**Программные измерительные мониторы.**

Программные измерительные мониторы (ПИМ) - это совокупность команд или программ, выполняемых исключительно с целью проведения измерений. Обычно ПИМ – это специальные программные средства, под управлением которых производится выполнение программы на той же ЭВМ, на которой измеряемая программа и должна выполняться. При этом ПИМ собирает данные о ходе выполнения программ и накапливает их в памяти.

То, что эти команды ПИМ должна выполнять сама измеряемая система, приводит к возникновению искажений. Количество искажений зависит от частоты обнаруживаемых событий и от операций, выполняемых измерителем при обнаружении каждого события.

Идея использовать измерительные средства в процессе разработки программ - не нова. В настоящее время существует несколько типов таких программных средств. Особое место среди них занимают профилировщики. Профилировщики (называемые также анализаторами процесса выполнения программ) - это программы, позволяющие получить ряд количественных данных о процессе выполнения объекта разработки (снять профиль разрабатываемой программы) и на основании этих данных выявить в программе "узкие места", отрицательно сказывающиеся на эффективности ее работы. Профиль программы может содержать, например, следующую информацию о процессе выполнения программы:

- как и на что расходуется время работы программы;

- сколько раз выполняется данная строка программы;

- сколько раз и какими модулями вызывается данный модуль программы.

Один из самых известных профилировщиков - Turbo Profiler фирмы Borland International. Система Turbo Profiler контролирует каждый шаг выполнения программы и предоставляет подробные статистические сведения о всех этапах ее работы. Такой подход к измерениям позволяет проводить профилирование в интерактивном режиме и получать большое количество сведений о работе программы, но вносит большие временные искажения в процесс ее выполнения. Кроме того, для проведения измерительного эксперимента тестируемая программа должна быть откомпилирована с отладочной информации при отключении оптимизации компилятора. Это делает профилирование неадекватным реальному выполнению программы. Для получения достоверной информации о времени выполнения от программиста требуется достаточно высокая квалификация в методике планирования и проведения измерительного эксперимента. Таким образом, Turbo Profiler ориентирован прежде всего на профессиональных программистов. За счет грамотного планирования эксперимента и отказа от ряда сервисных функций можно повысить точность временных измерений до 55 мс. Тем не менее, такая точность в ряде случаев недостаточна.

Другой подход реализован, например, в пакете Interval Performance Monitoring (IPM) Калифорнийского Университета. IPM ориентирован исключительно на временные измерения и состоит из набора отдельных функций для замера времени и протоколирования результатов измерения. Функции вставляются в нужные места программы, после чего программа компилируется. Такой подход позволяет резко повысить точность измерений, но требует значительных дополнительных усилий по сбору и классификации полученной информации. Трудоемкость проведения измерительного эксперимента не позволяет считать системы типа IPM полноценными профилировщиками.

Многие из полезных принципов организации системы Turbo Profiler и пакета IPM были реализованы на кафедре МОЭВМ в ПИМ Sampler. Особенностью работы ПИМ Sampler являются:

1. Исследуемая программа запускается под управлением монитора и монитор берет на себя все функции по накоплению результатов измерений и протоколированию.

2. Исследуемая программа выполняется "естественным" образом, а не по шагам и без использования отладочной информации.

3. Измерения времени выполняются с максимально возможной точностью.

4. Искажения, вносимые ПИМ за счет потребления им ресурсов ЭВМ, минимальны и не сказываются на итоговых результатах.

5. По результатам измерений формируется файл отчета печатного вида и представление их в графической форме.

На этапе планирования измерительного эксперимента производится задание контрольных точек. После этого исследуемая программа компилируется. Задание контрольной точки предполагает вызов некоторой функции на языке программирования исследуемой программы. Назовем эту функцию функцией задания контрольных точек или просто функцией контрольных точек. Эта функция не должна вносить больших искажений в работу исследуемой программы. Для этого она должна быть небольшого размера и обладать высоким быстродействием. Вызов функции контрольных точек осуществляется с помощью механизма прерываний, в частности, используется прерывание пользователя с номером 65. Функция контрольных точек выполняет измерение времени дважды: сразу после входа в нее и непосредственно перед выходом. Оба значения времени передаются измерительному монитору. После первого замера управление передается монитору и он выполняет действия по идентификации и накоплению результатов измерений. После возврата управления в функцию задания контрольных точек, выполняется второе измерение времени. Такой подход наиболее полно соответствует условиям 3 и 4, поскольку позволяет точно измерить не только время выполнения участка программы между контрольными точками, но и убрать временное искажение вносимое измерительным монитором.

Проведение измерительного эксперимента начинается с запуска монитора. Работа измерительного монитора состоит из нескольких фаз.

* Подготовительные операции. На этом этапе производится проверка допустимости входных параметров, формирование необходимых структур данных и операции по калибровке измерительного монитора. После окончания подготовительных операций управление передается исследуемой программе. Управление передается по схеме - "запуск программы-потомка с возвратом в предок".
* Идентификация и накопление измерительный информации. Эти действия производится после перехвата прерывания от исследуемой программы. Идентификация информации подразумевает не только анализ положения контрольных точек, но и определение ряда статистических сведений о проходах между контрольными точками. Накопление такого рода измерительной информации облегчает последующее формирование отчета и позволяет резко сократить расход памяти, потребляемой монитором. При накоплении измерительная информация корректируется для снятия временных искажений монитора. Таким образом, исследуемая программа выполняется без трассировки и освобождается от операций по анализу и протоколированию измерительной информации. Это полностью соответствует принципам 1, 2 и 4.
* Операции по анализу измерительной информации и формированию отчетов. Эта фаза является заключительной и наступает после окончания работы исследуемой программы. Реализация условия 5 требует значительных ресурсов ВС. В этой связи, целесообразно графическую интерпретацию снятого профиля выполнять в отдельной программе - просмотрщике. Снятый профиль программы выдается в виде текстового файла и в виде двоичного.

В «Sampler»e замеры системного времени проводятся, как и для IPM, внутри тестируемой программы вызовом функции задания точки профилирования. Этим достигается высокая точность измерения времени и отсутствие ограничений на использование отладочной информации и оптимизации компилятора. Функция задания точки профилирования определена в модуле на соответствующем языке программирования, подключаемом к тестируемой программе. С другой стороны, запуск тестируемой программы осуществляется, как и для Turbo Profiler, через монитор. Это позволяет освободить пользователя от сбора и классификации измерительной информации. Результаты измерения могут быть представлены как на экране компьютера, так и твердой копией после вывода протокола измерения на принтер.

ПИМ Sampler состоит из двух исполняемых модулей следующего функционального назначения:

* Измерительный монитор Sampler. Монитор запускает программу пользователя и перехватывает прерывания, несущие информацию о прохождении тестируемой программы пользователя через определяемые им точки (контрольные точки ). После отработки тестируемой программы, монитор выгружает собранную информацию в текстовый файл с профилем тестируемой программы и создает специальный двоичный файл для графической интерпретации результатов;
* Программа графической интерпретации результатов профилирования Smpview. Программа выдает снятый профиль на экран в максимально информативной форме. Такой формой является ориентированный граф с вершинами в контрольных точках снятия отсчетов. Программа анализирует схему переходов между контрольными точками и отображает ее в виде дуг графа. Информацию о времени перехода можно получить отметив нужные вершины на графе.

Обоснование методики измерения времени.

В ПИМ Sampler используются два режима измерения моментов времени, соответствующих выполнению фрагментов программы.

Режим 1: *Измерения времени через 0 канал таймера с использованием портов ввода-вывода.*

Для задания временных интервалов и формирования сигналов с различными временными параметрами в компьютерах IBM PC используется программируемый таймер 8253. Таймер работает независимо от процессора и считает реальное время. Формирование времени суток в компьютерах IBM PC можно пояснить схемой, показанной на рис.9.

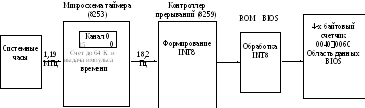
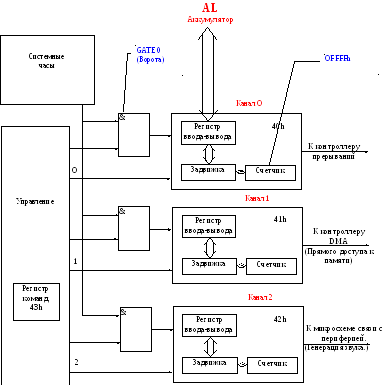
IRQ0

Рис.9

Микросхема таймера имеет три канала, основным из которых с точки зрения измерения времени, является канал 0. Импульсы частотой 1.19 МГц от системных часов поступают на микросхему таймера, где происходит деление частоты на 64 К и на выходе канала 0 мы получаем импульсы с частотой 18,2 Гц ( с периодом следования 55 мс). Эти импульсы поступают на контроллер прерываний в виде запроса аппаратного прерывания на уровне 0 и, если разрешены прерывания, вырабатывается сигнал **INT8.**В результате обработки прерывания INT8 накапливаются данные в двойном слове с адресом 0040:006Сh, расположенном в области данных BIOS.

Более подробно возможность измерения временных характеристик программ с использованием микросхемы таймера можно пояснить с помощью рис.10.

Рис.10

Микросхема таймера содержит три канала с адресами 40h, 41h, 42h. Каждый канал содержит три регистра, два из которых двухбайтные - задвижка и счетчик и один - однобайтный - регистр ввода-вывода. Регистр ввода-вывода связан шиной с аккумулятором (регистром AL) для считывания информации.

Канал 0 таймера выполняет две функции:

1. Осуществляет контроль времени суток;

2. Синхронизирует дисковые операции (запись и считывание информации с магнитных носителей).

Если изменять заносимые в счетчик данные (не 0FFFF, а другие числа), то можно увеличить частоту выдачи импульсов на выходе таймера, но при этом могут наблюдаться сбои при выполнении дисковых операций.

Канал 1 таймера обеспечивает прямой доступ к памяти (DMA).

Канал 2 таймера служит для связи с периферийными устройствами, обеспечения выдачи звуковых сигналов.

Управление работой таймера осуществляется регистром команд по адресу 43h.

Структура команд порта по адресу 43h

|  |  |  |  |
| --- | --- | --- | --- |
| Канал  7 – 6 | Операция  5 – 4 | Режим  3-1 | 0 |

0 - бит указывает какая система счисления используется - двоичная или BCD - или двоично-кодированная десятичная.

Биты 1 - 3 задают режим работы таймера:

* 000 - запись данных счетчика в задвижку;
* 011 - выполнение основных операций по считыванию информации канала 0 и подсчета поступивших импульсов.

Биты 4 - 5 определяют возможные операции:

* 00 - передача данных из счетчика в задвижку или наоборот;
* 01 - чтение или запись старшего байта;
* 10 - чтение или запись младшего байта;
* 11 - чтение или запись последовательно младшего и старшего байтов.

При снятии измерений регистр задвижки загружается кодом (0FFFFh), который потом перемещается в счетчик. Счетчик считает поступающие тактовые импульсы в обратном направлении (в убывающем порядке), поэтому при его загрузке и считывании надо брать дополнение до FFFFFh. После того как счетчик обнулится, вырабатывается сигнал контроллера прерывания INT-8. Содержимое счетчика следует поместить в задвижку и считать его в аккумулятор AL. После этого восстановить содержимое задвижки и продолжить работу. Интервал времени в 1/18.2 сек (55 мс) получил название «тик», а минимально различимый интервал, равный 1/FFFF «тика» получил название «минитик» = 0.84 мкс.

Наиболее точное и полное значение промежутков времени можно получить, используя совместно значение переменной области данных BIOS (0040:006С) и внутреннего счетчика таймера. Это позволяет измерять промежутки времени длительностью от 0,84 мсек до 24 часов.

Достоинства режима измерения времени через 0 канал таймера с использованием портов ввода-вывода:

- возможность применения на машинах типа IBM PC любого класса.

Недостатки:

- низкая точность измерений;

- большая погрешность функции контрольных точек, обусловленная медленным снятием отсчетов времени через порты ввода-вывода.

Режим 2. *Измерения времени через счетчик Time Stamp Counter (TSC).*

Режим применяется только для старших моделей IBM PC, начиная с Pentium. В составе этих компьютеров появился внутренний 64 битный таймер, который может быть прочитан в регистры EDX:EAX с помощью инструкции RDTSC (Read Time Stamp Counter). Каждый такт этот счетчик увеличивает свое значение на 1. Это очень полезно для замера точного количества тактов, потребовавшихся на исполнение некоторого фрагмента кода.

К сожалению, имеются проблемы чтения содержимого счетчика TSC. В Pentium добавлен внутренний регистр CR4, флаг TSD которого управляет возможностью чтения счетчика TSC. При CR4.TSD, равном 0, счетчик TSC можно читать в любом случае. При CR4.TSD, равном 1, содержимое счетчика можно прочитать только на 0-ом уровне привелегий. В общем случае флаг CR4.TSD установлен. Поэтому, инструкция RDTSC не может выполняться в режиме виртуального 8086 (3-й уровень привилегий).

При использовании EMM386 или Windows процессор находится в защищенном режиме и Dos - приложения типа Sampler запускаются в виртуальном режиме и имеют 3 уровень привелегий. Поэтому, если запускать измеряемую программу под DOS, необходимо закомментировать EMM386 (или любой другой менеджер памяти) в CONFIG.SYS и не следует запускать программу в режиме DOS из под Windows.

Достоинства данного режима измерений:

* малая погрешность функций контрольных точек;
* высокая точность измерений (до нескольких машинных тактов):
* для регистрации отсчета времени требуется только команда RDTSC и сохранение регистров EAX:EDX (20-30 тактов).

Недостатки:

* возможность применения только на машинах класса Pentium и выше.
* возможность полноценной работы только под DOS и в отсутствие драйверов памяти типа EMM386 (QEMM).

- презентация