

ПРЯМОЙ ДОСТУП К РАСШИРЕННОЙ ПАМЯТИ ИЗ РЕАЛЬНОГО РЕЖИМА ДЛЯ 32-РАЗРЯДНЫХ ПРОЦЕССОРОВ

Тарасенко А.Н., Пискунов Я.М.

Кафедра ЭВМ
taras@cs.dgtu.donetsk.ua

Abstract

Alexandr N. Tarasenko, Yaroslav M. Piskunov. Direct extended memory access from real mode of 32-bit microprocessors. The method of access to large data array and extended memory is proposed. The method is easily realizable and allows essentially to increase speed of access to large data array in specialized computers.

Введение

При разработке программного обеспечения для спецЭВМ на базе микропроцессоров семейства x86 иногда возникает необходимость обрабатывать массивы данных большого размера, либо обслуживать адресное пространство более 1Мб. В первом случае можно использовать дальние указатели и при вычислении смещения элементов в массиве производить нормализацию указателя. Во втором случае необходимо использование драйвера расширенной памяти. На IBM-совместимых машинах для этого используется драйвер HIMEM.SYS. Но если архитектура ЭВМ будет отличаться от архитектуры IBM-совместимых машин, его применение невозможно. Поэтому необходима разработка собственного драйвера.

Все вышеперечисленные проблемы можно решить, используя защищенный режим. Но при его использовании возникает необходимость работать с системными структурами данных, что значительно усложняет программы и затрудняет их отладку.

Всех этих недостатков можно избежать, используя flat-сегмент (под flat-сегментом следует понимать сегмент, имеющий произвольный базовый адрес и размер до 4Гб). Выполнив инициализацию сегментного регистра параметрами такого сегмента, можно адресовать любую ячейку памяти, используя лишь ближний 32-разрядный указатель, что значительно увеличивает быстродействие.

Техническое обоснование

Для доступа к расширенной памяти драйвер HIMEM.SYS использует недокументированную команду LoadAll [1]. Эта команда производит загрузку всех регистров процессора из определенной области памяти. При этом один сегментный регистр загружается параметрами flat-сегмента. Таким образом, драйвер производит доступ к расширенной памяти в реальном режиме. Динамику этого процесса можно понять, рассмотрев механизм загрузки сегментного регистра и формирование линейного исполнительного адреса операнда.

Сегментный регистр состоит из двух основных частей: селекторной части и дескрипторной части. Селекторная часть имеет 16 двоичных разрядов и программно доступна. Дескрипторная часть программно недоступна и содержит собственно

описание сегмента: базовый адрес - 32 разряда, размер - 20 разрядов и атрибуты доступа [2-3]. При инициализации сегментного регистра в защищенном режиме в селекторную часть загружается селектор сегмента, а в дескрипторную часть - дескриптор сегмента, соответствующий этому селектору. В реальном режиме в селекторную часть записывается сегментный адрес, базовый адрес сегмента вычисляется путем сдвига сегментного адреса на 4 разряда влево и записывается в дескрипторную часть, размер сегмента устанавливается равным 64Кб. Исполнительный адрес вычисляется путем суммирования смещения ячейки памяти и базового адреса сегмента из дескрипторной части [1-3],[www.intel.com]. Команда LoadAll позволяет загрузить дескрипторную часть независимо от текущего режима и, следовательно, получить flat-сегмент.

Но есть еще более быстрый и простой способ. Можно временно переключиться в защищенный режим, загрузить сегментный регистр и выйти в реальный режим. Этот временный вход в защищенный режим требует наличия только таблицы глобальных дескрипторов, состоящей из 2-х дескрипторов (по 8 байт), включая один нулевой. При этом дальний переход не выполняется, так как нет необходимости перезагружать регистр CS, в дескрипторной части которого находятся допустимые значения базового адреса, размера и атрибутов. В этом случае конвейер процессора не сбрасывается, что обеспечивает минимальное время загрузки сегмента. Если выделить какой-нибудь сегментный регистр только для доступа к flat-сегменту (рекомендуется fs или gs), то его инициализацию можно производить только один раз при старте программы.

Реализация

На основании вышесказанного предлагается следующий порядок действий:

- создание дескриптора flat-сегмента с базовым адресом 0 и размером до 4Гб в таблице глобальных дескрипторов;
- загрузка регистра GDTR;
- запрещение аппаратных прерываний;
- установка бита режима в регистре CR0;
- загрузка сегментного регистра flat-сегментом;
- сброс бита режима в регистре CR0;
- разрешение аппаратных прерываний.

Данный порядок действий рекомендуется для всех процессоров семейства x86. В процессорах Pentium и Pentium-Pro для повышения быстродействия следует учитывать возможности распараллеливания команд. Изложенная идея экспериментально проверена при оценке эффективности.

Оценка эффективности

Для оценки эффективности необходимо произвести какую-нибудь обработку большого массива различными методами доступа. В данном примере применяется следующая методика, Массивы различного размера заполняются случайными числами, упорядоченными в порядке возрастания. Затем производится двоичный поиск случайного числа в массиве методом приведения дальнего указателя и методом flat-сегмента. Параметром оценки является количество выполненных поисков в единицу времени (в данном случае между двумя прерываниями системного таймера, 18.2 Гц).

Измерение производится 10 раз, и в качестве результата берется усредненное значение. Процесс повторяется для массивов различного размера. Ниже приведены результаты теста, выполненные на компьютере IBM на базе процессора Intel Pentium-200MMX, оснащенный памятью SDRAM с временем доступа 10 нс.

Таблица 1. Результаты тестирования:

| Размер массива, Кб | Метод приведения указателя, кол-во поисков | Метод доступа через flat-сегмент, кол-во поисков | Коэффициент увеличения быстродействия |
|--------------------|--|--|---------------------------------------|
| 64 | 19 341 | 28 944 | 1.4965 |
| 128 | 18 037 | 27 347 | 1.5162 |
| 192 | 16 794 | 25 867 | 1.5403 |
| 256 | 15 778 | 24 563 | 1.5568 |

Как видно из таблицы, при увеличении размера массива эффективность второго метода возрастает.

При необходимости работать с массивами еще большего размера эффективность второго метода значительно возрастет, так как в противном случае придется применять драйвер расширенной памяти, который будет пересылать участки массива в основную память.

Выводы

В данной статье был показан метод доступа к массивам большого размера с использованием flat-сегмента. На основании проведенных экспериментов можно утверждать, что данный метод является простым в реализации и обладает наилучшим быстродействием в сравнении с другими методами доступа в реальном режиме.

Литература

1. А.В.Фролов, Г.В.Фролов. "Защищенный режим процессоров Intel 80286/80386/80486" Москва. "ДИАЛОГ-МИФИ". 1993г. 240с.
2. В.Б.Бродин, И.И.Шагурин. "Микропроцессор i486. Архитектура, программирование, интерфейс". Москва. "ДИАЛОГ-МИФИ". 1993г. 240с.
3. К.Айден, Х.Фибельман, М.Крамер. Аппаратные средства РС: Пер. с нем. - СПб.: BHV- Санкт-Петербург, 1996 - 544 с., ил.