

Б. Х. ЗИНГЕР
(Новосибирск)

ОБ ОДНОЙ ЗАМКНУТОЙ СИСТЕМЕ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ

Введение. В статье содержится краткое описание двух версий редактора библиотечных наборов данных, одна из которых реализована на ЭВМ ЕС-1020, а другая — на ЭВМ М-4030. Редактор предназначен для создания и редактирования библиотечных наборов данных, хранящихся в библиотеке исходных программ. Рассматривается форма эксплуатации ЭВМ с разделением ресурсов. Пользователь включен на линию с ЭВМ через терминальное устройство на базе алфавитно-цифрового дисплея «Видеотон-340». Такую организацию работы, когда несколько терминалов имеют непосредственную связь с ЭВМ, обычно называют режимом коллективного доступа [1].

Основной проблемой в реализации данного режима является организация защиты программных компонент от взаимного влияния пользователей как на уровне оперативной, так и на других уровнях внешней памяти. Разделение аппаратных (процессор, память, внешние устройства) и программных ресурсов системы также ставит сложные задачи управления.

Основные положения. В основу реализации редактора положен принцип, который заключается в поочередном использовании одних и тех же программных компонент всеми пользователями системы (инвариантных компонент [2]). Планирующая программа организует поочередное переключение редактирующей программы на обслуживание каждого пользователя. Переход от одного пользователя к другому происходит через достаточно короткие промежутки времени, что создает впечатление одновременности обработки, т. е. монополизации процессора каждым пользователем. Другими словами, реальный процессор как бы разделяется на несколько процессоров. Каждый оператор, связанный с таким имитируемым процессором (обычно называемым виртуальным), может судить о присутствии в системе аналогичных процессоров только по уменьшению фактического быстродействия [2]. В то время как один из таких виртуальных процессоров ожидает завершения операции ввода или вывода, другой может продолжать работу.

Для того чтобы программа разделилась между несколькими пользователями, т. е. была инвариантной, необходимо соблюдать следующие ограничения: программа должна содержать только такие инструкции, которые не модифицируются в процессе счета; любая ячейка памяти, содержимое которой изменяется в процессе обработки, должна включаться в рабочее поле каждого пользователя системы.

Таким образом, с каждым терминалом связано рабочее поле для хранения переменных, регистров и состояний реального процессора при переходе от одного виртуального процессора к другому.

Параллельная обработка. Рассмотрим условия прерывания текущей программы и выбор пользователя для передачи ему управления. Так как алфавитно-цифровые дисплеи подключены к ЭВМ в качестве активных внешних устройств, то информация, поступающая с клавиатуры, воспринимается в момент ее выработки. Но это не означает, что она сразу же начинает обрабатываться, так как существует иерархия временных ограничений, приводящих к тому, что в момент поступления информации адресуемый виртуальный процессор может находиться в пассивном состоянии, а реальный процессор занят. Для регистрации информации в момент ее передачи вычислитель прерывает исполнение текущей работы и передает управление подпрограмме обработки прерывания. По окончании подпрограммы обработки прерывания выполнение прерванной программы возобновляется с того места, где она была приостановлена. Обработка запросов пользователей осуществляется по принципу «первым пришел — первым обслуживается».

Каждый виртуальный процессор может находиться в четырех состояниях:

- 1) естественное прерывание (переход в состояние ожидания конца ввода-вывода);
- 2) активное (обработка запроса на обслуживание);
- 3) пассивное (ожидание запроса на обработку);
- 4) ожидание управления (состояние после получения запроса на обслуживание).

Планирующая программа анализирует состояние каждого виртуального процессора. Если статус последнего — «Ожидание управления», то инициируется его работа и он становится активным или текущим, иначе его состояние остается прежним. Независимо от планирующей программы состояние каждого виртуального процессора может изменяться следующим образом:

активное состояние → пассивное (в результате завершения обработки запроса);

активное состояние → естественное прерывание;
естественное прерывание → ожидание управления (после окончания операций ввода-вывода);
пассивное → ожидание управления (после получения запроса на обслуживание).

Следует отметить, что виртуальный процессор может перейти в активное состояние лишь в результате передачи управления от планирующей программы; при этом происходит восстановление состояния реального процессора и настройки базового регистра редактирующей программы на рабочее поле текущего пользователя.

Механизм защиты. Защита программных компонент от взаимного влияния виртуальных процессов основана на применении инвариантной редактирующей программы, так как она не может сама себя модифицировать, а ячейки памяти, изменяемые ею, разнесены по рабочим полям всех пользователей. Каждому рабочему полю соответствует свое значение базового регистра редактирующей программы. Когда базовый регистр редактирующей программы настроен на рабочее поле текущего пользователя, то доступ осуществляется лишь к этому рабочему полю. Поскольку виртуальным процессорам не разрешено изменять содержимое базового регистра, то гарантируется их взаимная изоляция.

Управление памятью. Функционально память можно разбить на три класса: рабочая, промежуточная, архивная.

А. Рабочая память (речь идет об оперативной памяти). В ней находится программа и рабочие поля каждого пользователя.

Б. Промежуточная память. Ограниченный объем рабочей памяти не позволяет хранить в ней совокупность наборов данных. Поэтому используется промежуточная память. В каждый момент времени лишь один виртуальный процессор может быть активным. При передаче управления от одного виртуального процессора к другому может возникнуть необходимость в обмене информацией между рабочей и промежуточной памятью. Так как такая ситуация возникает, как правило, часто, то промежуточная память должна быть достаточно быстрой, потому что именно характеристики промежуточной памяти фактически определяют производительность всей системы. Согласно этим соображениям, использованы метод прямого доступа к информации на магнитном диске и сегментная организация памяти на устройстве прямого доступа. Информация о начальных адресах и длинах сегментов организована в линейный список, для управления которым написаны макрокоманды обработки линейных списков [3].

В. Архивная память. Пользователи имеют возможность проводить запись массивов в архив. В нашем случае под архивом понимается стандартная библиотека исходных программ [4]. Для записи, считывания и удаления исходных программ специально написаны подпрограммы, реализующие библиотечный метод доступа.

ЛИТЕРАТУРА

1. Уилкс М. Системы с разделением времени. М., Мир, 1972.
2. Бергэн Ж., Риту М., Ружие Ж. Работа ЭВМ с разделением времени. М., Наука, 1972.
3. Кнут Д. Искусство программирования для ЭВМ. М., Мир, 1976.
4. Дисковая операционная система ДОС АСВТ. Т. VI. Служебные программы. Кн. I. М., ИЭУМ, 1974.

Поступило в редакцию 4 октября 1979 г.

УДК 621.372.001.24 : 681.3

В. В. ЕФИМЕНКО, Ю. А. СТУКАЛИН
(Новосибирск)

КОМПЛЕКС ПРОГРАММ ДЛЯ ЧИСЛЕННОГО АНАЛИЗА РАДИОТЕХНИЧЕСКИХ СХЕМ

При машинном проектировании нелинейных электронных схем важной задачей является анализ переходного и установившегося (стационарного) режимов нелинейных систем с периодическим входом. К настоящему времени создан ряд универсальных программных комплексов (см., например, [1]), успешно решающих задачу анализа нелинейных систем. Однако необходимо отметить, что использование