS DOS. Эти флаги проверяются при выполнении функций 01h-0Ch прерывания MS DOS. Функция 33h прерывания MS DOS позволяет распространить контроль по Ctrl-Break на все функции ввода/вывода.

Прерывание BIOS INT 16h позволяет работать с клавиатурой на несколько более высоком уровне, чем INT 09h, и более низком уровне, чем функции MS DOS. Фактически все описанные ниже функции DOS используют в своей работе прерывание 16h, однако программист может использовать его и непосредственно, если в этом есть необходимость.

В зависимости от значения в регистре AH прерывание INT 16h может выполнить одну из следующих функций.

Функция 00h выполняет чтение символа из буфера клавиатуры. Если буфер пуст, функция ожидает нажатия клавиши. Если клавиша нажата, значение кода заносится в регистр AX. Функция возвращает в AL ASCII-код, а в AH - scan-код если это возможно. В противном случае в AH возвращается расширенный код, а AL равен нулю. Функция не обрабатывает нажатие клавиш, код которых не помещается в буфер прерыванием INT 09h (Ctrl, Shift и т.п.), а также дополнительных клавиш 101-клавишной клавиатуры.

Функция 01h проверяет, есть ли символы в буфере клавиатуры. Если есть, то флаг нуля сбрасывается (ZF=0), если нет - флаг устанавливается (ZF=1). При наличии символа его код заносится в регистр AX, в том же виде, как для функции 00h, однако из буфера код не удаляется. Таким образом, если после этого еще раз вызвать ту же функцию, то она еще раз прочтет тот же символ.

Функция 02h записывает в регистр AL первый из четырех байтов состояния клавиатуры.

Функции 10h и 11h аналогичны функциям 00h и 01h, но позволяют обрабатывать также и дополнительные клавиши 101-клавишной клавиатуры.

Функция 12h выполняет действия, аналогичные 02h, но кроме того помещает в AH второй байт состояния клавиатуры.

Функция 03h позволяет установить задержку и частоту повторения для клавиатуры, а функция 05h - поместить двухбайтовый код в буфер клавиатуры.

Функции DOS для работы с клавиатурой

Функция 01h (т.е. прерывание 21h при значении AH=01h) выполняет ввод с клавиатуры с ожиданием нажатия клавиши, если буфер клавиатуры пуст. Код введенного символа помещается в регистр AL. Введенный символ отображается на экране (эхо-отображение).

Функция 01h проверяет также, не нажимал ли пользователь в ходе работы программы комбинацию клавиш Ctrl/C (или Ctrl/Break). В этом случае управление передается на подпрограмму обработки Ctrl/C, которая обычно прекращает выполнение программы пользователя.

Если на клавиатуре была нажата одна из клавиш, которым не соответствует никакой код ASCII, то функция 01h возвращает в регистре AL значение 0. В этих

случаях следует еще раз вызвать ту же функцию, тогда будет выдан расширенный код данной клавиши или комбинации клавиш.

Функция 08h работает аналогично 01h, за исключением того, что не выполняется эхо-отображение введенного символа.

Функция 07h работает аналогично 01h, за исключением того, что не выполняется эхо-отображение и не проверяется нажатие Ctrl/C.

Функция 06h может выполнять как ввод с клавиатуры, так и вывод на экран. Если в момент вызова регистр DL содержит значение 0ffh, то данная функция выполняет ввод без ожидания. Если буфер не пуст, то флаг нуля ZF сбрасывается в 0, а код символа из буфера заносится в AL. Если же буфер пуст, то устанавливается ZF=1, при этом значение в AL не играет роли. Таким образом, эта функция не ждет, пока будет нажата клавиша, а сразу выдает какой-то результат. Функция 06h не выполняет эхо-отображения и не проверяет нажатие Ctrl/C.

Функция 0Bh не выполняет ввод символа, а только проверяет, есть ли символы в буфере. Если есть, то устанавливается AL=0ffh, если нет, то AL=00h. Выполняется также проверка на Ctrl/C.

Функция 0Ah выполняет буферизованный ввод строки с клавиатуры. При этом символы вводятся один за другим, как при многократном применении функции 01h, до тех пор, пока не будет введен код 0dh (код клавиши "Enter"), завершающий строку. В ходе ввода строки пользователь может редактировать строку, и в частности - использовать "забой". При вводе выполняется также проверка на Ctrl/C.

При вызове функции 0Ah требуется, чтобы в регистре DX содержал смещение (в сегменте данных) области памяти (буфера), в которую система поместит введенную строку. В первом байте этого буфера должна быть записана его длина, т.е. максимальное число символов (включая 0dh), которое можно записать в буфер. Эта длина должна быть по крайней мере на 2 меньше, чем число зарезервированных байт. После окончания ввода строки функция помещает во второй байт буфера действительное число введенных символов (не считая 0dh), а начиная с третьего байта буфера размещаются введенные символы. Последним всегда будет код 0dh.

Функция 0Ch вначале очищает буфер клавиатуры (т.е. "забывает" предыдущие нажатия клавиш), а потом выполняет любую из функций 01h, 06h, 07h, 08h или 0Ah. Номер этой функции задается в регистре AL. Если задано иное число, то выполняется только очистка буфера. Кроме того, всегда выполняется проверка на Ctrl/C.

Для вывода на экран пользователь также имеет выбор: использовать простые функции DOS или более сложные, но и богатые по возможностям прерывания BIOS. Кроме того, имеется возможность выводить данные на экран, просто записывая их по адресам видеопамати ЭВМ.

Использование BIOS для работы с экраном

Прерывание 10h, обслуживаемое BIOS, предоставляет программисту гораздо более богатые возможности работы с экраном, чем функции DOS. Значение, занесенное в регистр AH, позволяет выбрать одну из функций, выполняемых этим прерыванием. В данной работе рассмотрены только некоторые из этих функций. Следует также отметить, что производители видеооборудования обычно поставляют

расширения BIOS для работы с видеосистемой. Поэтому на конкретной системе могут присутствовать различные дополнительные функции.

Функция 00h переключает режим работы видеоадаптера. Код нового режима должен содержаться в регистре AL.

Повторным вызовом функции 00h часто пользуются для быстрой очистки экрана.

Может использоваться текстовый режим 02h, ниже приводится описание основных функций для текстовых режимов. Каждому символу, отображаемому на экране, соответствует в текстовом режиме два байта в видеопамяти. По младшему байту (байту с четным адресом) располагается код символа, а по старшему байту - код атрибута (см. далее). В текстовых режимах начало видеобуфера располагается по адресу B800:0000h, в зависимости от объема видеопамяти видеобуфер может содержать несколько текстовых страниц. Только одна, активная в данный момент страница, отображается на экране. Программа имеет возможность отображать на экране одну страницу, и в то же время изменять информацию в других страницах. Коды символов, отображаемых на экране в текстовом режиме, хранятся в специальном формате в виде таблиц шрифтов и используются знакогенератором видеоадаптера. Пользователь имеет возможность использовать собственные таблицы шрифтов (прерывание 1Fh). Символы с кодами 0 - 127 соответствуют ASCII-кодам, при этом многие управляющие символы не выводятся на экран, а выполняют соответствующие управляющие функции. Начертание символов с кодами 128-256 зависит от используемой ДОС кодовой страницы, которая поддерживает соответствующий национальный алфавит.

Функция 0Fh возвращает в AL номер текущего видеорежима, в AH - число столбцов символов на экране, в BH - номер активной кодовой страницы.

Функция 01h задает форму курсора в текстовом режиме. Биты 0-4 регистра CH задают верхнюю строку развертки курсора в символьной матрице 8x8, биты 0-4 регистра CL - нижнюю строку развертки. Строки развертки нумеруются сверху вниз, начиная с нуля. Если бит 5 CH установить в 1, то курсор будет не виден. Биты 6 и 7 должны содержать нули.

Функция 02h позволяет перемещать курсор в заданную точку экрана. Номер строки (от 0 до 24, считая сверху вниз) задается в регистре DH, а номер позиции в строке (от 0 до 79, слева направо) - в регистре DL. Регистр BH должен содержать номер видеостраницы, в данной работе его следует задавать равным 0.

Функция 03h позволяет прочитать текущую позицию курсора. Перед вызовом регистр BH должен содержать номер видеостраницы. После вызова в регистре DH будет номер строки, а в DL - номер позиции курсора. CH и CL будут содержать верхнюю и нижнюю строки развертки курсора.

Функция 05h переключает активную страницу, AL должен содержать номер страницы.

Функция 06h выполняет сдвиг содержимого заданной прямоугольной области экрана на одну или несколько строк вверх (скролинг). При этом в регистрах должны содержаться следующие параметры: CH и CL - номер строки и номер позиции верхнего левого угла области; DH и DL - номер строки и номер позиции нижнего правого угла; AL - число строк для скролинга; BH - атрибуты для заполнения освободившихся строк. Эта функция также часто используется для очистки экрана.

или его части. В этом случае надо указать число строк сдвига не меньше, чем число строк в области.

Функция 07h аналогична 06h, но скроллирует область вниз.

Функция 08h позволяет прочитать символ, находящийся на экране в позиции курсора. Перед вызовом ВН должен содержать активную страницу. После вызова регистр AL содержит ASCII-код символа, а AH содержит байт атрибутов символа (только в текстовом режиме).

Атрибуты символа задают цвет символа и цвет фона, на котором он виден. Биты этого байта имеют следующее значение:

бит 7= 1/0 - символ мигает / не мигает;
 биты 6-4 - цвет фона (0 - черный, 7 - белый, остальные значения - разные цвета);
 бит 3= 1/0 - символ имеет повышенную/нормальную яркость;
 биты 2-0 - цвет символа.

Функция 09h выдает указанный символ с заданными атрибутами, начиная с текущей позиции курсора. Перед вызовом ВН должен указывать активную страницу, AL должен содержать ASCII-код выдаваемого символа, BL - его атрибуты, CX - число повторений символа (обычно 1). После вызова текущая позиция курсора не изменяется. Для управляющих символов выводятся их графические изображения.

Функция 0Ah отличается от 09h только тем, что атрибуты символа не задаются, а остаются такими же, какие раньше имел символ в текущей позиции.

Функция 0Eh просто выводит символ из AL в текущую позицию, остальные регистры значения не имеют. При этом курсор смещается на 1 позицию. Управляющие символы обрабатываются как управляющие.

Функции DOS для вывода на экран

Имеются три функции для вывода на экран: две функции для вывода одного символа, а третья - для вывода строки символов.

Функция 02h выдает в текущую позицию экрана символ, код которого содержится в регистре DL. Известно, что для машин типа IBM PC AT почти каждому из 256 возможных кодов соответствует какое-то графическое изображение, однако при выводе по функции 02h некоторые коды не выдаются на экран, а служат управляющими. В частности:

0dh (CR) - перевод курсора в начало текущей строки;
 0ah (LF) - перевод курсора вниз на 1 строку;
 08h (BS) - перевод курсора влево на 1 позицию;
 07h (BEL) - звонок.

Функция контролирует Ctrl-C.

Функция 06h при DL не равном 0ffh выполняет действия , аналогичные предыдущей , но без проверки на Ctrl-C.

Функция 09h выдает начиная с текущей позиции экрана строку символов, смещение которой (в сегменте данных) содержится в регистре DX. Строка может

содержать управляющие символы (CR, LF и т.п.), и таким образом на экране занимать несколько строк. Концом выдаваемой строки служит символ "\$" (код 24h).

В центре экрана расположен "шарик". При нажатии любой из 4 клавиш со стрелками он начинает двигаться, оставляя за собой на экране след. Если нажимается другая стрелка - направление движения изменяется. Если шарик достигает края, он останавливается. При нажатии пробела цвет оставляемого следа меняется на противоположный.

Сначала выбираются функции для работы с клавиатурой и экраном. Поскольку требуется, чтобы при отсутствии нажатий программа продолжала перемещать шарик в ранее заданном направлении, необходимо использовать ввод без ожидания. Это можно выполнить с помощью функции DOS 06h. Что же касается вывода на экран, то функций DOS здесь будет недостаточно, так как нужно управлять перемещением курсора. Следует использовать прерывание BIOS 10h. Будут нужны следующие функции этого прерывания: 02h - для перемещения курсора, 09h - для выдачи символа и 06h - для очистки экрана. Программа использует текущий видеорежим.

Байты состояния клавиатуры

1	й Байт 417h:	2	й Байт 418h:
бит7	1:режим Ins включен	бит7	1:клавиша Ins нажата
бит6	1:режим Caps Lock включен	бит6	1:клавиша Caps Lock нажата
бит5	1:режим Num Lock включен	бит5	1:клавиша Num Lock нажата
бит4	1:режим Scroll Lock включен	бит4	1:клавиша Scroll Lock нажат
бит3	1:клавиша Alt (любая)нажата	бит3	1:режим "Пауза" установлен
бит2	1:клавиша Ctrl(любая)нажата	бит2	1:клавиша SysReq нажата
бит1	1:левый Shift нажат	бит1	1:левый Alt нажат
бит0	1:правый Shift нажат	бит0	1:левый Ctrl нажат
3	й Байт 496h:	4	й Байт 497h:
бит7	1:прочитать идентификатор	бит7=1	ошибка при передаче
бит6	1:код ABh	бит6=1	модификация индикаторов
бит5	1:включить инд. Num Lock	бит5=1	повторение
бит4	1:признак 101-кл.клавиатуры	бит4=1	получено подтверждение
бит3	1:правый Alt нажат 1:правый	бит3=0	резерв
бит2	Ctrl нажат 1:код E0h	бит2	:состояние инд. CapsLock
бит1	1:код E1h	бит1	:состояние инд. NumLock
бит0		бит0	:состояние инд. ScrollLock

Ячейки памяти , используемые для прерываний от клавиатуры

Ячейка памяти	Байт	Функция	
0000:0024	4	Вектор INT 09h	
0000:0058	4	Вектор INT 16h	
0000:0066	4	Вектор INT 1Bh	
0040:0017	2	Байты состояния клавиатуры (1 и 2)	
0040:0019	1	Рабочий байт для ввода кода по клавише Alt	
0040:001A	2	Указатель начала буфера(на текущий символ)	
0040:001C	2	Указатель конца буфера(на последний символ)	
0040:001E	32	Буфер клавиатуры (16 символов)	
0040:0071	1	Признак запроса по Ctrl	Break
0040:0080	2	Смещение начала буфера (001Eh)	
0040:0082	2	Смещение конца буфера	1 (003Eh)
0040:0096	2	Дополнительные байты состояния клавиатуры	

* - если текущие указатели на начало и конец буфера равны , то буфер пуст

Коды ASCII (0-127)

МЛ. ПОЛУ БАЙТ	СТАРШИЙ ПОЛУБАЙТ КОДА							
	0	1	2	3	4	5	6	7
0	NUL	DLE	!	0	@	P	`	p
1	SOH	DC1	"	1	A	Q	a	q
2	STX	DC2	#	2	B	R	b	r
3	ETX	DC3	\$	3	C	S	c	s
4	EOT	DC4	%	4	D	T	d	t
5	ENQ	NAK	&	5	E	U	e	u
6	ACK	SYN	'	6	F	V	f	v
7	BEL	ETB	(	7	G	W	g	w
8	BS	CAN	)	8	H	X	h	x
9	HT	EM	*	9	I	Y	i	y
A	LF	SUB	+	:	J	Z	j	z
B	VT	ESC	,	;	K	[	k	
C	FF	FS	-	<	L	\	l	
D	CR	GS	.	=	M	]	m	
E	SO	RS	/	>	N	^	n	~
F	SI	US		?	O	_	o	DEL

Таблица расширенных кодов некоторых комбинаций клавиш

КЛАВИШИ И КОМБИНАЦИИ КЛАВИШ	КОДЫ
Ctrl/2	03
Shift/Tab	0F
Alt/Q,W,E,R,T,Y,U,I,O,P	10 - 19
Alt/A,S,D,F,G,H,J,K,L	1E - 26
Alt/Z,X,C,V,B,N,M	2C - 32
F1 - F10	3B - 44
Home	47
Стрелка вверх	48
PageUp	49
Стрелка влево	4B
Стрелка вправо	4D
End	4F
Стрелка вниз	50
PageDown	51
Ins	52
Del	53
Shift/F1 - Shift/F10	54 - 5D
Ctrl/F1 - Ctrl/F10	5E - 67
Alt/F1 - Alt/F10	68 - 71
Ctrl/PrintScreen	72
Ctrl/Стрелка влево	73
Ctrl/Стрелка вправо	74
Ctrl/End	75
Ctrl/PageDown	76
Ctrl/Home	77
Alt/1,2,3,4,5,6,7,8,9,0,-,=	78 - 83
Ctrl/PageUp	84

Функции прерывания INT 10h

АН	Действие
00h	Установить режим видео
01h	Установить тип курсора
02h	Установить позицию курсора
03h	Прочитать позицию курсора
04h	Прочитать позицию светового пера
05h	Выбрать активную страницу дисплея
06h	Прокрутить вверх активную страницу
07h	Прокрутить вниз активную страницу
08h	Прочитать атрибут/символ
09h	Записать атрибут/символ
0Ah	Записать только символ
0Bh	Установить палитру
0Ch	Записать точку
0Dh	Прочитать точку
0Eh	Записать TTY
0Fh	Прочитать текущее состояние видео