

4. Коварцев А.Н., Кораблин М.А., Шамашов М.А. Имитационное моделирование систем автоматизации эксперимента с использованием мультиторов полной конфигурации.—УСИС, 1979, № 4, с.124—127.

УДК 681.142.2

Л.М.Нусинзон

ОРГАНИЗАЦИЯ ДРАЙВЕРА УПРАВЛЕНИЯ АСИНХРОННОЙ РАБОТОЙ ПРИБОРОВ ЧЕРЕЗ ГРУППОВОЙ КОНТРОЛЛЕР

(г. Свердловск)

При создании средств прикладного программного обеспечения для автоматизированных систем экспериментальных исследований, когда приборы и другие компоненты системы (модули) объединены приборным интерфейсом через групповой контроллер с машинным интерфейсом [1], оказывается удобным считать приборные модули или их совокупности как дополнительные пронумерованные внешние устройства, подключенные к ЭВМ. Для выполнения этими приборными модулями некоторых макроопераций необходим реентерабельный драйвер (процедура), который может взять на себя все функции по управлению асинхронной работой этих модулей на протяжении всей макрооперации: от настройки модулей до ввода или вывода данных в память ЭВМ или на модуль. Для каждой группы приборных модулей, алгоритм управления которыми одинаков, составляется ряд программных обрабатывающих блоков, которые включаются в состав драйвера.

Программное управление модулем или группой модулей при выполнении какой-либо макрооперации можно представить в виде последовательности команд. Время выполнения приборным модулем каждой из команд различно и может быть соизмеримо или меньше времени реакции операционной системы $t_{пр}$ на внешнее прерывание, либо быть значительно большим. Поэтому команды целесообразно разбить на ряд последовательных групп. Каждая группа заканчивается командой, время выполнения которой $t_1 > n t_{пр}$, $n \geq 5$.

Время t_1 может определяться временем срабатывания или быстродействием модулей и приборного интерфейса. Например, временем, необходимым для коммутации заданного канала коммутатором или аналого-цифрового преобразования. Операции, связанные с передачей

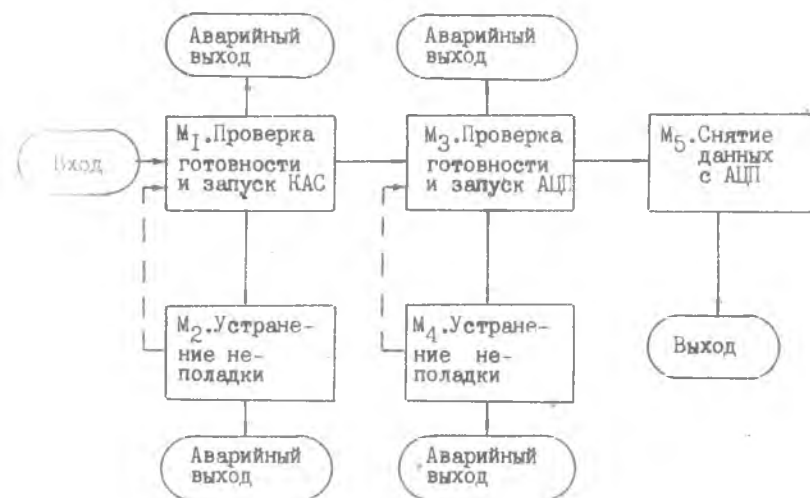
каждой группы команд, программируются в виде отдельного обрабатываемого модуля.

Таким образом, выполнение макрооперации по вводу-выводу сводится к последовательному выполнению группы обрабатываемых модулей (ОМ). Выполнение каждого ОМ может быть инициировано при: 1) обращении к драйверу прикладной программой для выполнения макрооперации; 2) команде прерывания, поступающей от группового контроллера, которая связана либо с окончанием выполнения приборным модулем последней "долгой" команды, посланной предыдущим программным модулем, либо запросом программного модуля на его обслуживание компьютером; 3) прерывании от таймера по истечении заказанного ранее интервала времени.

При наступлении любого из перечисленных выше событий инициируется выполнение единого управляющего модуля (УМ), который, в свою очередь, анализирует наступившее событие. В соответствии с событием и номером приборного модуля (вызвавшего это событие или указанном при обращении к драйверу) управляющий модуль инициирует выполнение очередного обрабатываемого модуля. В свою очередь, каждый ОМ анализирует прерывание и в соответствии с алгоритмом работы приборного модуля производит необходимые действия: выдачу управляющих кодов, обмен информацией с прибором, сообщения оператору и т.п. Выполнение каждого обрабатываемого модуля заканчивается либо возвратом в прикладную программу, либо переходом в ожидание следующего прерывания. Непосредственно перед переходом в "ожидание" каждый ОМ изменяет номера "двух последующих" обрабатываемых блоков в соответствующей таблице. Работа одного из них инициируется при следующем прерывании от соответствующего приборного модуля, а работа второго — по истечении необходимого времени. Прерывание по времени может быть заказано, например, когда для срабатывания приборного модуля необходима некоторая задержка или для предотвращения "зависания" модуля, которое может произойти, если приборный модуль в процессе выполнения "долгой" операции сломается и через время t_1 не сможет ответить собственным прерыванием. В последнем случае заказывается прерывание через время $t_2 = \max t_1$. Здесь следует отметить, что если приборный модуль ответил собственным прерыванием через время, меньшее t_2 , соответствующее прерывание по времени снимается очередным выполняющимся ОБ.

Таким образом, алгоритм последовательности работы ОБ заложен в самих ОБ.

Для примера рассмотрим структуру драйвера, осуществляющего управление измерением аналогового сигнала посредством коммутатора аналоговых сигналов (КАС) и аналого-цифрового преобразователя (АЦП), соединенных последовательно. Предполагается, что максимальное время, в течение которого производится коммутация t_k и аналого-цифровое преобразование t_n , несоизмеримо больше времени прерывания $t_{пр}$. В остальном быстродействие аппаратных модулей меньше $t_{пр}$. Тогда набор будет состоять из 5 модулей (рис. 1). Модуль



Р и с. 1. Структура драйвера при изменении аналогового сигнала посредством КАС и АЦП

M_1 осуществляет проверку работоспособности коммутатора (незанятость, исправность, готовность и т.п.) и подготовку его к коммутации заданного в обращении канала. Перед завершением работы модуль M_1 планирует инициализацию модуля M_3 при прерывании от этого коммутатора и модуля M_2 по прошествии времени t_k . Если раньше иницируется модуль M_3 , что возможно при прерывании от коммутатора, необходимо ликвидировать инициализацию модуля M_2 . В противном случае через время t_k модуль M_2 начнет свою работу по диагностике

несвоевременного срабатывания коммутатора, и в соответствии с причиной ϵ дается принято решение об аварийной передаче управления из драйвера в прикладную программу, сообщении оператору, повторной запуске коммутатора и т.п.

Организация работы модулей M_3 , M_4 и M_5 аналогична.

Драйвер имеет таблицу, каждое слово которой соответствует приборному модулю, слово содержит информацию о состоянии этого модуля (свободен, занят, неисправен, отсутствует в системе и т.п.). Если модуль занят (т.е. он еще не закончил участие в выполнении какой-либо макрооперации), то соответствующее слово содержит адрес информационного массива A . Для мини-ЭВМ СМ-3 и "Электроника-60" A может быть заключен в пределах $500_8 \leq A < 157776$. Поэтому для кодировки незанятых состояний приборного модуля остается значительный диапазон чисел.

Информационный массив предоставляется в распоряжение драйвера прикладной программой и содержит в себе данные, необходимые для выполнения требуемой макрооперации, код этой макрооперации и необходимое рабочее поле. В этом же массиве содержатся результаты работы драйвера, в том числе и код завершения его работы (макрооперация в процессе выполнения, выполнена успешно, модуль неисправен, занят и т.д.).

Для эффективного использования вычислительных ресурсов ЭВМ целесообразно воспользоваться системными макрокомандами ФОБОСа: в драйвере обмена — *RSUM* (возобновить работу задания, которое ранее было прервано по макрокоманде *SPND*), а в прикладной программе — *SPND* (приостановить выполнение задания).

Перечисленные способы организации драйвера позволяют параллельно управлять работой как различных, так и одинаковых приборных модулей с помощью одного драйвера.

Фортран операционной системы ФОБОС позволяет использовать все системные возможности ФОБОСа, что позволяет создавать системное математическое обеспечение на Фортране. Описанная выше организация драйвера была реализована на Фортране в виде двух процедур реального времени. Первая процедура содержит в себе все первые модули каждой последовательности обрабатываемых модулей. Здесь же (по таблице драйвера, являющейся общей для обеих процедур) определяется состояние необходимых приборных модулей. Если модули исправны и свободны, адрес информационного массива помещается в таблицу драйвера по адресам, соответствующим необходимым приборным модулям.

Затем выполняется первая группа команд по управлению приборными модулями. После окончания выполнения этой группы команд планируется ввод второй процедуры драйвера в качестве асинхронной программы завершения в случае возникновения прерывания от контроллера и по истечении времени t_2 . В последнем случае идентификационное целое равняется номеру "опаздывающего" приборного модуля. В информационном массиве указываются номера меток соответствующих обрабатываемых модулей. На этом работа первой из процедур драйвера завершается и производится возврат в прикладную программу. До полного окончания выполнения макрооперации возможно следующее обращение к драйверу, т.е. к первой из процедур.

Во второй процедуре (содержащей все последующие программные модули и работа которой инициируется соответствующим прерыванием с наивысшим приоритетом) в соответствии с номером прибора или идентификационным целым определяется информационный массив. Затем по номеру метки, указанному в этом массиве, соответствующему модулю передается управление. После завершения всех групп операций и установки в "свободное состояние" использованных приборных модулей в таблице драйвера выполняется библиотечная системная процедура *RESUME*.

Использование описанного выше драйвера особенно эффективно при создании средств программного обеспечения автоматизированных систем научного эксперимента с относительно неизменным набором приборных модулей. Модульность структуры драйвера позволяет реализовать его не только в ФОБОСе, но и во многозадачных операционных системах для ЭВМ типа СМ-3 - таких, как ОС РВ. В последнем случае каждый из обрабатываемых модулей можно рассматривать как подзадачу со своим приоритетом.

Л и т е р а т у р а

1. Цветков Э.И. Развитие работ по созданию измерительно-вычислительных комплексов. - Приборы и системы управления, 1980, № 1, с. 17-19.