Подпрограммы, прерывания и сервисы

Подпрограмма

Подпрограмма - это отдельная записанная последовательность инструкций, которую можно вызывать из разных точек программы. Механизм реализации подпрограммы выполняется с помощью инструкций CALL и RET. Подпрограммы могут быть NEAR (в том же сегменте, что и вызывающая программа / внутрисегмент) или FAR (в другом сегменте / дополнительном сегменте). При вызове подпрограммы (через CALL) адрес, по которому должен быть выполнен возврат, сохраняется в стеке, а при возврате из подпрограммы (через RET) адрес восстанавливается из стека и регистра IP (возможно, также CS) перезагружается. Механизм вызова подпрограммы показан на рисунке.

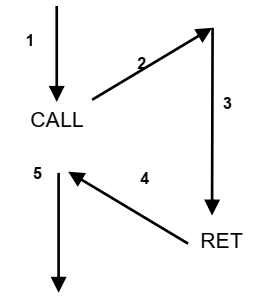
1. основная программа

2. вызов подпрограммы

3. выполнение подпрограммы

4. возврат из подпрограммы

5. возобновить основную программу



Механизм вызова подпрограммы

Внутрисегментные вызовы сохраняют только смещение адреса возврата, а при RET это значение перезагружается в IP. Межсегментные вызовы также сохраняют содержимое регистра CS и содержимое регистра IP, так что при возврате оба будут восстановлены. Для двух случаев инструкции CALL и RET имеют разные коды.

У инструкции RET нет операндов; оператор CALL имеет единственный операнд. Режимы адресации идентичны режимам JMP, за исключением относительной адресации. Если подпрограмма изменяет (использует) регистры, значение которых все еще требуется в программе ppellant, эти

регистры должны быть сохранены в стеке с помощью PUSH, а затем перестроены с помощью POP перед возвратом из подпрограммы.

Прерывания

Прерывание - это сигнал, передаваемый в компьютерную систему, который объявляет о наступлении события, требующего внимания.

Когда событие произошло, выполняются следующие действия по порядку:

- приостановка текущей программы; сохранение адреса возврата в стеке (IP, CS);

- запуск специальной процедуры, называемой программой обработки прерывания, которая обслуживает прерывание;

- возобновление выполнения приостановленной программы путем восстановления адреса возврата из стека.

Причины этих событий могут быть двух типов: внутренние и внешние. Прерывание считается только между выполнением двух последовательных машинных инструкций. Если два прерывания происходят одновременно, аппаратные схемы компьютерной системы решают, какое из них будет обслуживаться в первую очередь.

В случае прерывания расчетная система должна в таком порядке:

- определить тип события, вызвавшего прерывание (внутреннее, внешнее),

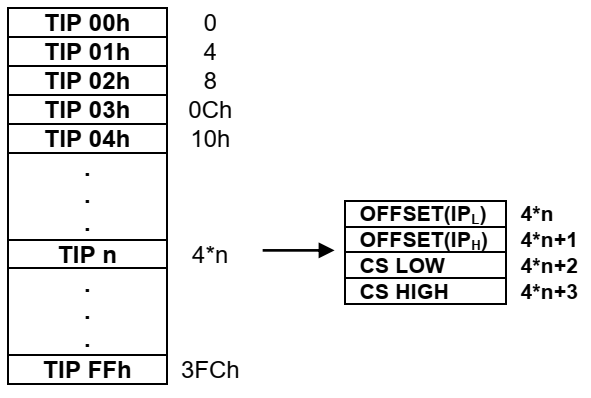
- выяснить причину прерывания,

- определить адрес процедуры обработки прерывания (RTI).

RTI строится для каждого типа события и для каждой возможной причины. Метод, используемый для быстрой локализации RTI, - это векторизация прерываний. Это включает в себя привязку фиксированной адресной ячейки памяти для каждого сбоя. В этом месте введите адрес RTI, соответствующий прерыванию. Микропроцессор 8086 имеет 256 прерываний, пронумерованных от 00h до FFh. Их векторизация производится следующим образом: в начале ОЗУ зарезервировано 256

дублюкувинте; каждое двойное слово содержит FAR-адрес RTI. Первое двойное слово, расположенное по адресу 0000: 0000, содержит адрес RTI для прерывания 00h; второй, расположенный по адресу 0000: 0004, содержит адрес RTI для прерывания 01h и т. д. Для прерывания k удвойте слово, которое

содержит адрес RTI находится по адресу 0000: 4 \* k



Прерывания делятся на hard и soft. hard прерывание 00h исходит от микропроцессора (внутреннего) и возникает при попытке деления на ноль; прерывание 08h исходит от схемы отсчета времени и используется для отсчета времени; Прерывание 02h сигнализирует об ошибке четности при доступе к области памяти и является немаскируемым прерыванием, то есть его триггер не может управляться флагом прерывания.

Программные прерывания обеспечивают доступ к службам BIOS и DOS; широко используются благодаря предлагаемым возможностям, программной базе, которую можно использовать как библиотеку готовых программ (подпрограмм), что значительно облегчает работу программиста на языке ассемблера. Эти процедуры называются услугами. Будут проиллюстрированы примеры услуг для некоторых прерываний, интенсивно используемых при написании программ.

Механизм обработки прерывания аналогичен подпрограммам:

- при запуске прерывания, независимо от его типа, текущее состояние микропроцессора сохраняется в стеке; сохраняется в стеке и PSW;

другие прерывания отключены;

- микропроцессор идентифицирует адрес, по которому находится подпрограмма обработки прерывания; Для этого номер, связанный с прерыванием (тип прерывания), используется как индекс в массиве векторов прерываний;

- адрес подпрограммы из соответствующей позиции таблицы векторов массива загружается в CS и IP; процедура обработки прерывания выполняется до тех пор, пока не будет выполнена инструкция IRET;

- он восстанавливается после прерывания путем перезагрузки IP, CS и PSW со значениями, сохраненными в стеке вызовов.

Специфичаские инструкцый для прерываний

Мягкое прерывание может быть вызвано оператором INT с синтаксисом INT n, вызывая активацию обработчика, соответствующего прерыванию с номером n. Он выполняет четыре последовательных действия:

- флаги стека (PSW);

- стек обратного FAR-адреса (CS, IP);

- поставить 0 во флаги TF и ​​IF;

- вызывает по косвенному адресу обработчик, связанный с прерыванием.

Инструкции STI и CLI воздействуют на флаг прерывания IF, сообщая процессору, как вести себя при возникновении прерывания. После CLI (Clear Interrupt, IF = 0) процессор больше не поддерживает прерывания. Обычно он появляется в начале обработчика, чтобы не мешать его работе.

STI (Set Interrupt) позволяет процессору принимать прерывания, IF = 1.

OBS. Немаскируемые прерывания не учитывают состояние флага IF!

Инструкция IRET вызывает возврат прерывания. Это последняя инструкция, выполняемая в обработчике, имеющая дополнительный эффект к инструкции INT:

- восстанавливает флаги из стека;

- возвращается к инструкции, чей FAR-адрес находится наверху стека

|  |  |  |
| --- | --- | --- |
| **INT N** SP ← Sp-2 | **IRET** IP ← [SP] |  |
| [SP] ← PSW | SP ← SP+2 |  |
| I ← 0 | CS ← [SP] |  |
| SP ← SP-2  [SP] ← CS  SP ← SP-2 | SP ← SP+2 PSW ← [SP] SP ← SP+2 |  |
| [SP] ← IP IP ← [4\*N+1 4\*N] CS ← [4\*N+3 4\*N+2] |  |  |

Прерывания и сервисы BIOS и DOS

BIOS отвечает за управление оборудованием ввода / вывода. Так что в BIOS записана целая куча подпрограмм, относящихся к этому оборудованию. Их вызов в приложении осуществляется прерываниями; Внутри каждого прерывания может выполняться несколько сервисов, выбор нужной услуги

выполняется путем загрузки в регистр AH номера службы перед вызовом прерывания. Параметры вызова службы при необходимости загружаются в определенные регистры.

Некоторые прерывания BIOS показаны ниже:

INT 10 h - экран / видео услуги

INT 13 h - дисковые сервисы

INT 14 h - услуги последовательной связи

INT 16 h - услуги клавиатуры

INT 20 ч - конец программы

Видео сервисы - INT 10 ч

Функция 00 - установка видеорежима

AH = 00

AL = код видеорежима (см. Приложение 3)

Функция 02 - установка позиции курсора

AH = 02

BH = номер страницы видео (0 для графического режима)

DH = строка

DL = столбец

Функция 09 - записать символ в курсор (не перемещая курсор)

AH = 09

AL = код ASCII записывающего символа

BH = страница видео

BL = атрибут цвета

Функция 0ch - записать графический пиксель в координатах

AH = 0ch

AL = цвет

BH = страница видео

CX = столбец

DX = ряд

0eh функция - запись символа в режиме телетайпа (с перемещением курсора)

AH = 0eh

AL = ASCII-код записывающего символа

BH = страница видео

Услуги клавиатуры - INT 16h

Функция 00 - ждет нажатия клавиши и считывает набранный символ

AH = 00

AL = (вернул) код ASCII набранного символа

Функция 01 - считывает состояние клавиатуры

AH = 01

Z = 0 - была нажата клавиша; Z = 1 - ни одна клавиша не нажата

Функции DOS

Основное прерывание DOS - 21:00. Задачи некоторых из упомянутых выше прерываний были взяты на себя и иногда расширены некоторыми функциями этого прерывания.

Функция 01 - чтение символа с клавиатуры

AH = 01

AL = (возвращенный) прочитанный символ

Функция 09 - запись строки, оканчивающейся на «$»

AH = 09

DS: DX - указатель на строку, оканчивающуюся на «$»

Функция 4ch - завершить процесс с кодом возврата

AH = 4 канала

AL = код возврата метода завершения программы; не поддерживается версиями

Подраздел DOS 2.x.

Перенаправить прерывание

Процедура обработки определенного прерывания может быть переписана пользователем с соблюдением определенных правил, связанных, прежде всего, со структурой соответствующей процедуры, а затем с манипуляциями с адресами этих программ. Чтобы заменить RTI подпрограммой, написанной пользователем, он будет заменен по адресу, соответствующему подпрограмме в TVI, на адрес новой подпрограммы. В качестве естественной меры предосторожности человек, который изменяет адрес обработчика, должен сохранить старый адрес и восстановить его, когда он больше не хочет использовать свой собственный обработчик.

Функция 35 прерывания 21h позволяет считывать адрес RTI в TVI для определенного прерывания, желаемого пользователем. Введите код функции (35h) в AH и тип прерывания в AL. После вызова INT 21h в регистрах ES: BX получается FAR-адрес обработчика.

Функция 25h прерывания 21h позволяет изменить адрес RTI в TVI для определенного желаемого прерывания. В AH введите код функции (25h), в AL введите желаемый тип прерывания, а в DS: DX введите FAR-адрес нового обработчика.