

мера, являющаяся простой модификацией статистики χ^2 для проверки гипотезы независимости в таблице сопряженности признаков 2×2 , составленной из чисел n_+ , n_- , $k - n_+$, $k - n_-$, где k — общее число реализаций. После вычисления указанной меры для всех окон определяются те из них, в которых достигаются достаточно выраженные локальные экстремумы.

Описанный метод может быть модифицирован в двух направлениях. Во-первых, вычисление меры для каждого временного окна может производиться не для всех, а для части последовательных реализаций, а поиск экстремумов — соответственно не только по параметрам временного окна, но и по параметрам, определяющим данную последовательность реализаций. Такая модификация в определенном смысле аналогична переходу от простых перистимульных гистограмм к динамическим. Во-вторых, описанный метод может быть модифицирован для решения задачи сопоставления реакций, возникающих при выполнении психологических тестов, в которых реакция испытуемого определяется смыслом предъявляемого стимула. Здесь возможны эффекты двух типов. В определенном временном окне может наблюдаться значимо большее число реализаций с компонентами положительной амплитуды при одном типе реакции, чем при другом, либо аналогичное преобладание для компонент с отрицательной амплитудой. Обе рассмотренные модификации также реализованы в виде программ описываемого пакета.

ЛИТЕРАТУРА

1. Гоголицын Ю. Л., Каминский Ю. Л., Кропотов Ю. Д., Пахомов С. В. Аппаратурно-программный комплекс для исследования динамики нейронной активности мозга человека. — Автотметрия, 1981, № 4, с. 69.
2. Бехтерева Н. П. Нейрофизиологические аспекты психической деятельности человека. — Л.: Медицина, 1974.
3. Бехтерева Н. П. Здоровый и больной мозг человека. — Л.: Наука, 1980.
4. Данько С. Г., Каминский Ю. Л. Система технических средств нейрофизиологических исследований мозга человека. — Л.: Наука, 1982.
5. Bechtereva N. P., Gogolitsin Yu. L., Plyukhina V. A., Pakhomov S. V. Dynamic neurophysiological correlates of mental processes. — Int. J. Psychophysiol., 1983, v. 1, p. 187.
6. Гоголицын Ю. Л., Пахомов С. В. Анализ изменений частоты разрядов нейронов мозга человека в ходе однократного выполнения психологических проб. — Физиология человека, 1984, т. 10, с. 796.
7. Гоголицын Ю. Л., Пахомов С. В. Нейрофизиологические возможности изучения вызванных изменений частоты разрядов нейронов методом разложения на компоненты. — Физиология человека, 1985, т. 11, с. 544.

Поступила в редакцию 19 ноября 1985 г.

УДК 681.3 : 612.014

Г. А. ЗАЙТМАН, Б. Я. ПЯТИГОРСКИЙ

(Киев)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ АППАРАТУРЫ СО ВСТРОЕННОЙ МИКРОЭВМ

Введение. Стоимость современных микроЭВМ делает возможным их использование в составе отдельных приборов для электрофизиологических исследований. Это позволяет существенно повысить сложность алгоритмов проведения эксперимента и первичной обработки результатов. Однако во многих случаях по ряду причин (экономических, надежных, габаритных) встроенные микроЭВМ лишены внешних запоми-

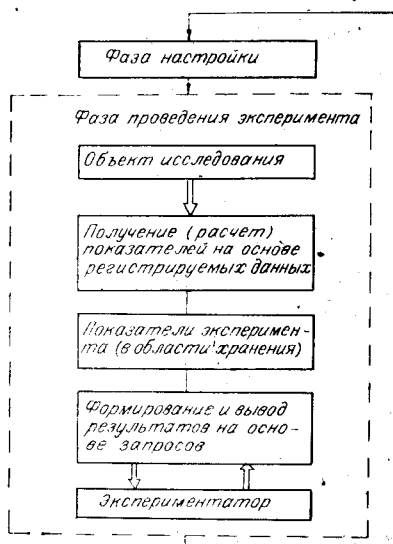


Рис. 1

программирование необходимо выполнять на машинно-ориентированных языках.

В настоящей статье описан один из вариантов разработки такого ПО, выполняющего функции специализированной исполнительной ОС РВ.

Принципы проведения электрофизиологических экспериментов с помощью автоматизированных приборов. Анализ типичных электрофизиологических экспериментов показывает, что технология их проведения, как правило, строится по определенной схеме. В соответствии с ней (рис. 1) работу с ЭВМ можно разбить на две фазы: настройки параметров и проведения эксперимента и обработки данных. Фаза настройки параметров, включающая в себя фиксацию конкретных диапазонов изменения параметров, выбор режимов работы оборудования и т. д., выполняется не в режиме реального времени и далее не рассматривается.

Процесс получения показателей эксперимента по ряду соображений удобно разбить на две части: регистрация данных, поступающих от объекта, и их обработка, т. е. получение необходимых показателей. Кратко перечислим эти соображения. Некоторые показатели вычисляются не на основе отдельных измерений, а на основе группы (массива) измерений, выполненных за определенный временной интервал. Примерами таких показателей являются перистимульные гистограммы, параметры кардиокомплекса и т. д. Назовем группу данных, необходимых для расчета того или иного показателя, элементом данных. Размер элемента данных может колебаться от нескольких до сотен байт. Еще одна причина разделения процессов сбора и обработки данных связана с тем, что моменты поступления данных и время их обработки могут быть случайными величинами. Это ведет к тому, что обработка одного элемента данных может оказаться незавершенной к моменту поступления следующего. Поэтому для согласования скоростей ввода данных и их обработки необходима буферизация элементов данных. Можно также отметить, что логическое разделение ввода данных и их обработки улучшает гибкость системы в целом.

Получение характеристик исследуемого процесса состоит из трех этапов: ввод команды экспериментатора (нажатие соответствующей функциональной клавиши), формирование соответствующих этой команде результатов, их вывод (дисплей, другие средства индикации).

Анализ вариантов реализации системы рационально осуществлять, используя средние значения интенсивности, S_i возникновения запросов

нающих устройств. Их отсутствие выдвигает два основных требования к разработчикам программного обеспечения: 1) необходимо осуществлять обработку экспериментальных данных по мере их поступления, так как объем оперативной памяти, как правило, недостаточен для предварительного накопления всего объема информации; 2) весь комплекс программ должен располагаться в оперативной памяти. Выполнение этих требований возможно только на основе учета специфики предметной области и эффективного использования ресурсов ЭВМ. Следствием такого подхода является, во-первых, отказ от применения универсальных резидентных операционных систем реального времени (ОС РВ), поскольку они занимают порядка 10 кбайт под собственные программы, а все программное обеспечение (ПО) в данном случае должно располагаться в 16—32 кбайт, во-вторых,

на запуск программных процессов ввода, обработки, представления, вывода, а также длительности t_i выполнения этих процессов [1]. Произведение $S_i * t_i$ определяет часть времени занятости ЭВМ на выполнение i -го процесса. Общий коэффициент загрузки ЭВМ $\sum S_i * t_i$. Как показывает опыт, средняя суммарная интенсивность запросов на выполнение программ ввода данных для решения широкого класса задач с использованием электрофизиологических методик находится в пределах 100—1000 с. Чем выше скорость ввода данных по некоторому каналу, тем больше обычно размеры соответствующих элементов данных, передаваемых программам расчета показателей. Поэтому суммарная средняя интенсивность запуска этих программ варьируется в небольших пределах (1—4 с). Верхняя граница средней интенсивности поступления запросов на вывод результатов определяется временем их восприятия экспериментатором и не превышает 1 с. В большинстве случаев она значительно ниже. В процессе работы системы требуется выдавать различные диагностические сообщения. Полезно предусмотреть единый механизм их выдачи. Интенсивность появления диагностик не превышает 0,1—0,3 с. Таким образом, можно выделить часто запускаемые процессы ввода данных и более редкие процессы обработки и преобразования.

Если при одновременной работе нескольких процессов процесс i вынужден ждать завершения процесса j , то к нему может образоваться очередь длиной в $S_i * t_j$ запросов. Время выполнения программ ввода данных t_i (на ЭВМ типа «Электроника 60») составляет единицы миллисекунд. В то же время для «редких» программ вполне допустимо время исполнения в сотни миллисекунд. Если программы ввода данных будут ожидать конца выполнения длинных программ обработки или преобразования, то к ним будут накапливаться слишком большие очереди запросов ($S_i * t_j = 10—100$ и более). Поэтому часто запускаемые программы должны обладать абсолютным приоритетом, т. е. прерывать выполнение других программ. При этом такие программы вполне могут ожидать конца выполнения друг друга, так как в этом случае $S_i * t_j < 10$.

Отказ от прерываний работы одним «редким» процессом другого, даже менее приоритетного, значительно упрощает функции управляющей программы, облегчает координацию работы системы в целом. Например, выполняющемуся в некоторый момент времени процессу гарантируется монопольный доступ к общим данным.

Действия процессов ввода данных должны ограничиваться заполнением буферов, являющихся входными для программ обработки. Поэтому прерывания быстро выполняющимися программами долго работающих никак не влияют на работу последних. Программы ввода друг друга не прерывают, а выполняются в порядке «первым пришел — первым обслуживается».

Структура проблемно-ориентированной системы реального времени. Функционирование программных систем во многих случаях естественно представлять в виде квазипараллельной работы нескольких программных процессов. Эти процессы могут взаимодействовать между собой на основе различных видов синхронизации, которые обеспечивают обмен сообщениями и согласование доступа к общим ресурсам, таким, как общие участки памяти, внешние устройства и т. д. Несмотря на принципиальную эквивалентность различных методов, применяемых для решения этих задач, использование того или иного из них для конкретной системы может оказаться эффективней. Как указано в [2], при фиксированном и относительно небольшом числе программных процессов, выполняемых в системе, предпочтительнее организация их взаимодействия с помощью сообщений, передачу которых удобно вести посредством портов или очередей [3]. Под очередью понимается упорядоченная совокупность сообщений некоторому процессу. Сообщения, предназначенные для передачи, направляются в его входную очередь и считываются в подходящий для процесса-получателя момент времени. Для синхронизации сообщений в случае фиксированного числа процессов в системе управ-

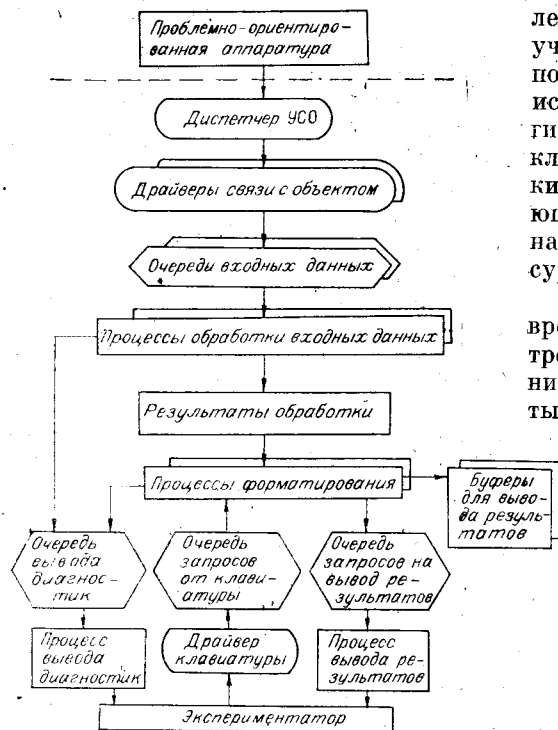


Рис. 2

удалить (первый элемент из очереди).

Отметим, что при выполнении операции **получить** элемент становится доступным, но из очереди не удаляется. Для его удаления нужно выполнить соответствующую команду. Такое разделение функций позволяет при получении данных из очереди не копировать их в локальные переменные, а проводить обработку непосредственно в буфере очереди. Операция **получить** может использоваться также для проверки пустоты очереди. Очереди определяются статически, т. е. изменение параметров очереди (максимальной длины и размера элементов), а также создание новой очереди в процессе работы системы невозможно.

Выделены следующие типы процессов (рис. 2):

Прикладные. Обработка поступающих от объекта данных, форматирования (подготовка результатов для вывода экспериментатору). Эти процессы работают с общими переменными, процессы обработки имеют приоритет перед процессами форматирования.

Служебные. Вывод результатов и диагностических сообщений на терминал. Общим ресурсом для этих процессов является экран терминала, процесс диагностик имеет приоритет перед процессом результатов.

Управляющий. Этот процесс выполняет функции монитора и согласовывает порядок доступа процессов к общим ресурсам.

Процессы ввода данных через устройство связи с объектом или с клавиатуры терминала (драйверы). При работе системы эти процессы получают управление не через монитор, а по прерываниям от внешних устройств.

Прикладные процессы являются обработчиками запросов, поступающих от объекта или экспериментатора. Они не обращаются к внешним устройствам, память для них выделена статически. Поэтому, получив управление от монитора для обработки, они, безусловно, выполняют ее полностью, т. е. не могут быть приостановлены из-за ожидания не-

ление выполнением критических участков удобно осуществлять с помощью специального процесса, исполняющего функции, аналогичные монитору Хоара [4]. Прикладные процессы общаются с таким процессом через соответствующие порты, передавая запросы на доступ к неразделяемым ресурсам.

В данной системе реального времени, разработанной для электрофизиологических исследований, число процессов в ходе работы не меняется, а определяется при настройке системы на конкретный эксперимент. Основой для синхронизации вычислительных процессов и передачи сообщений являются однонаправленные очереди конечной длины. Очередь при создании процессов рассматривается как абстрактный тип данных [5], для которого определены следующие примитивы:

поместить (новый элемент в очередь),

получить (первый в очереди элемент),

которого ресурса или события. Прикладные процессы зависят от характера проводимых исследований.

Драйверы имеют в системе абсолютный приоритет, в том числе и по отношению к управляющему процессу. Это необходимо для обеспечения требуемого времени реакции системы при связи с объектом. Результаты работы драйверов проявляются в добавлении новых элементов в очереди сообщений к прикладным процессам и в очереди запросов к монитору на активизацию этих процессов.

Реализация и применение. Реализация данной разработки была выполнена на языке PL-11 для ЭВМ типа «Электроника 60». В ней не используются команды программных прерываний и не требуется в процессе работы поддержки со стороны штатного программного обеспечения. Поэтому система может функционировать как автономно, будучи подготовленной в формате абсолютной загрузки (наиболее типичный случай), так и под управлением некоторой ОС как прикладная программа. В качестве инструментальной ОС при разработке системы и ее настройках на конкретные виды исследований используется РАФОС.

Для такой настройки необходимо подготовить и поместить в библиотеку объектные модули, соответствующие процессам обработки, форматирования и ввода. Очереди создаются с помощью макропроцессора ПАГЕН [6] по описанию в виде ОЧЕРЕДЬ <имя очереди> = <число элементов в очереди> = <размер элемента в байтах>. Создается модуль трансляции на PL-11 и затем объектный модуль. При этом не только прикладные процессы и драйверы, но и управляющая программа не зависят от характера реализации очередей в системе.

Разработанная система использована при проведении экспериментов с биологической обратной связью, ряда нейрофизиологических исследований, анализа кардиоритмов. Для каждого вида исследований программное обеспечение встроенной ЭВМ содержит различное число (до 16 и более) прикладных процессов, несколько очередей экспериментальных данных. Его общий объем в отдельных случаях достигал 4 тыс. операторов PL-11, занимая около 24 кбайт. В качестве источников загрузки использовался перфоленточный ввод либо ППЗУ.

Заключение. Таким образом, основные концепции, принятые при разработке программного обеспечения встроенных в оборудование для электрофизиологических исследований ЭВМ, подразумевающие разделение программных процессов на «быстрые» и «медленные», обеспечение их взаимодействия и синхронизации с помощью единого механизма однонаправленных очередей, учет приоритетов программных процессов только на этапе выбора очередного вида работы, позволяют подготовить достаточно простые по структуре средства, удовлетворяющие в то же время основные потребности многопараметрических электрофизиологических экспериментов. Следует отметить, что описанная в работе технология подготовки программного обеспечения ориентирована на многократное проведение однотипных экспериментов, что является достаточно типичным в медико-биологических исследованиях.

Описанный подход, по нашему мнению, может быть применен для встроенных ЭВМ различной архитектуры.

ЛИТЕРАТУРА

1. Байцер Б. Микроанализ производительности вычислительных систем.— М.: Радио и связь, 1983.
2. Вальковский В. А., Котов В. Е., Марчук А. Г., Миренков Н. И. Элементы параллельного программирования.— М.: Радио и связь, 1983.
3. Майерс Г. Надежность программного обеспечения.— М.: Мир, 1980.
4. Hoare C. A. R. Monitors: an operating system structuring concept.— Comm. ACM, 1974, v. 17, p. 549—557.
5. Агафонов В. Н. Типы и абстракция данных в языках программирования.— В кн.: Данные в языках программирования. М.: Мир, 1982.
6. Вигдорчик Г. В. Интерактивный макропроцессор на языке ФОРТРАН-IV.— В кн.: Программное обеспечение управляющих комплексов системы малых ЭВМ. М.: ИНЭУМ, 1980.

Поступила в редакцию 10 октября 1985 г.