

68

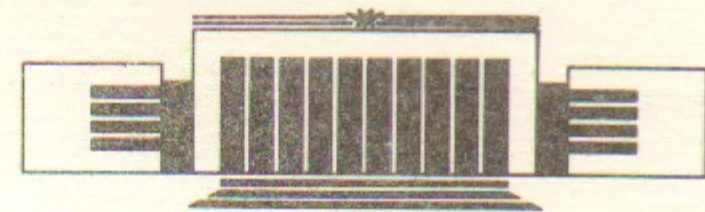
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР



Б.Л.Сысолетин

СПЕЦИАЛИЗИРОВАННАЯ
ОПЕРАЦИОННАЯ СИСТЕМА ОС/Р-5
ДЛЯ МИНИ-ЭВМ М-6000

ПРЕПРИНТ 86-111



НОВОСИБИРСК
1986

АННОТАЦИЯ

Кратко описываются назначение и конструкция специализированной операционной системы для мини-ЭВМ М-6000. Система предназначена для использования в режиме мультидоступа к фиксированному числу задач в условиях многомашинного комплекса и обеспечивает обслуживание линий связи, а также мультипрограммное исполнение переменного числа процессов. Настройка системы на конфигурацию ЭВМ и набор задач осуществляется при помощи управляющих таблиц.

ВВЕДЕНИЕ

Эволюция многомашинного комплекса РАДИУС, имеющего звездообразную конфигурацию и базирующегося на мини-ЭВМ М-6000 [1—3], протекала в направлении выделения централизованных специализированных подсистем [4], образующих ядро комплекса, и расширения состава периферийных ЭВМ. Эти централизованные («коммунальные») подсистемы должны работать в режиме мультидоступа, то есть одновременно обрабатывать несколько запросов различных пользователей, причем количество и типы этих запросов являются случайными. Так как первые такие подсистемы реализовывались на несколько модифицированных ЭВМ М-6000, то для этой машины и была, в первую очередь, разработана специализированная операционная система Р-5.

Аппаратные компоненты, поддерживаемые ОС/Р-5, включают:

- процессор А131-7 с расширением оперативной памяти до 256 К слов [5];
- пульт оператора VT-340 или KSR-180;
- модифицированный канал прямого доступа к памяти;
- эмулятор селекторного канала ЕС ЭВМ;
- таймер ИВИ-1;
- контроллер НМД ЕС-5060;
- контроллер НМД ЕС-5061;
- контроллер НМЛ ЕС-5017;
- последовательные линии связи с интерфейсом ИПС-1.

Модифицированный канал прямого доступа к памяти, эмулятор селекторного канала ЕС ЭВМ, таймер ИВИ-1, контроллер НМД ЕС-5060, а также интерфейс ИПС-1 разработаны и реализованы в ИЯФе. Ниже приводятся их краткие характеристики.

Модифицированный канал прямого доступа к памяти, имеющий два подканала, выполняет все функции КПДП М-6000, но в отличие от него может программно подключаться к любому интерфейсу, расположенному в расширителе ввода-вывода М-6000. Оба подканала работают независимо друг от друга.

Эмулятор селекторного канала ЕС ЭВМ (ЕСК) является устройством, обеспечивающим подключение к М-6000 внешних устройств ЕС ЭВМ. Он реализует алгоритм селекторного канала ЕС ЭВМ и независимо от ЭВМ выполняет каналные программы. По сравнению с селекторным каналом ЕС ЭВМ эмулятор накладывает на каналные программы следующие ограничения: не допускается выполнение чтения или записи одного физического блока данных с использованием нескольких разрывных областей памяти, отсутствует команда обратного чтения, каналные команды управления могут располагаться только в конце каналной программы (или быть ее единственной командой). Пропускная способность эмулятора ограничивается пропускной способностью оперативной памяти М-6000.

Таймер ИВИ-1 позволяет генерировать сигналы прерывания с программно-настраиваемыми интервалами 100 мкс; 1, 10, 100 мс; 1, 10, 100 и 1000 с. Кроме того, таймер имеет доступный программе 16-разрядный счетчик с программно-управляемой ценой младшего разряда (1, 10, 100 мкс; 1, 10, 100 мс). Еще одной особенностью таймера является индикация пропуска опроса сигнала готовности (прерывания), что позволяет при использовании его для реализации программных часов поддерживать достоверность их показаний.

Контроллер НМД ЕС-5060 управляет несколькими (до четырех) накопителями, обеспечивает чтение/запись произвольного объема данных (кратного размеру сектора НМД) по одной команде и индикацию условия завершения операции. Канал обмена данными может подключаться либо к КПДП М-6000, либо к модифицированному каналу прямого доступа к памяти.

Интерфейс последовательной связи ИПС-1 позволяет с помощью телефонной линии (витая пара) связывать машины, удаленные друг от друга на расстояние до 600 м. Тактовая частота в линии 1,25 МГц. Интерфейс предназначен для работы в асинхрон-

ном режиме и обеспечивает передачу двух типов 16-разрядных слов (команды и данные) с контролем каждого байта по четности, а также короткого (3 бита) сигнала, используемого для подтверждения приема данных. Совместимые с ИПС-1 интерфейсы разработаны также для ЭВМ Электроника 60 и Электроника 100-25.

Операционная система настраивается на конкретную конфигурацию внешних устройств и набор допустимых процессов прикладной системы при помощи ряда таблиц, оформляемых в виде модулей на языке ассемблера. Результатом процесса генерации являются образы памяти резидентной части (операционной системы вместе с резидентом прикладной системы), сегментов прикладных процессов и справочник сегментов. Готовая к работе система помещается на резидентное устройство, откуда и загружается для исполнения.

В следующих разделах приводится краткое описание состава операционной системы Р-5, управления процессами, управления внешними устройствами и подсистемы связи, обеспечивающей взаимодействие построенной на базе ОС прикладной системы с ее абонентами.

СОСТАВ ОПЕРАЦИОННОЙ СИСТЕМЫ

Все множество функций, заложенных в операционную систему, разбито на несколько достаточно замкнутых групп. Особое внимание уделялось определению интерфейса между группами функций с целью минимизации и упрощения взаимодействия функций, принадлежащих к разным группам. Ниже приводится список этих групп и их краткая характеристика.

- Управление очередями и семафорами.
- Управление оперативной памятью.
- Служба времени.
- Управление процессами.
- Супервизор обмена.
- Подсистема связи (управление сообщениями).
- Взаимодействие с оператором.
- Управление сегментами.
- Сбор статистических данных.

Управление очередями и семафорами служит для упорядочения

и синхронизации процессов, исполняющихся под управлением ОС. Очередь определяется своим заголовком, который состоит из двух указателей—на первый и последний элементы в очереди. Структура элементов произвольна за одним исключением: каждый из них должен содержать однословное поле связи. В качестве признака отсутствия связи используется ноль. Для очередей определены четыре операции: постановка в хвост очереди, выборка из начала очереди, удаление из произвольного места в очереди и проверка состояния очереди.

Управление оперативной памятью предназначено для расширения оперативной памяти и выполняет перемещение процессорного и четырех канальных «окон» (меняет значения регистров базы процессора и каналов) в адресном пространстве расширения памяти. Размер и шаг «окна» имеют значения 4 К слов и 1 К слов, соответственно. Аппаратное время смещения «окна» равно 3,5 микросекунды. Для процессорного «окна» реализовано ещё и временное изменение значения регистра базы. При этом старое значение регистра упрятывается в стек, а новое заносится в регистр. Восстановление старого значения заключается в замене текущего значения базового регистра значением, выбираемым из стека. Временные изменения регистра базы процессора используются, в частности, при обработке прерываний.

Служба времени операционной системы обеспечивает работу программных часов, а также измерение величин интервалов между некоторыми событиями. Кроме того, она позволяет асинхронно исполнять указанные подпрограммы по истечению заданного интервала времени. Единицей измерения таких интервалов (период генерации сигналов прерывания) является секунда. Выбранная цена младшего разряда счетчика таймера (точность определения значения времени) равна одной миллисекунде.

Управление процессами в каком-то смысле является основной компонентой операционной системы и предназначено для порождения и уничтожения процессов, смены состояния процессов и постановки процессов в различные очереди. К этой же группе относятся и планировщик процессора. Модули управления процессами имеют выходы в программы сбора статистики, что позволяет получить достаточно полную картину функционирования системы.

Супервизор обмена предназначен для управления внешними устройствами и обработки прерываний. В состав супервизора обмена входят независимые от конфигурации системы обработчики прерываний, модули запуска и обработки завершения операции

обмена, а также драйверы устройств, набор которых зависит от состава внешних устройств ЭВМ. Также как и управление процессами, супервизор обмена содержит обращения к входным точкам программ сбора статистики.

Подсистема связи реализует два нижних уровня трехуровневой системы протоколов взаимодействия процессов в многомашинном комплексе. По существу она является частью супервизора обмена, а выделение ее в отдельную группу обусловлено как специфичностью функций, так и желанием локализовать в отдельной компоненте взаимодействие с другими машинами комплекса. Кроме чисто связных функций, подсистема связи выбирает конкретный процесс, выполняющий принятый запрос, и организует его запуск. Тем самым она устанавливает сеанс связи, в течение которого поддерживается обмен блоками данных между удаленным процессом и исполнителем его запроса.

Взаимодействие с оператором включает в себя драйвер пульта оператора, а также модули печати сообщений и запросов к оператору, приема ответов на запросы и приема команд оператора. Принятые команды оператора преобразуются в формат межмашинных сообщений, которые затем обрабатываются подсистемой связи. Модуль связи с оператором также может рассматриваться как часть супервизора обмена, а его выделение в отдельную группу было связано с упрощением супервизора и логической выделенностью пульта оператора среди других внешних устройств.

Управление сегментами организует исполнение сегментированных процессов вызывая, сегменты в транзитные области. Количество транзитных областей не лимитируется, но ввиду отсутствия возможности перемещения программ в памяти каждый сегмент должен быть настроен на соответствующую область во время генерации системы.

Строго говоря, группа сбора статистики не входит в состав операционной системы, так как количество фиксируемых и обрабатываемых статистических данных определяется прикладной системой. С помощью подпрограмм этой группы реализуется интерфейс между прикладной системой и модулями операционной системы.

Кроме перечисленных функциональных групп, в состав операционной системы включен ряд подпрограмм, часть из которых доступна не только программам ОС, но и прикладным процессам.

Основным объектом операционной системы является процесс (точнее его динамический экземпляр). В силу «коммунальности» реализуемых на базе ОС прикладных систем понятие процесса имеет двойственную природу. С одной стороны, процесс является объектом мультипрограммирования, конкурирующим во время своего исполнения с другими процессами за использование различных ресурсов (процессора, общих данных, устройств и т. д.). С другой стороны, процесс может рассматриваться как абонент, участвующий в проведении сеанса связи (исполнения запроса) с некоторым удаленным, то есть расположенным в другой ЭВМ, абонентом. Операционная система, получив некоторый запрос, порождает динамический экземпляр процесса, исполняющего этот запрос (устанавливает сеанс связи), и обеспечивает взаимодействие (обмен сообщениями) между абонентами, участвующими в проведении этого сеанса. Прекращение сеанса происходит либо по инициативе одного из абонентов, либо по инициативе операционной системы.

Для обеспечения режима мультидоступа необходимо обеспечить мультипрограммное исполнение различных процессов, а также возможность параллельного исполнения однотипных запросов различных абонентов. Проще всего параллельное исполнение однотипных запросов реализуется с помощью реентерабельных программ, однако для реализации таких программ, в общем случае, необходимо выполнение следующих условий:

1. Наличие индивидуальной рабочей области для каждого вызова данной программы.
2. Обеспечение размещения всех рабочих (в терминологии АЛГОЛа-60 собственных) переменных в отведенной рабочей области.

Если выполнение первого условия достаточно просто в реализации, то для выполнения второго необходима либо адекватная аппаратная организация ЭВМ, либо специальная организация как трансляторов, так и системного программного обеспечения. К сожалению, оба этих условия для ЭВМ М-6000 не выполняются.

Рассмотрим более подробно причины, мешающие реализации реентерабельных программ. Во-первых, введем понятия постоянных и временных переменных. Под постоянными переменными будем понимать явно описанные в тексте программы переменные. Временными переменными мы будем называть такие переменные,

которые создаются трансляторами и скрыты от программиста. Кроме того, введем понятие активности переменной как такого ее состояния, когда ее значение уже вычислено, но еще не использовано. При этом интервал активности переменной начинается от вычисления ее значения до последнего по времени использования этого значения. Во-вторых, выделим два класса переключения процессора с текущего процесса на другой: синхронные и асинхронные. Первый класс связан с переходом текущего процесса в состояние ожидания по «собственной» инициативе, например, при запросе некоторого ресурса или при выдаче запросов на обмен. Во второй класс попадают переключения, инициированные событием, переводящим некоторый процесс из состояния ожидания в состояние активности. Такими событиями могут быть различные прерывания: по таймеру, по концу обмена и т. п. Основную трудность вызывают именно асинхронные переключения процессора, так как для синхронных переключений можно (приложив некоторые усилия) гарантировать отсутствие (на момент переключения) активных временных переменных, а постоянные переменные могут быть размещены в рабочей области.

Таким образом, запретив асинхронные переключения процессора (кроме, естественно, переключений на программы обработки прерываний), мы обеспечим квазиреентерабельность программ, заплатив за это отсроченным переключением процессора на более приоритетные процессы. Однако во многих случаях применения ЭВМ такая отсрочка либо вполне допустима, либо вообще все процессы имеют одинаковый приоритет.

Такое решение позволяет достаточно просто реализовать квазиреентерабельные программы в операционной среде с ограниченным режимом мультипрограммирования, не поддерживающим явное квантование времени. Этот способ и применен в Р-5 для параллельной обработки нескольких однотипных запросов при наличии одного экземпляра программного кода процесса.

Описание процессов прикладной системы строится в операционной системе следующим образом (см. рис. 1): все процессы конкретной прикладной системы разбиваются на группы, названные подсистемами. Каждой такой подсистеме присваивается свой сетевой адрес (см. пункт Подсистема связи). Для поиска конкретной подсистемы создается статическая таблица подсистем, вход в которую осуществляется по номеру подсистемы (ее сетевому адресу). Элемент этой таблицы содержит указатель на таблицу базовых дескрипторов процессов (БДП) подсистем, которая связывает не-

Таблица подсистем

Число подсистем (элементов таблицы)
указатель на таблицу БДП подсистемы 0
.....
указатель на таблицу БДП подсистемы N

Таблица Базовых Дескрипторов Процессов (БДП) подсистемы

БДП процесса 1
.....
БДП процесса N

Базовый дескриптор процесса

Количество существующих экземпляров процесса
Максимально допустимое число экземпляров процесса
Адрес точки входа
Заголовок очереди ожидания процесса
Верхняя граница значений выполняемых процессом функций (для последнего процесса в таблице 177777₈)

Динамический дескриптор процесса

Поле связи
Состояние процесса
Значения регистров для продолжения
Адрес продолжения
База и адрес буфера с обрабатываемым сообщением
Указатель на БДП

Рис. 1. Структуры данных управления процессами

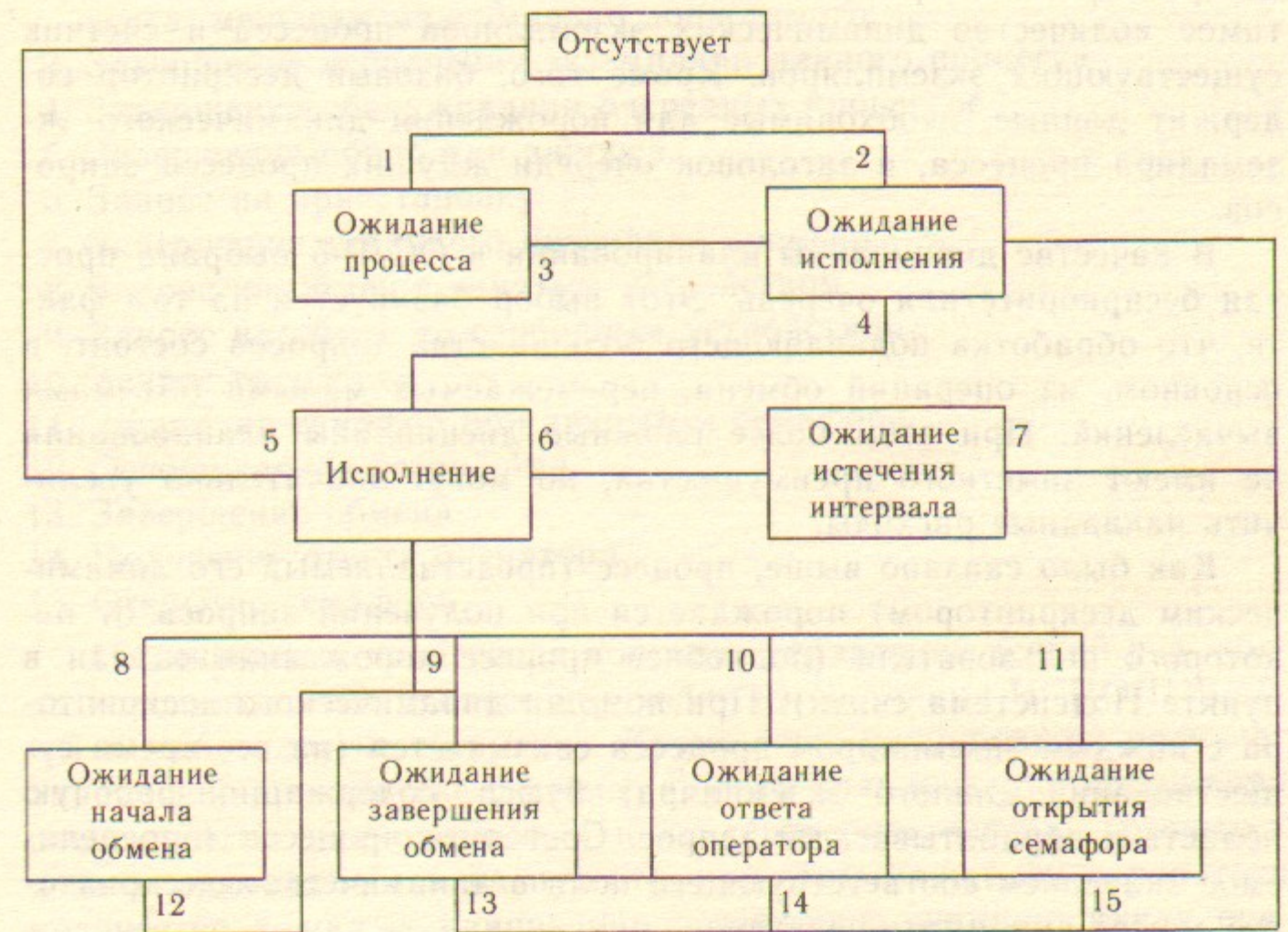


Рис.2. Граф переходов между состояниями процесса

который поддиапазон реализуемых подсистемой функций с определенным процессом. На этом уровне процесс описывается базовым дескриптором, который, в частности, содержит максимально допустимое количество динамических экземпляров процесса и счетчик существующих экземпляров. Кроме того, базовый дескриптор содержит данные, необходимые для порождения динамического экземпляра процесса, и заголовок очереди ждущих процесса запросов.

В качестве дисциплины планирования в ОС/P-5 выбрана простая бесприоритетная очередь. Этот выбор базируется на том факте, что обработка подавляющего большинства запросов состоит, в основном, из операций обмена, перемежаемых малыми объемами вычислений. При этом более сложные дисциплины планирования не имеют заметного преимущества, но могут значительно увеличить накладные расходы.

Как было сказано выше, процесс (представляемый его динамическим дескриптором) порождается при получении запроса от некоторого пользователя (подробнее процесс порождения описан в пункте Подсистема связи). При помощи динамического дескриптора с каждым экземпляром процесса связывается (на все время существования данного экземпляра) буфер, содержащий рабочую область и обрабатываемый запрос. Состояние процесса (определяемое значением соответствующего поля в динамическом дескрипторе) может принимать следующие значения:

- ждет процесса,
- ждет процессора,
- ждет завершения интервала времени,
- ждет начала операции обмена,
- ждет завершения операции обмена,
- ждет открытия семафора,
- ждет ответа оператора,
- выполняется.

Со всеми состояниями (за исключением последнего) связана некоторая очередь, в которую процесс попадает при переходе в указанное состояние. Переход процесса из одного состояния в другое инициируется либо самим процессом, либо некоторыми (внешними по отношению к процессу) событиями. Граф допустимых переходов между состояниями и события, вызывающие эти переходы, показаны на рис. 2. На этом рисунке цифрами обозначены события, вызывающие указанную смену состояния процесса:

1. Получение запроса при количестве активных экземпляров процесса больше, или равном максимально допустимому.
2. Получение запроса при количестве активных экземпляров процесса, меньшем максимально допустимого.
3. Завершение исполнения экземпляра данного процесса.
4. Завершение обслуживания очередных процессов.
5. Завершение обработки запроса.
6. Запрос на приостановку.
7. Завершение заданного интервала времени.
8. Запрос на обмен с занятым устройством.
9. Запрос на обмен со свободным устройством.
10. Запрос оператору.
11. Запрос на семафор при закрытом семафоре.
12. Освобождение устройства.
13. Завершение обмена.
14. Получение ответа оператора.
15. Открытие семафора.

Планировщик процессора получает управление всякий раз, когда исполняющийся процесс переходит из состояния ИСПОЛНЯЕТСЯ в любое другое состояние. Кроме того, планировщик получает управление при запуске системы. Логика планирования процессора очень проста: планировщик в цикле проверяет очередь готовых к исполнению процессов на пустоту, при этом может фиксироваться количество холостых циклов. По исчерпанию счетчика цикла время простоя процессора передается одной из подпрограмм сбора статистики, после чего счетчику присваивается начальное значение. При появлении в очереди готовых к исполнению процессов какого-либо процесса он выбирается из очереди, процессорная база расширения памяти устанавливается в соответствии с ее значением в динамическом дескрипторе выбранного процесса, после чего из области спасения регистров в динамическом дескрипторе восстанавливаются значения регистров и управление передается в точку продолжения процесса.

Синхронизация исполнения различных процессов реализована при помощи программных семафоров. Семафор состоит из счетчика и заголовка очереди семафора, в которую попадают процессы, пытающиеся пройти под закрытым семафором. При открытии семафора активным становится процесс, находящийся в голове очереди. Реализация позволяет использовать как двоичные, так и общие семафоры (отличающиеся от двоичных только начальной установкой счетчика семафора).

Каждый семафор определяется двухкомпонентным номером — номером группы и номером в группе, причем каждый из этих номеров может принимать значения от 0 до 255, а полный номер размещается в одном слове М-6000. Семафоры одной группы располагаются в таблице, причем номер семафора является его индексом в этой таблице. Доступ к таблице семафоров осуществляется через таблицу групп семафоров по номеру группы.

Для функционирования ОС необходимы две группы семафоров. Первая (содержащая один семафор) используется для установления и поддержания очереди процессов на выдачу запроса оператору с получением от него ответа. Вторая (содержащая $N+1$ семафор) необходима для обслуживания N транзитных областей для сегментов процессов. В существующих реализациях ОС максимальное число транзитных областей равно двум. Другие группы семафоров должны (при необходимости) определяться потребностями конкретной прикладной системы.

Для семафоров реализованы три операции:

- Р-операция (закрытие семафора).
- V-операция (открытие семафора).
- Опрос состояния семафора.

Если процесс использует несколько семафоров, то выполнение Р- и V-операций над ними должно подчиняться правилу вложенности: если рассматривать Р- и V-операцию над одним и тем же семафором как помеченные левую и правую скобки, то последовательность всех Р- и V-операций должна образовать правильную скобочную структуру. Правила работы с семафорами, предотвращающие возникновение тупиковых ситуаций, явно не задаются и должны вырабатываться при реализации конкретного набора процессов (прикладной подсистемы).

При выполнении всех перечисленных выше операций над семафорами первым действием является проверка заданного номера семафора. Если указанному номеру не соответствует реальный (определенный в системе) семафор, то управление возвращается вызвавшей программе с соответствующим кодом возврата.

Для выхода из состояний «зависания», возникающих в результате потери каких-то событий, в ОС/Р-5 используется служба времени. Эта служба, в частности, обеспечивает работу программных «будильников», предназначенных для асинхронного (по отношению к процессам) исполнения подпрограмм реакций на ситуации тайм-аута. Эти подпрограммы исполняются в состоянии обработки прерывания и либо активируют повторное выполнение «за-

висшего» действия, либо индицируют соответствующему процессу возникшую ситуацию. Кроме того, служба времени выполняет функции часов астрономического времени и позволяет получать текущие значения даты и времени в различных форматах.

«Будильники» работают следующим образом: любая программа может поставить свободный «будильник» на интервал указанный в квантах таймера (секундах). При этом указывается адрес подпрограммы обработки «звонка» будильника (программы асинхронного выхода в терминах ОС ЕС) и задается единственный параметр, передаваемый этой подпрограмме. По истечении заданного интервала служба времени запускает указанную подпрограмму. До истечения указанного интервала можно сбросить «взведенный» будильник. Опрос состояния и обработка «зазвеневших» будильников выполняются во время обработки прерываний от таймера. Все будильники размещаются в одной таблице. Доступ к конкретному будильнику выполняется либо последовательным поиском (при установке будильника), либо по его номеру (при сбросе). Механизм «будильников» используется и для приостановки исполнения процесса на некоторое время по его запросу.

Установка начальных значений различных системных параметров (например, даты и времени) должна выполняться процессом инициализации прикладной системы. Этот же процесс должен вызвать подпрограмму, инициализирующую механизм прерываний ЭВМ.

Некоторые процессы, для которых время обработки запросов не слишком критично (например, обработка команд оператора), могут сегментироваться, то есть разбиваться на головную часть, постоянно находящуюся в оперативной памяти, и несколько последовательно исполняющихся частей. Сегментация существенна для экономии оперативной памяти. Исполнение сегментов реализуется модулем управления сегментами. Для исполнения сегменты загружаются в свои транзитные области (количество областей операционной системой не ограничивается). Модуль управления сегментами использует (для N транзитных областей) $N+1$ семафор. (Один семафор используется для защиты самого модуля, а N — для защиты транзитных областей.)

Справочник сегментов, содержащий 6-символьные имена сегментов, их размеры и номера используемых транзитных областей, а также дескрипторы транзитных областей, создаются во время генерации прикладной системы. Справочник хранится на резидентном устройстве (НМД) и при запуске системы переписывается в расширение ОЗУ.

Сегменты реализуются в виде подпрограмм-функций, возвращающих значение кода возврата. Этот код используется для указания модулю управления сегментами действий требуемых после исполнения сегмента: возврата управления головной части процесса или запуска очередного сегмента.

Для связи с оператором процессам доступны три операции:

- Печать информационных сообщений.
- Печать диагностических сообщений.
- Печать запросов к оператору и прием ответа.

Печать информационных сообщений заключается в буферизации сообщений и независимом от запросившей программы выводе накопленных сообщений. Буферизация сообщений позволяет запросившей программе продолжать исполнение, не ожидая завершения печати сообщения, и произвольным образом использовать память, ранее занимаемую сообщением. При переполнении внутреннего буфера процесс ожидает освобождения буфера и только после переписи указанного в обращении сообщения продолжает работу. Ситуация переполнения буфера регистрируется подпрограммами сбора статистики.

Печать диагностических сообщений по существу совпадает с печатью информационных сообщений и отличается от нее только тем, что перед выводом сообщения на пульте оператора печатаются выбираемые из базового дескриптора имя процесса и время генерации сообщения. Текст сообщения печатается непосредственно вслед за подстрокой идентификации.

Печать запросов к оператору с приемом ответа отличается от первых двух операций. Ее выполнение состоит в запоминании времени поступления запроса, переводе процесса в состояние ЖДЕТ ОТВЕТА ОПЕРАТОРА и запуске печати текста запроса. После появления оператора, о чем он уведомляет систему нажатием клавиши «пробел» на пульте, ему еще раз печатается запрос в формате диагностического сообщения, причем в строке идентификации указывается время поступления запроса, а не время печати. После приема ответ оператора записывается в указанный процессом буфер, а процесс переводится в состояние ЖДЕТ ПРОЦЕССОРА. Печать запроса оператору и прием ответа выполняется под semaфором. Время ожидания ответа оператора регистрируется подпрограммами сбора статистики.

Так как переключение процессора на другой процесс в операционной системе выполняется только при освобождении процессо-

ра исполняющимся процессом, то для процессов введена операция приостановки исполнения процесса. Она состоит в переводе исполняющегося процесса в состояние ЖДЕТ ПРОЦЕССОРА и постановке его в хвост очереди ждущих процессора процессов. Наличие такой операции позволяет конструировать интенсивно использующие процессор процессы без существенного ухудшения пропускной способности прикладной системы.

Уничтожение динамического экземпляра процесса происходит по запросу процесса и имеет две модификации. Процесс может завершиться после окончания обработки запроса от абонента, а может завершиться, передав частично обработанный запрос другому процессу. В любом случае при завершении процесса в базовом дескрипторе проверяется очередь ожидающих запросов и, если она не пуста, порождается новый динамический экземпляр процесса, обрабатывающий запрос из головы очереди. При этом передаваемый очередному процессу запрос должен быть синтаксически правильным.

УПРАВЛЕНИЕ ВНЕШНИМИ УСТРОЙСТВАМИ

Управление внешними устройствами выполняется супервизором обмена, который включает в себя модули обработки запросов на обмен, обработки прерываний, драйверы устройств и сервисные программы. Все внешние устройства в системе делятся на классы, в каждый из которых входят (возможно разнотипные) устройства, выполняющие совпадающие функции. Описания устройств, их текущего состояния и состояния выполняемой на устройстве операции сосредоточены в нескольких таблицах (рис. 3): классов устройств, блоках управления устройствами (БУУ) и их расширениях. Физические операции выполняются драйверами конкретных устройств, причем один драйвер может обслуживать произвольное число одинаковых устройств.

Некоторые устройства подключаются к ЭВМ не непосредственно, а через контроллеры и каналы. Информация о типе подключения также содержится в БУУ. Драйверы таких устройств чтобы занять контроллеры и каналы, которые являются ресурсами, не допускающими параллельного использования, должны обращаться к внутренним подпрограммам супервизора. Освобождение контроллеров и каналов также выполняется по явным запросам драйве-

Таблица классов устройств

Число элементов таблицы

Элемент класса 0

.....
Элемент класса N

Элемент таблицы классов устройств

Число устройств в классе

Адрес первого БУУ для устройств данного класса

Блок Управления Устройством (БУУ)

Поле связи (для постановки в очереди)

Адрес входа в драйвер устройства

Описание подключения устройства

Состояние устройства

Код и параметры текущей операции

Число и коды ошибок

Времена начала и завершения операции

Заголовок очереди ждущих запросов

Указатель на расширение БУУ

Расширение БУУ для ЕС-5060

Состояние тома

Имя тома

Регистры модифицированного КПДП для завершенной операции

Расширение БУУ для ЕС-5061/ЕС-5017

Состояние тома

Имя тома

Параметры тома (для ЕС-5017 не используется)

Последнее управляющее слово ЕСК

Адрес на томе (для ЕС-5017 не используется)

Канальная команда, слово состояния канала и байты уточненного состояния для первой ошибки

Канальная команда, слово состояния канала и байты уточненного состояния для последней ошибки

Область размещения канальных программ

Расширение БУУ для линий связи

Оставшееся число сеансов для данной линии

Поле для приема транспортного заголовка

Рис. 3. Структуры данных управления устройствами

ров. Для предотвращения ситуаций взаимной блокировки захват контроллера должен выполняться перед захватом канала.

Так как оперативная память ЭВМ имеет ограниченную пропускную способность, то для некоторых пар устройств параллельное выполнение обменов необходимо запретить. Для этого введен фиктивный последовательно используемый ресурс, М-канал, логически аналогичный каналу (или контроллеру). Драйверы таких устройств перед выполнением обмена должны захватить фиктивный ресурс. При занятости ресурса операция задерживается супервизором до его освобождения. Для предотвращения ситуаций взаимной блокировки захват М-канала должен выполняться после захвата реального канала.

Текущее состояние устройства описывается несколькими компонентами, каждому из которых соответствует бит поля состояния:

- неработоспособно;
- в автономе (offline);
- тестируется;
- ждет печати диагностики;
- занято (исполняется операция).

Кроме того, устройства, допускающие смену томов (внешняя память), обладают расширенным состоянием, которое описывает установленный на устройстве том и состояние тома.

Интерпретация перечисленных компонент состояния достаточно очевидна. Две последних компоненты («ждет печати», «занято») определяют состояние текущей операции на устройстве. Новая операция может быть начата только в случае готовности устройства (равенства этих двух компонент нулю). Бит «неработоспособно» определяется состоянием аппаратуры и может быть взведен в результате неудачной попытки выполнения операции. Переход аппаратуры в состояние «работоспособно» обычно сопровождается прерыванием, обработка которого сопровождается анализом состояния аппаратуры. Компоненты «тестируется» и «offline» могут сбрасываться и устанавливаться как по запросам процессов, так и по командам оператора. В результате установки некоторых битов состояния устройства другие биты могут быть сброшены. Так, при переводе устройства в состояние «тестируется» или «offline» сбрасывается признак неработоспособности.

Для выполнения операций обмена процесс выдает запрос супервизору обмена. В качестве параметров запроса используются код требуемой операции, значения базы, адреса и размера области

памяти, а также адрес блока устройства (для НМД). Обработка этих запросов состоит в подготовке запрошенной операции. При этом проверяется наличие заданного в запросе устройства и допустимость параметров. Параметры запроса определяющие операцию обмена переписываются в рабочую область процесса, а сам процесс переводится в состояние ЖДЕТ НАЧАЛА ОБМЕНА и ставится в очередь к устройству. Если заданное устройство свободно, то делается попытка запуска операции. Если же устройство занято (выполняется операция обмена), то запуск операции откладывается. После обработки запроса управление передается планировщику процессора.

Так как большинство действий с устройствами выполняется в режиме прерываний, то продолжение и завершение таких действий выполняется другой частью супервизора—обработчиком прерываний, который, как это и следует из его названия, обрабатывает все прерывания. Множество прерываний М-6000 можно разделить на два класса: внутренние и внешние. Из внутренних прерываний реальной обработке подвергаются только прерывания по особым случаям обмена, которые обрабатываются стандартным для М-6000 способом. Остальные внутренние прерывания вызывают останов процессора с индикацией причины останова.

Первый уровень обработки внешних прерываний получает управление аппаратно после возникновения условия прерывания. Этот обработчик спасает текущее состояние ЭВМ, временно запрещает все внешние прерывания и определяет код источника прерывания, который используется как индекс для входа в таблицу прерываний. В этой таблице каждому источнику прерываний соответствует слово, содержащее либо нуль (для прерываний неиспользуемых в конкретной системе), либо адрес обработчика второго уровня (групповое подключение), либо адрес БУУ (индивидуальное подключение). Прерывания от неиспользуемых источников маскируются. Для прерываний, имеющих два уровня обработки запускается соответствующая программа обработки второго уровня.

Если в таблице прерываний записан адрес БУУ, то перед запуском драйвера проверяется состояние устройства. Для активных устройств (например, линий связи) драйвер запускается даже если устройство находится в свободном состоянии. При этом в БУУ формируются все параметры специальной операции активного чтения. Драйверы обычных (пассивных) устройств запускаются только в случае их занятости. Возвращая управление обработчику прерываний драйвер указывает, завершилась ли текущая опера-

ция. Если операция продолжается, то выполняется восстановление состояния машины, разрешение внешних прерываний и возврат в точку прерывания. Если же драйвер завершил операцию, то далее вызывается программа обработки окончания операции, после чего делается попытка запуска очередной операции.

Обработчики второго уровня определяют тип прерывания и конкретное устройство, вызвавшее это прерывание. Затем они выполняют предварительный анализ состояния (аналогично обработчику первого уровня для индивидуально подключенных устройств) и запускают соответствующий драйвер.

После попытки запуска следующей операции и по выходу из обработчика второго уровня выполняется проверка освобождения ранее занятых ресурсов. При наличии запросов на освободившиеся ресурсы запускается соответствующий драйвер. После этих проверок и запусков выполняется возврат в точку прерывания.

При запуске операции, который может выполняться непосредственно после обработки запроса на обмен или после завершения очередной операции, процесс остается в очереди устройства, но его состояние меняется на ЖДЕТ ЗАВЕРШЕНИЯ ОПЕРАЦИИ ОБМЕНА.

По завершении (или прекращении) операции, заказанной некоторым процессом, выполняются следующие действия: проверка условий завершения и (при необходимости) диагностирование ошибок, проверка освобождения ресурсов и продолжение задержанных операций, перевод запросившего операцию процесса в состояние ЖДЕТ ПРОЦЕССОРА и индикация условий завершения операции и, наконец, выполняется проверка очереди к устройству и запуск очередной операции.

Условие завершения операции определяется по коду возврата из драйвера. В случае наличия ошибок (не обязательно фатальных) запускается зависящая от состава внешних устройств подпрограмма диагностики. Выводимая на пульт оператора диагностика зависит от типа устройства и, кроме того, может селективно (по классам устройств) отключаться и включаться оператором.

ДРАЙВЕРЫ ВНЕШНИХ УСТРОЙСТВ

Драйверы внешних устройств предназначены для выполнения взаимодействия с внешними устройствами на физическом уровне.

Ниже описывается интерфейс драйверов с остальной частью операционной системы.

Каждый драйвер имеет одну точку входа, а семантика конкретного обращения к нему задается тремя параметрами: адресом БУУ устройства, для которого вызывается драйвер; кодом запуска и кодом источника прерывания (для соответствующего кода запуска). Код запуска может принимать следующие значения:

- 0 — начало пассивной операции,
- 1 — начало активной операции,
- 2 — продолжение операции (прерывание),
- 3 — освобождение ранее запрошенного ресурса,
- 4 — звонок будильника ресурса,
- 7 — сброс (прекращение) текущей операции.

Возвращая управление, драйвер передает два параметра. Один из них определяет продолжение или завершение операции, а второй, имеющий смысл только при завершении операции, содержит код возврата для процесса.

Во время работы драйвер использует записанный в БУУ тип подключения для запроса и освобождения различных ресурсов. Кроме того, драйверам доступны подпрограммы запуска и останова канала прямого доступа, эмулятора канала ЕС, установки базы каналов, установки и сброса будильника. При обнаружении сбоев устройства драйвер должен сформировать однобайтовый код ошибки и обратиться к определенной подпрограмме для записи этого кода в БУУ.

В настоящее время в составе ОС/Р-5 могут работать драйверы следующих устройств: ЕС-5060 (НМД с фиксированными головками), ЕС-5061 (подключаемые при помощи контроллера ЕС-5561 и ЕСК), комбинированный драйвер ЕС-5061 и ЕС-5017 (подключаемые при помощи контроллеров ЕС-5561, ЕС-5517 и ЕСК) и драйвер линий связи с интерфейсами ИПС-1.

Драйвер ЕС-5060 реализует операции чтения и записи блоков данных с размером, кратным сектору НМД (256 байт). В случае возникновения случайных ошибок операция может повторяться до 6 раз. Коды ошибок содержат информацию о типе ошибки и фазе операции, на которой возникла ошибка.

Комбинированный драйвер ЕС-5061 и ЕС-5017 рассчитан на работу с блоками данных, имеющими максимальный размер 2048 байт. Дисковые пакеты перед использованием инициализируются секторами по 2048 байт с ключами по 24 байта. По отношению к

ЕС-5061 драйвер реализует операции чтения ключа и сектора, записи сектора и смены ключа. По отношению к ЕС-5017 реализуются операции чтения/записи блока, записи ленточного маркера, поиска маркера. В случае сбоев операции повторяются (возможно с корректирующими действиями). Кроме стандартных операций, этот драйвер допускает выполнение канальных программ, сформированных процессом.

Драйвер линии связи реализует операции обмена блоками стандартизованного формата. При этом реализуется линейный уровень протокола (см. ниже). Кроме этого, драйвер может работать в режиме программной «заглушки», возвращая в линию связи принятый код. Эта возможность позволяет тестировать связное обеспечение на фоне обычной работы прикладной системы.

ПОДСИСТЕМА СВЯЗИ

В многомашинном комплексе РАДИУС используется трехуровневая структура протоколов межпроцессного взаимодействия. Низший (линейный) уровень обеспечивает обмен стандартизованными блоками данных между двумя машинами. (Описание линейного уровня протокола приведено в Приложении 1). Второй (транспортный) уровень, кроме чисто транспортных функций передачи блока между произвольными машинами, обеспечивает установление, поддержание и завершение сеансов связи. И, наконец, третий уровень образуется протоколами прикладных систем. Количество таких протоколов третьего уровня определяется числом «коммунальных» систем, включаемых в состав комплекса.

Подсистема связи ОС/Р-5.0 обеспечивает реализацию двух низших уровней протокола. В силу того, что сообщения тесно связаны с обрабатываемыми их процессами, подсистема связи должна взаимодействовать с компонентами управления процессами. Кроме того, подсистема связи, в некотором смысле, является и частью управления устройствами, так как, в основном, сообщения поступают и выводятся из машины по линиям связи, драйверы которых управляются супервизором обмена. Подсистема связи в целом выполняет следующие функции:

- обмен сообщениями по линиям связи,
- установление сеанса связи с абонентом,
- обеспечение обменов сообщениями в течение сеанса,

- прекращение сеанса по инициативе одного из абонентов,
- аварийное прекращение сеанса в случае обнаружения «зависания».

Как указывалось выше, в качестве линий связи используются витые пары с тактовой частотой 1,25 МГц, обеспечивающие двухточечную связь. Аппаратура линий связи позволяет обмениваться 16-разрядными кодами, причем допускаются коды двух аппаратно различаемых типов — «команды» и «данные». Команды — используются для запроса, поддержания и разрыва связи на линейном уровне. Передаваемые сообщения организуются в блоки, содержащие, кроме собственно данных, заголовок (Приложение 2.). Контроль передаваемых данных выполняется аппаратно. Прием каждого 16-разрядного кода подтверждается, причем «команды» подтверждаются программно, «данные» подтверждаются короткой (3-битовой) последовательностью аппаратно после их приема.

Заголовок каждого блока содержит (Приложение 2) поле идентификации протокола, используемое транспортным уровнем поле типа блока (начало, продолжение, конец сеанса), размер тела блока и 16-разрядные адреса двух абонентов. Адрес каждого абонента состоит из номера машины (старший байт) и идентификатора абонента внутри машины (младший байт), интерпретация которого зависит от конкретной системы. В ОС/Р-5 идентификатор абонента является номером прикладной подсистемы (см. рис. 1).

В описываемой системе обмен блоком разбит на две части: обмен заголовком и обмен телом блока. Такое разбиение введено для того, чтобы обеспечение приемника могло проанализировать заголовок и, возможно, отказаться от приема блока. Контроль передаваемых данных возложен на аппаратуру, индицирующую наличие ошибок в принятом 16-разрядном коде. Обнаруживший ошибку абонент (программно) извещает об этом партнера. Решение о повторной передаче или прекращении обмена возложено на принимающего абонента.

После приема транспортного заголовка в расширение БУУ линии связи подсистема связи проверяет тип блока, в зависимости от значения которого выполняются различные действия. По запросу сеанса связи проверяется допустимость вызываемого абонента (отсутствие проводимого сеанса между указанными в заголовке абонентами) и возможность установления нового сеанса (существует ограничение на максимально допустимое число сеансов, в которых обмен данными выполняется по конкретной линии связи). В случае отрицательного решения партнеру выдается отказ на ли-

нейном уровне и передача тела блока не происходит. В случае положительного решения для этого сеанса выбирается буфер, в который и принимается весь блок.

После приема блока с запросом сеанса связи подсистема связи анализирует заголовок пользователя, который содержит, в частности, поле запрошенной функции (см. Приложение 2). При отсутствии соответствующего процесса порождается экземпляр процесса отказа, диагностирующий ситуацию на пульте оператора и передающий запросившему абоненту блок, содержащий отказ. При наличии описателя процесса, обрабатывающего указанную функцию формируется динамический дескриптор, связывающий буфер с принятым блоком и порождаемый экземпляр процессом. Для порождения процесса и постановки его в очередь к процессору подсистема связи обращается к модулю управления процессами.

Взаимодействие между абонентами после установления сеанса (порождения динамического экземпляра некоторого процесса) происходит при помощи обмена блоками. В запросе на передачу блока удаленному абоненту процесс указывает, ожидается ли ответ. Если требуется ответ, то запросивший передачу процесс после успешного завершения передачи ставится в очередь ожидания ответов абонентов, а его состояние получает соответствующее значение. При постановке процесса в эту очередь взводится будильник, «звонок» которого сигнализирует о ненормальности поведения удаленного абонента. После «звонка» будильника (или в случае безуспешной передачи блока) процессу индицируется аварийная ситуация и он ставится в очередь к процессору.

Если в заголовке принимаемого блока адресные части совпадают, то принятый блок ставится в очередь к линии связи, с которой он был принят. Эта возможность обеспечивает проверку функционирования, а в некоторых случаях и отладку связного обеспечения удаленной машины без нарушения функционирования прикладной системы.

ГЕНЕРАЦИЯ СИСТЕМЫ

Операционная система генерируется вместе с процессами прикладной системы в несколько этапов. На первом этапе создаются перемещаемые (объектные) файлы всех программ и таблиц. На втором этапе из перемещаемых программ при помощи перемещаю-

щего кросс-загрузчика генерируются коды в формате абсолютных записей М-6000. Затем при помощи генератора образа памяти формируются файлы, содержащие образы памяти резидентной части и сегментов, а также справочник сегментов. И, наконец, полученные образы памяти записываются на резидентное устройство системы (НМД ЕС-5060). После этого система готова к запуску.

Первые три этапа исполняются на инструментальной машине (работающей под управлением ДОС М-6000) после чего образы памяти переносятся в оперативный архив (центр РАДИУСа-4). Последний этап выполняется на рабочей машине независимой программой.

Существующая версия операционной системы занимает 10 К 16-разрядных слов М-6000, из которых 1,5 К занято таблицами.

Автор выражает свою глубокую признательность всему коллективу ВЦ-3 за внимательное отношение и поддержку в работе. Особо хотелось бы отметить Б.Н. Шувалова и М.В. Ясенева, принимавших активное участие в разработке протокола, а также Л.М. Миленького, написавшего ряд программ для генерации системы.

ЛИТЕРАТУРА

1. Система ЭВМ для автоматизации экспериментов / В.А. Гусев и др. — В сб.: Тр. семинара «Обработка физической информации». — Ереван, 1975.
2. В.А. Сидоров, Б.Л. Сысолетин, Б.Н. Шувалов. Программное обеспечение системы РАДИУС. — Управляющие системы и машины, №1, 1978.
3. Система ЭВМ РАДИУС. Опыт эксплуатации и развития. / В.А. Гусев и др. — В сб.: 11 Всесоюзное совещание по автоматизации научных исследований в ядерной физике. — Алма-Ата, 1978.
4. Б.Л. Сысолетин. Общесистемное математическое обеспечение проблемно-ориентированного многомашинного комплекса: Дисс. на соиск. уч. степени канд. тех. наук (01.01.10). — Новосибирск: ИЯФ СОАН СССР, 1983.
5. В.Ш. Банзаров, А.Н. Малыгин, Б.Н. Шувалов. Оперативное запоминающее устройство ОЗУ-256К для мини-ЭВМ М-6000. / Препринт ИЯФ № 82-134. Новосибирск, 1982.

ОПИСАНИЕ ЛИНЕЙНОГО УРОВНЯ ПРОТОКОЛА

Линейный протокол предназначен для управления линиями связи и определяет:

- вхождение в связь;
- обмен данными;
- обнаружение ошибок;
- процедуры восстановления.

Ниже приводится список команд в линиях связи и описание линейного протокола.

Команды в линиях связи

Команда	Сокр.	Источник	Код 8-ичный
Запрос Связи	ЗС	передатчик	170222
Запрос Принят	ЗП	приемник	170111
Нет Буфера	НБ	приемник	170303
Конец Заголовка	КЗ	передатчик	170051
Конец Связи	КС	передатчик	170224
Конец Заголовка Принят	КЗП	приемник	170112
Конец Связи Принят	КСП	приемник	170045
ОШибка в Команде	ОШК	пер/прием	170332
ОШибка Повторить	ОШП	приемник	170155
ОШибка Закончить	ОШЗ	приемник	170266
Отказ Транспортный	ОТ	приемник	170133

РЕАКЦИЯ НА КОДЫ СО «СВОБОДНОЙ» ЛИНИИ

1. При получении команды ЗС начать процедуру вхождения в связь для приемника.
2. При получении команды ОШК не предпринимать никаких действий.
3. При получении любой другой команды или данных (с паритетом или без) выдать в линию команду ОШК. При повторении ситуации более N раз команду ОШК не выдавать, линию считать неработоспособной.

ВХОЖДЕНИЕ В СВЯЗЬ

Процедура для передатчика:

1. Если на команду ЗС не получено никакого ответа в течение 5 секунд, то линия связи считается неработоспособной.
2. Если на команду ЗС получен любой ответ, кроме команд ЗП, ЗС и НБ, повторять (по счетчику) команду ЗС.
3. При получении команды НБ передатчик повторяет команду ЗС после завершения любого сеанса на данной линии связи.
4. Если принята команда ЗП, то сеанс связи считается начатым и передатчик переходит к обмену данными.
5. Если принята команда ЗС, то действия зависят от относительного приоритета: при более высоком приоритете никаких особых действий предпринимать не надо (перейти к п.1), при более низком приоритете начинается вхождение в связь для приемника.

Процедура для приемника:

1. При получении команды ЗС выдать команду ЗП или НБ.
2. Если после выдачи этих команд приходит команда ЗС, то перейти к п.1. (Количество повторений определяет передатчик.)
3. Если на команду ЗП в течение 5 секунд не получено никакого ответа, то линия считается свободной.
4. Если после выдачи команды ЗП приходят данные, то связь считается установленной и приемник переходит к обмену данными.

ОБМЕН ДАННЫМИ

Процедура для передатчика:

1. Передатчик выдает в линию Т-заголовок. После этого при нулевой длине сообщения происходит переход на п.4, иначе—на п.2.
2. Передатчик выдает команду КЗ. При получении команды ОТ или ОШЗ сеанс связи считается законченным аварийно. При получении команды НБ передатчик повторяет процедуру вхождения в связь для данного сеанса после завершения любого сеанса на данной линии связи. Получив команду КЗП, передатчик переходит к п.3. По команде ОШП—переход на п.1.
3. Передатчик выдает тело сообщения, а затем команду КС.
4. При получении команды ОШП передатчик переходит на п.1

(длина сообщения равна 0) или на п.3 (длина не равна 0). При получении команды ОШЗ сеанс связи считается законченным с ошибкой. При получении команды КСП или любой другой команды, кроме перечисленных выше, или данных сеанс связи считается законченным нормально.

Процедура для приемника:

1. Приемник, приняв транспортный заголовок и команду КЗ/КС, проверяет доступность указанного в заголовке абонента-приемника и наличие буфера для сообщения. Если абонент доступен и буфер для сообщения существует, то выдается команда КЗП (КСП) и приемник переходит к п.2. Если абонент доступен, но нет буфера, то приемник выдает команду НБ и считает линию свободной. Если абонент по каким-то причинам недоступен, то выдается команда ОТ. После выдачи команд КСП и ОТ сеанс считается завершенным.
2. Приняв тело сообщения и команду КС, приемник выдает команду КСП, после чего сеанс считается завершенным.

ОБНАРУЖЕНИЕ ОШИБОК

1. Для обнаружения неработоспособности партнера и приемник и передатчик после выдачи каждой команды и начала обмена данными взводят будильник на 5 секунд. Если до звонка будильника ответной команды (или данных) не будет получено, то считается, что сеанс завершен аварийно по тайм-ауту.
2. При обнаружении паритета или команды, не соответствующей выполняемой последовательности, или данных вместо команды передатчик выдает команду ОШК, а приемник команды ОШП или ОШЗ. Этот пункт не относится к процедуре вхождения в связь и распознаванию команды КСП.

ПРОЦЕДУРЫ ВОССТАНОВЛЕНИЯ

1. При получении команды ОШК и приемник, и передатчик находясь внутри сеанса связи повторно выдают последнюю команду и переходят в соответствующую фазу сеанса.
2. Приемник при получении команды ОШК может выдать команду ОШЗ, после чего сеанс считается законченным аварийно.
3. Передатчик при приеме команды ОШП переходит на начало передачи данных (заголовок или тела).

Приложение 2.

Формат линейного блока

T-заголовок

	Номера битов в слове M-6000														
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
00	T	TO		Тип			Канал								
01	Размер T-тела в словах M-6000														
02	Сетевой адрес абонента-приемника (кому)														
03	Сетевой адрес абонента-передатчика (кто)														

T-тело:

	Номера битов в слове M-6000														
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
U-заголовок 00	Размер U-тела в словах M-6000														
01—02	Идент пользователя*														
03	Пароль пользователя*														
04	Код завершения сеанса					Номер функции*									
05—19	Параметры функции*														
U-тело 00—1023	U-данные*														

T — идентификатор протокола (пока всегда 0).

TO — тип обмена: 0 — обмен сообщениями (архив, диспетчер),
1 — Терминальный обмен.

Тип — тип блока в типе обмена. Для обмена сообщениями:

0 — запрос,

1 — ответ,

3 — отказ в рамках протокола пользователей,

F — T-отказ (нет доступа к абоненту).

Канал — идентификатор логического канала (сеанса) абонента.

Используется для идентификации одного из нескольких сеансов, проводимых одним абонентом с одной коммунальной подсистемой. Номер определяется абонентом, начинающим сеанс, и не изменяется вторым абонентом.

Размер T-тела в словах M-6000 — длина сообщения без T-заголовка (для ОС/Р-5 до 1044 слов).

Сетевой адрес абонента-приемника и

Сетевой адрес абонента-передатчика — сетевой адрес, в котором старший байт является номером машины, а младший номером, интерпретируемым данной машиной (для ОС/Р-5 это есть номер подсистемы).

Размер U-тела в словах M-6000 — размер сообщения без U-заголовка (для ОС/Р-5 до 1024 слов);

Идент пользователя — идент пользователя в коде ГОСТ-67.

Пароль пользователя — 16-разрядное 2-число без знака.

Код завершения сеанса — индикатор завершения сеанса. При работе с архивом старший бит поля определяет последний запрос/ответ. Остальные 7 бит используются для индикации конкретной причины завершения сеанса (собственно код завершения).

Номер функции — номер затребованной в запросе функции.

Параметры функции — различные параметры, необходимые для указанной функции.

U-данные — собственно данные для указанной функции.

Поля, отмеченные знаком * операционной системой не анализируются. Описанное выше содержание этих полей относится ко второму и третьему уровням протокола работы с архивом данных (одной из прикладных систем комплекса РАДИУС).

Б.Л.Сысолетин

**СПЕЦИАЛИЗИРОВАННАЯ
ОПЕРАЦИОННАЯ СИСТЕМА ОС/Р-5
ДЛЯ МИНИ-ЭВМ М-6000**

Ответственный за выпуск С.Г.Попов

Работа поступила 26 мая 1986 г.
Подписано к печати 2 июля 1986 г. МН 11766
Формат бумаги 60×90 1/16 Объем 2,9 печ.л., 2,5 уч.-изд.л.
Тираж 180 экз. Бесплатно. Заказ № 111

*Набрано в автоматизированной системе на базе фото-
наборного автомата ФА1000 и ЭВМ «Электроника» и
отпечатано на ротапринтере Института ядерной физики
СО АН СССР,
Новосибирск, 630090, пр. академика Лаврентьева, 11.*