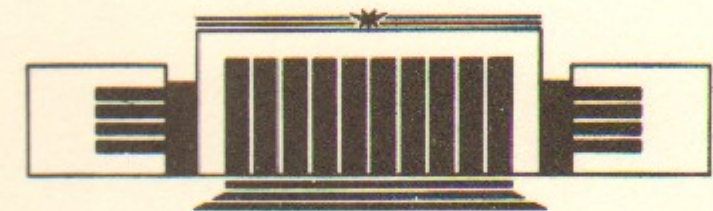




Э.Л. Неханевич, М.В. Яснев

СИСТЕМА ОРТ — ЛОКАЛЬНАЯ СЕТЬ
МИНИ- И МИКРО-ЭВМ

ПРЕПРИНТ 88-112



НОВОСИБИРСК

Система ОРТ-локальная сеть
мини- и микро-ЭВМ

Э.Л. Неханевич, М.В. Яснев

Институт ядерной физики
630090, Новосибирск 90, СССР

АННОТАЦИЯ

Описывается система ОРТ—низкоскоростная локальная сеть мини- и микро-ЭВМ на базе простой системы связи ДОЗА. Описаны топология связей, аппаратная конфигурация включенных в систему машин и организация связных протоколов. В приложении приведено детальное описание основных протоколов.

© Институт ядерной физики СО АН СССР

ВВЕДЕНИЕ

В начале 80-х годов в Институте ядерной физики появилось большое количество микро-ЭВМ «Электроника-60» в минимальной комплектации. Отсутствие необходимых устройств внешней памяти не позволяло в полной мере реализовать богатые функциональные возможности этих машин. Встал вопрос об объединении микро-ЭВМ в локальную сеть с целью их эффективного использования как в задачах автоматизации физического эксперимента, так и для наладки экспериментального оборудования. Требовалось обеспечить возможность быстро и удобно подготовить и загрузить для исполнения рабочую программу; обеспечить возможность доступа к удаленным обобщенным ресурсам (диски, ленты) через линию связи. Для обеспечения этих требований в состав сети обычно включается одна или несколько мини-машин, программно совместимых с «Электроникой-60». Такая сеть, объединявшая на начальном этапе около двадцати периферийных микро-ЭВМ и три рабочих машины «Электроника-100/25», эксплуатировалась на ВЦ-3 ИЯФ с середины 1982 года. В отличие от известных работ такого рода (см., например, [2, 5]), эта система с самого начала проектировалась как подсистема многомашинного комплекса РАДИУС ([1, 4]). Это естественно обусловило преимущество физической организации связи и сетевых протоколов. На первых этапах развития комплекса РАДИУС последовательно проводилась идея радиальной организации межмашинных соединений (что нашло свое отражение и в названии комплекса). Такая структура

связей обеспечивает возможность взаимодействия между собой любой пары машин и является более экономной по аппаратным затратам, чем «сеточная» конфигурация. Предполагалось использовать конфигурацию «звезды» и для узла машин серии «Электроника». Все машины были радиально подключены к КОММУТАТОРУ — процессору мини-ЭВМ «Электроника-100/25», выполняющему функцию сетевого узла коммутации. Подключение осуществлялось через разработанный для этой цели последовательный связной интерфейс с режимом прямого доступа в память [1]. Для ускорения транзитных передач в связной интерфейс была заложена возможность передачи данных из линии в линию по магистрали центральной машины без участия ее процессора. Магистраль коммутатора играла роль укороченной «широковещательной шины». Кроме машин серии «Электроника», с коммутатором были связаны СПД (терминальный концентратор комплекса РАДИУС) и дисковый АРХИВ комплекса.

В качестве базовой операционной системы рабочих машин использовалась система RSX-11M, обладающая развитыми средствами мультипрограммирования и многопользовательской защиты. Для этой системы были обеспечены диалоговая работа с удаленных терминалов и возможность обмена данными между задачами, исполняющимися в различных машинах. Доступ в систему рабочей машины с терминалов удаленных микро-ЭВМ позволял вести в «Электронике-100/25» подготовку программного обеспечения для «Электроники-60». В системе РАДИУС выход в RSX-11M терминалов СПД стал средством организации интерактивной подготовки и исполнения небольших фортрановских задач, как альтернатива пакетному исполнению этих задач на машинах серии ЕС.

Механизм межзадачного обмена явился базой для большинства системных задач, обеспечивающих сетевые функции (в частности, начальную загрузку периферийных машин и удаленный доступ к файлам рабочих машин). Разработка и эксплуатация пользовательских сетевых задач, выполняющихся одновременно в периферийной и рабочей машинах была затруднена нивелированием рабочих машин и удалением файлов пользователя при выходе из системы RSX (мера, диктуемая малым размером имеющихся дисков). Рабочая машина фактически играла роль вычислителя без функции хранения файлов. Такая дисциплина использования рабочих машин, ориентированная на терминальную работу с RSX-11M большого числа клиентов СПД, оказалась неудобной для построе-

ния распределенных задач. Для пользователя микро-ЭВМ основным режимом работы стала подготовка программы для «Электроники-60» в системе рабочей машины (в режиме терминального соединения) с последующей загрузкой собранных программ для автономного исполнения.

Эксплуатация системы выявила и другой ее недостаток — снижение надежности работы коммутационной машины при подключении к ее магистрали большого числа связных интерфейсов, использующих режим прямого доступа. Уменьшить нагрузку на коммутатор как по числу подключенных к нему линий, так и по интенсивности обменного трафика позволило появление электронных дисков различного рода. Используемые как локальные рабочие диски микро-ЭВМ, они существенно повысили степень автономности периферийных машин. Первым получил массовое распространение электронный диск на основе выпускаемого промышленностью блока памяти «Электроника-256К» [8]. Емкость такого блока вполне достаточна для работы операционной системы RT-11, позволяющей вести разработку программного обеспечения на периферийной машине. Отпала необходимость в режиме терминального доступа — режиме, наиболее чувствительном к быстродействию системы связи. Снижение требований к времени реакции системы позволило сгруппировать несколько микро-ЭВМ для выхода на коммутатор (и, соответственно, на все коммунальные ресурсы) через промежуточное звено — КОНЦЕНТРАТОР периферийных машин. Для организации «куста» (кустом будем называть концентратор с его периферийными машинами) была разработана система связи ДОЗА [7] с простыми байтовыми интерфейсами и программно-управляемым протоколом передачи данных. Продолжая «геометрическую» терминологию РАДИУСа, система концентраторов с подключенными к ним абонентскими машинами получила название ОРТ. Первый концентратор ОРТа был запущен в 1985 году.

В последующих главах описаны организация системы ОРТ и ее протоколов, а также возможности пользователей этой системы.

СТРУКТУРА И ФУНКЦИИ СИСТЕМЫ «ОРТ»

Система связи ОРТа базируется на последовательных каналах типа «точка-точка», топологически организованных в виде дерева с числом уровней, равным трем (см. рис. 1). Иерархическая струк-

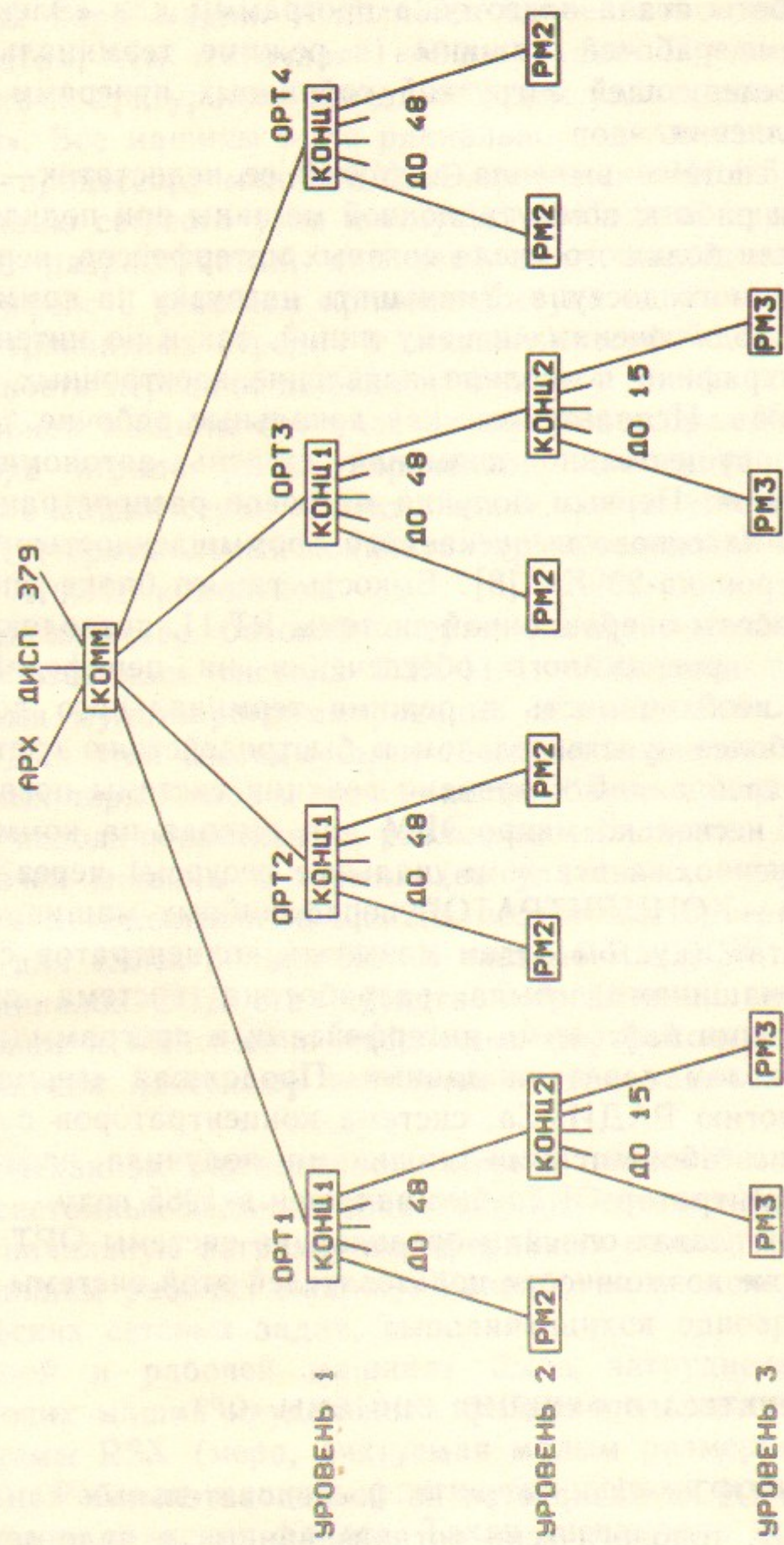


Рис. 1. Топологическая структура системы ОРТ. АРХ — дисковый архив; ДИСП — диспетчер машин серии ЕС; 379 — Электроника-79; КОНЦ1 — концентратор первого уровня; КОНЦ2 — концентратор второго уровня; РМ2, РМ3 — рабочие машины.

тура связей, с одной стороны, обеспечивает взаимодействие любых двух абонентских машин системы, с другой — замыкает в локальных «кустах» внутригрупповые потоки данных и сохраняет работоспособность «куста» при потере связи с остальной частью сети. Машины с интенсивным взаимным трафиком могут быть связаны дополнительным каналом, нарушающим древовидную топологию системы. Такие каналы не используются сетевой службой маршрутизации, их назначение — обмен между конкретной парой машин.

Машины, находящиеся в узлах дерева, будем называть концентраторами (КОНЦ1, КОНЦ2), остальные — рабочими (РМ). К каждому из четырех концентраторов первого уровня (ОРТ1 — ОРТ4) подключается до 48 абонентских машин. Ограничение на число абонентов определяется числом каналов используемого мультиплексора. На базе рабочей машины второго уровня (РМ2) также возможна организация концентратора (КОНЦ2). Такой концентратор поддерживает до 15 рабочих машин третьего уровня (РМ3). Здесь ограничение на число абонентов протокольное, определено размером поля, адресующего машины третьего уровня. Лишь концентраторы первого уровня (КОНЦ1), территориально размещенные в машинном зале ВЦ, являются вспомогательными сетевыми машинами. Концентратор второго уровня исполняет коммутационные функции «по совместительству», предоставляя пользователю практически полный набор услуг рабочей машины. Рабочие машины второго и третьего уровней полностью идентичны по возможностям, предоставляемым пользователю.

Каждой машине системы приписан номер, используемый для адресации сообщений к этой машине. Этот номер однозначно определяется номером концентратора, к которому подключена машина, и номером линии, через которую она подключена.

Для подключения абонентских машин в концентраторах первого уровня используется МПЛС48 — 48-канальный мультиплексор системы ДОЗА. В концентраторах второго уровня удобно использовать ППИ8 — 8-канальную плату последовательной связи [9]. Таблица 1 суммирует возможности использования различных типов интерфейсов.

В таблице использованы следующие сокращения: БИМ, БИС, КАМАК — связные интерфейсы системы ДОЗА [7] (БИМ — интерфейс с прерываниями; БИС — интерфейс без прерываний; КАМАК, КАМАК-ДОЗА — интерфейс в стандарте КАМАК), АДСП1 — адаптер последовательной межмашинной связи [9]. Все упомянутые интерфейсы разработаны в ИЯФ. Для связи (КОНЦ2-РМ3)

Таблица 1

Канал	Используемые интерфейсы	Среда передачи	Расст.
КОНЦ1-PM2	МПЛС48—БИМ, БИС, КАМАК	витая пара	1 км
КОНЦ1-КОНЦ2	МПЛС48—БИМ, КАМАК	витая пара	1 км
КОНЦ2-PM3	БИМ, КАМАК—БИМ, БИС, КАМАК	витая пара	1 км
КОНЦ2-PM3	ППИ8—АДСП1	коаксиал	100 м

могут быть использованы и другие интерфейсы типа DL-11, как байтовые, так и словные.

Аппаратная конфигурация КОНЦ1 включает в себя:

- процессор ИЦ-80 (МС1201.01) и ОЗУ (28К);
- связной мультиплексор системы ДОЗА (МПЛС48);
- интерфейс связи с коммутатором (ИПС60);
- двойной комплект винчестерских накопителей (ST225);
- двойной комплект электронного ВЗУ (Электроника-256К);
- пультовой терминал (общий для ОРТ1—ОРТ4).

Типичная конфигурация рабочей машины включает:

- процессор (М2, М6, PDP-нок [10]);
- ОЗУ;
- пультовой терминал;
- интерфейс связи (БИМ, БИС, КАМАК-ДОЗА для PM2; БИМ, БИС, КАМАК-ДОЗА, АДСП1 для PM3);
- начальный загрузчик (для процессора М6);
- дополнительные устройства (электронные и магнитные диски, магнитные ленты, печать, КАМАК, ЦДР, устройство виртуальной памяти для графических рабочих мест и т. д.).

В аппаратную конфигурацию КОНЦ2 для подключения рабочих машин включаются дополнительные связные интерфейсы типа DL (БИМ, КАМАК-ДОЗА или ППИ-8). При наличии в конфигурации КОНЦ2 энергонезависимого ВЗУ, «куст», состоящий из концентратора второго уровня и его рабочих машин, может эксплуатироваться вне связи с системой ОРТ.

Программное обеспечение концентратора реализует следующие функции:

- начальная загрузка рабочих ЭВМ;
- использование электронного диска концентратора в качестве общего системного диска рабочих машин;

- использование внешней памяти концентратора (например, винчестерского диска) как виртуального диска рабочих машин;
- обмен данными между рабочими машинами «куста»;
- транзит данных к другим машинам комплекса РАДИУС (дисковый архив, диспетчер машин серии ЕС, «Электроника-79»).

Для пользователя рабочей машины под операционной системой RT-11 доступны следующие функции:

- обмен файлами и управление дисковым архивом комплекса РАДИУС;
 - удаленный ввод заданий для машин серии ЕС;
 - взаимодействие удаленных процессов (RT-задач);
 - использование локального терминала как виртуального сетевого терминала;
 - управление удаленным устройством (частным случаем управления удаленным устройством являются виртуальные диски);
 - распределенный спулинг (модуль совместного использования устройств последовательного вывода обеспечивает ведение очереди файлов, выводимых на устройство, просмотр этой очереди и управление ею с удаленных машин);
 - доступ к периферийным устройствам рабочей машины с удаленных машин;
 - организация концентратора второго уровня (только для PM2).
- Сетевое программное обеспечение КОНЦ1, КОНЦ2 выполняется под управлением FB-монитора операционной системы RT-11. В зависимости от реализуемых функций оно занимает от 3 до 6 Кслов оперативной памяти. Минимальный объем памяти, необходимый для функционирования рабочей машины как пользователя сети, под RT-11 составляет около 700 слов.

В качестве рабочих машин системы ОРТ могут выступать мини- или микро-ЭВМ, работающие под операционной системой RSX11M (Электроника-100/25, СМ-4, МС1211). На абонентской машине под RSX11M на сегодня реализованы следующие функции:

- взаимодействие с дисковым архивом комплекса РАДИУС;
- терминальный доступ в систему RSX11M с удаленных терминалов сети;
- организация логического канала передачи данных между процессами в различных машинах, работающих под RSX11M (межзадачный обмен под RT-11 реализован в рамках протокола управления удаленным устройством, с использованием стан-

дартного драйвера межзадачной связи MQ).

Среди абонентов сети есть и несколько микро-ЭВМ ОДРЕНОК [11]. Они используют систему ОРТ для доступа к дисковому архиву файлов.

Развитие системного программного обеспечения ОРТа идет по следующим направлениям: расширение номенклатуры виртуальных устройств (в частности, реализуется возможность использования удаленных магнитных лент); предоставление пользователю дополнительных сервисных возможностей (сетевая информационная служба, электронная почта); включение в состав сети других операционных систем (TSX, UNIX).

ОРГАНИЗАЦИЯ ПРОТОКОЛОВ СИСТЕМЫ «ОРТ»

Как уже упоминалось выше, протоколы системы ОРТ «генетически» произошли из протоколов РАДИУСа. Немногочисленные различия продиктованы, во-первых, использованием дополнительных возможностей системы связи ДОЗА (индикация готовности передатчика и наличие потенциального сигнала «запрос»), во-вторых, необходимостью функционального расширения протокола.

В структуре связных протоколов комплекса РАДИУС традиционно выделяются три уровня: линейный, транспортный и пользовательский [6]. На рисунке 2 показано соответствие между этими уровнями протокола и семиуровневой эталонной моделью взаимодействия открытых систем [5].

Прикладной уровень	Пользовательские протоколы высокого уровня
Уровень представления	
Сеансовый уровень	
Транспортный уровень	Транспортный протокол
Сетевой уровень	Линейный протокол
Канальный уровень	
Физический уровень	Система связи ДОЗА

Рис. 2.

Единицей обмена на линейном уровне выступает сообщение,

рассматриваемое как последовательность байтов. Хотя в некоторых из протоколов высокого уровня и производится разбиение данных запроса на пакеты ограниченной длины для последовательной передачи через сеть, коммутации пакетов протоколы ОРТа не предусматривают. «Слоистая» структура сообщения, отражающая иерархию протоколов, изображена на рисунке 3.

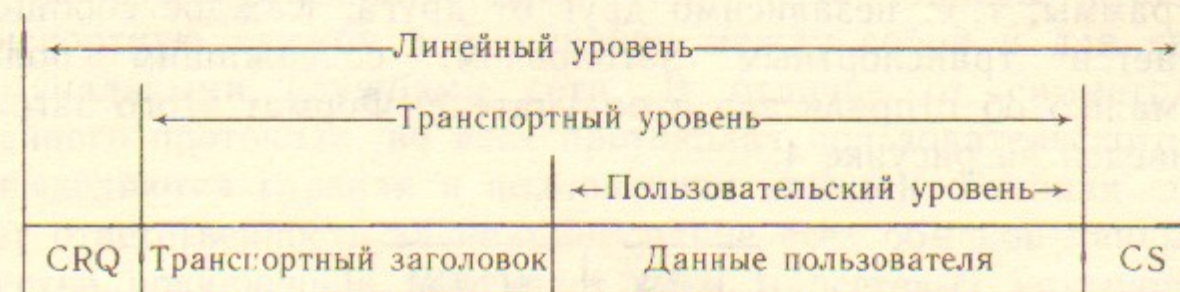


Рис. 3. Пользовательская и служебная информация в структуре сообщения (CRQ — команда «Запрос связи», CS — контрольная сумма сообщения).

Для некоторых подсистем в протоколах пользовательского уровня выделяются дополнительные уровни. Так, при работе с дисковым АРХИВОМ данные пользователя дополняются обязательной служебной информацией, называемой «заголовком пользователя» и содержащей параметры затребованной архивной функции.

Функция линейного уровня — безошибочная передача сообщения по каналу, соединяющему две машины. Для этого последовательность байтов сообщения обрамляется синхронизирующими командами и снабжается контрольной суммой. В случае несовпадения полученной контрольной суммы и подсчитанной суммы принятых байтов приемник запрашивает повторную передачу сообщения. Физически канал, соединяющий машины, представляет собой две взаимносимметричные линии, совместно обеспечивающие дуплексную передачу данных. Однако реализованный протокол линейного уровня является полудуплексным. В этом протоколе для синхронизации передачи и подтверждения правильности принятого блока используется линия, направленная от приемника к передатчику. Большинство из используемых интерфейсов являются байтовыми, поэтому протокол ориентирован на передачу сообщения по байтам. Для варианта связи (ППИ8-АДСП1) предусмотрена возможность использования словной передачи. Единственной коллизией линейного протокола является ситуация встречных запросов, т. е. ситуация, в которой две машины одновременно пытаются передать данные друг другу. В этом случае машина, расположен-

ная выше на дереве системы, считается более приоритетной и первой передает свое сообщение. Подробности линейного протокола приведены в Приложении.

Транспортный уровень отвечает за маршрутизацию сообщений и мультиплексирование, т. е. предоставление возможности нескольким процессам одновременно использовать один канал или ряд последовательных каналов. Сообщения передаются по сети как дейтаграммы, т. е. независимо друг от друга. Каждое сообщение снабжается транспортным заголовком, содержащим полную информацию об отправителе и получателе. Формат этого заголовка приведен на рисунке 4.

0	H.FNC	H.NUM
2	H.LNG	
4	H.RSV	
6	H.SND	
10	H.ADD	

Рис. 4. Формат транспортного заголовка.

Поля транспортного заголовка имеют следующий смысл:
 H.FNC — тип и функция пользовательского протокола (байт),
 H.NUM — номер логического канала (байт),
 H.LNG — длина данных пользователя в словах (2 байта),
 H.RSV — сетевой адрес отправителя (2 байта),
 H.SND — сетевой адрес получателя (2 байта),
 H.ADD — дополнительная информация, зависящая от функции.

Сетевой адрес состоит из двух однобайтовых полей — номера машины и номера процесса (для абонентских машин третьего уровня номер машины занимает старшие 12 разрядов сетевого адреса). Процесс, отправляющий запрос, может не знать номера машины, на которой он исполняется. Поле номера машины-отправителя заполняется концентратором, принимающим сообщение, что гарантирует правильную доставку ответа. Определяемый полем H.NUM логический канал является средством адресации нескольких портов внутри адресуемого сетевого процесса. Установление такого логического канала происходит на уровне пользовательско-

го протокола и никак не меняет дейтаграммный характер распространения сообщения. На древоподобной сети существует единственный путь, соединяющий два заданных узла, поэтому направление дальнейшей передачи сообщения определяется в каждом узле однозначно.

В класс протоколов «высокого» уровня попадают протоколы всех пользовательских сетевых утилит. Эти утилиты используют транспортную службу для общения между собой и для связи с коммунальными службами сети. В отличие от симметричного линейного протокола, во всех протоколах пользовательского уровня выделяются главная и подчиненная станции. Главная станция несет ответственность за инициирование всех обменов данными. В качестве подчиненной выступает одна из сетевых коммунальных служб, вызываемая процессами в абонентских машинах. Сообщения, которыми обмениваются станции, делятся на запросы и ответы. Запросы передаются от главной станции к подчиненной. Ответы представляют собой реакцию на запросы и передаются в обратном направлении. Вместо ответа процессу может быть возвращен транспортный отказ, генерируемый транспортной службой и указывающий на недоступность машины-адресата или отсутствие в указанной машине адресуемого процесса.

В Приложении приведены типы пользовательских протоколов и исполняемые ими функции. Там же подробно описан один из основных в системе ОРТ пользовательских протоколов — протокол управления удаленным устройством.

ЗАКЛЮЧЕНИЕ

Система ОРТ находится в эксплуатации три года. Все это время развивались и совершенствовались как аппаратная база системы, так и состав ее программного обеспечения. Препринт описывает состояние на начало 1988 года. Опыт эксплуатации системы показал возможность эффективной организации низкоскоростной локальной сети на базе простых байтовых интерфейсов. В настоящее время ОРТ объединяет около 150 абонентских мини- и микро-ЭВМ.

При использовании интерфейсов системы связи ДОЗА (базой для ОРТа) физическая скорость передачи в линии составляет 40 Кбайт/с, однако программное управление обменом снижает ее до 20 Кбайт/с (PM — процессор M2 с тактовой частотой 400 нс). Реальная скорость чтения винчестерского диска ST225 смежного

концентратора составляет 9 Кбайт/с, чтения псевдодиска 15.5 Кбайт/с (при чтении псевдодиска данные не буферизируются в оперативной памяти). При одновременной активности нескольких рабочих машин обмен с удаленным диском замедляется практически линейно.

ЛИТЕРАТУРА

1. Шувалов Б.Н. Организация многомашинного комплекса, ориентированного на обслуживание рабочих мест физика-экспериментатора. — В сб.: II Всесоюзный семинар по автоматизации науч. исследований в яд. физике и смеж. областях. Пленарные доклады. Новосибирск, 1982.
2. Бейлин М.В., Вьюшин О.В., Клименко А.Д. и др. Системное матобеспечение многомашинного комплекса автоматизации физического эксперимента. — Препринт ИЯФ 82-72, Новосибирск, 1982.
3. Дж. Дей, Ю. Зиммерман. Эталонная модель взаимосвязи открытых систем. — ТИИЭР, т.71, 12, 1983.
4. Шувалов Б.Н., Яснев М.В. Развитие многомашинного комплекса «РАДИУС» на машинах серии «Электроника». — В сб.: Материалы III Всесоюзного семинара по автоматизации исследований в яд. физике и смеж. областях. Тбилиси, 1984.
5. Большевский А.Б. Программное обеспечение системы ТЕВУС. — В сб.: Автоматизированные системы научных исследований, обучения и управления в вузах. Новосибирск, 1985.
6. Сысолетин Б.Л. Специализированная операционная система ОС/Р-5. — Препринт ИЯФ 86-111, Новосибирск, 1986.
7. Неханевич Э.Л., Яснев М.В. Система связи для малой локальной сети. — Препринт ИЯФ, Новосибирск, 1988 (в печати).
8. Банзаров В.Ш., Шувалов Б.Н., Яснев М.В. Электронный псевдодиск для микро-ЭВМ Электроника-60. — Препринт ИЯФ 85-106, Новосибирск, 1985.
9. Кочев А.А., Путьмаков А.Н., Зеленцов А.Г. Аппаратура связи для терминального класса. Республиканское совещание-семинар: Использование ЭВМ в учебной и научно-исследовательской работе студентов. Новосибирск, НГУ, 26—28 янв. 1988 г.
10. Козак В.Р., Тютюник А.Г., Уваров Н.П. Автономный контроллер крейта 10615. — Препринт ИЯФ 88-31, Новосибирск, 1988.
11. Пискунов Г.С. и др. Двадцатичетырехразрядная ЭВМ в стандарте КАМАК. — Автометрия, 1986, т.4.

ПРИЛОЖЕНИЯ

1. Линейный протокол системы ОРТ

Блок-схемы на рисунках 5—8 изображают различные фазы передачи сообщения по физическому каналу. Описание основной применяемой системы связи (ДОЗА) и ее интерфейсов приведено в [7]. На рис. 6 приведен алгоритм синхронизации передатчика с выявлением ситуации встречных запросов. Процедура передачи

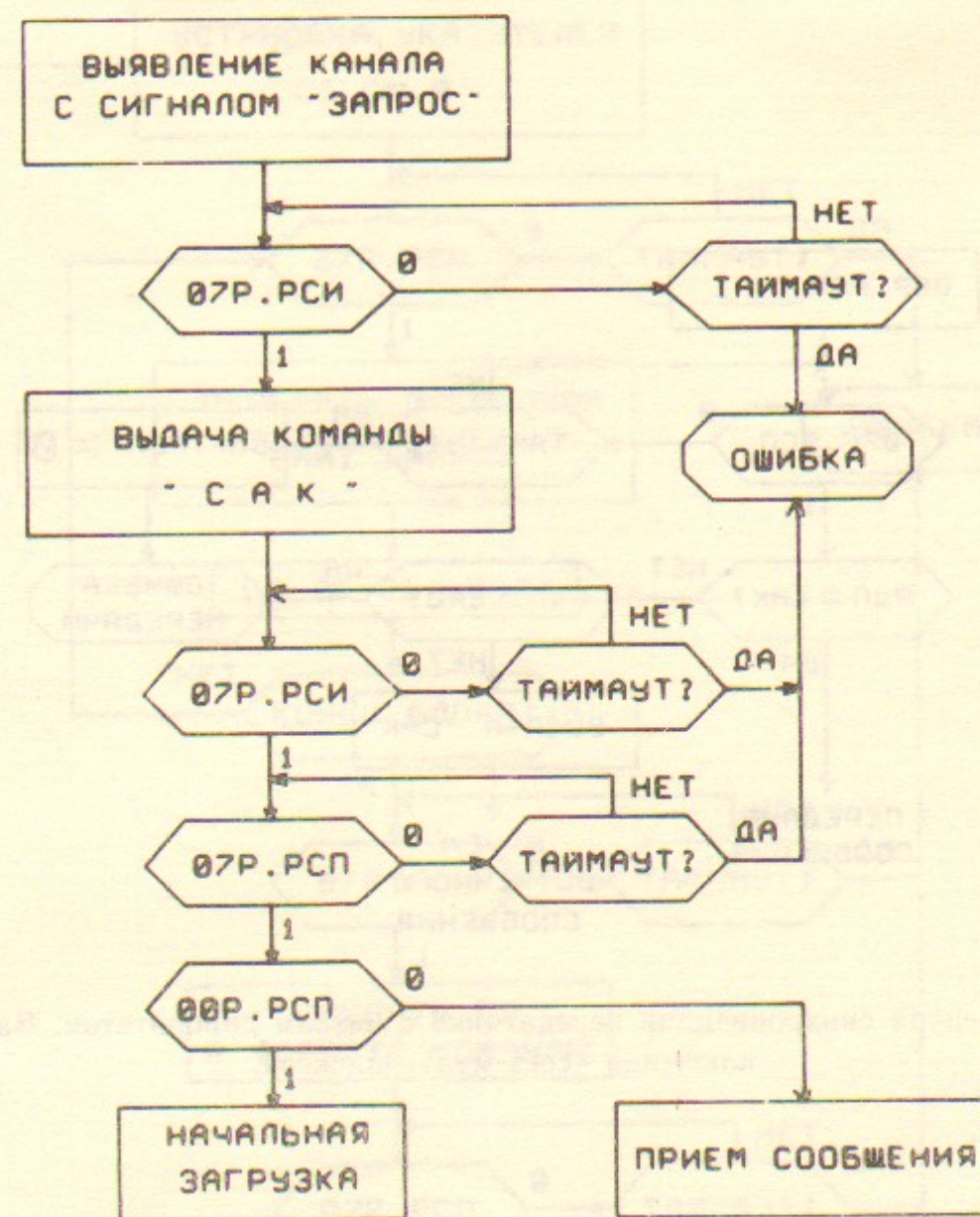


Рис. 5. Идентификация запроса начальной загрузки на мультиплексоре.

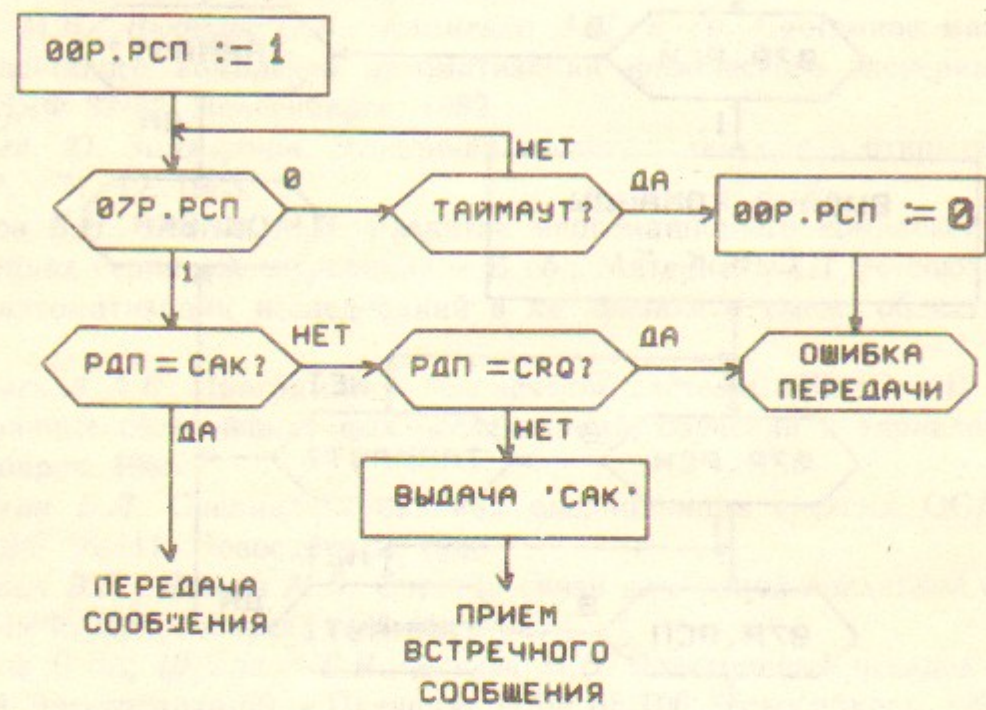


Рис. 6. Процедура синхронизации передатчика с низким приоритетом. Вариант подключения через мультиплексор.

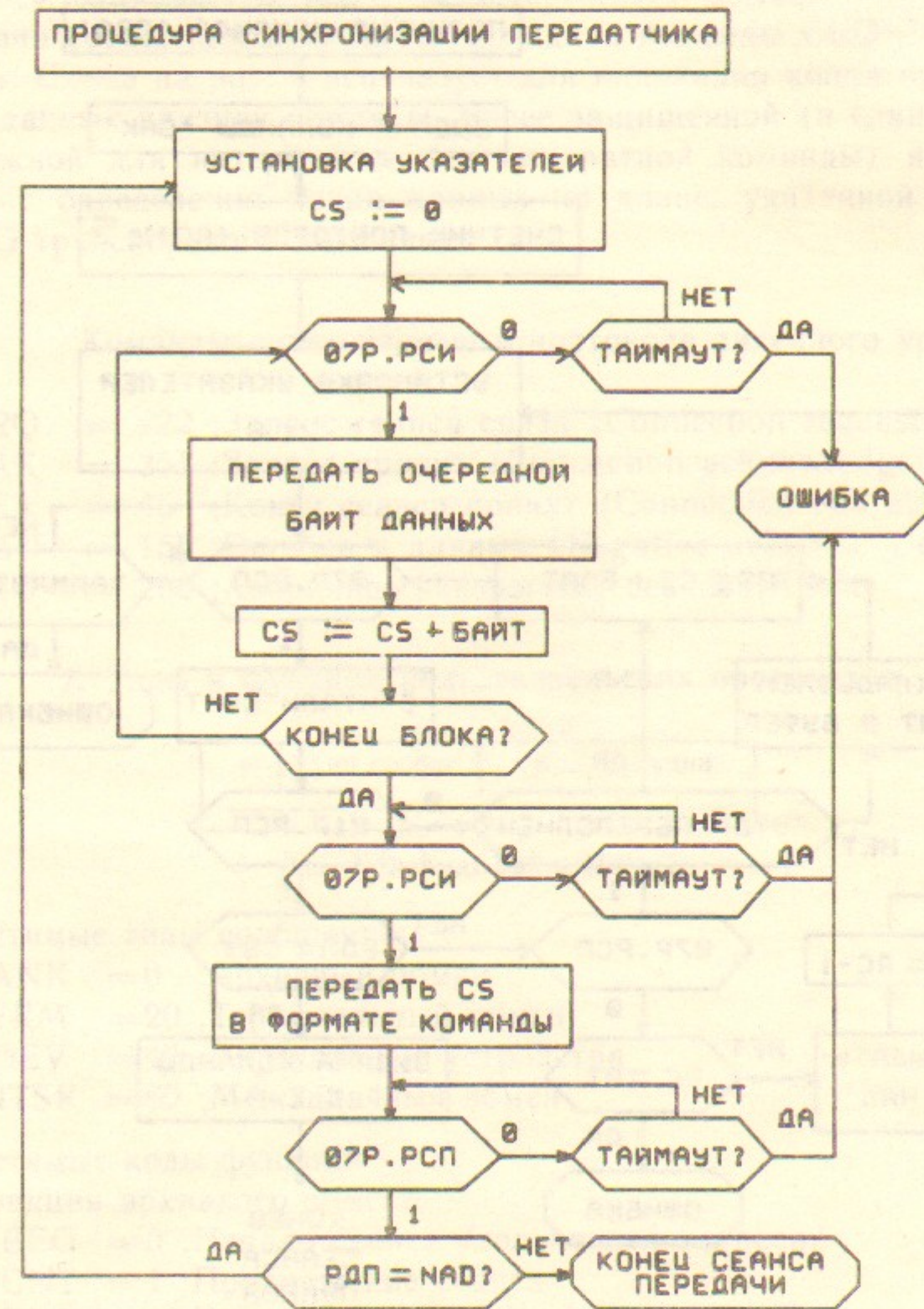


Рис. 7. Алгоритм передачи сообщения.

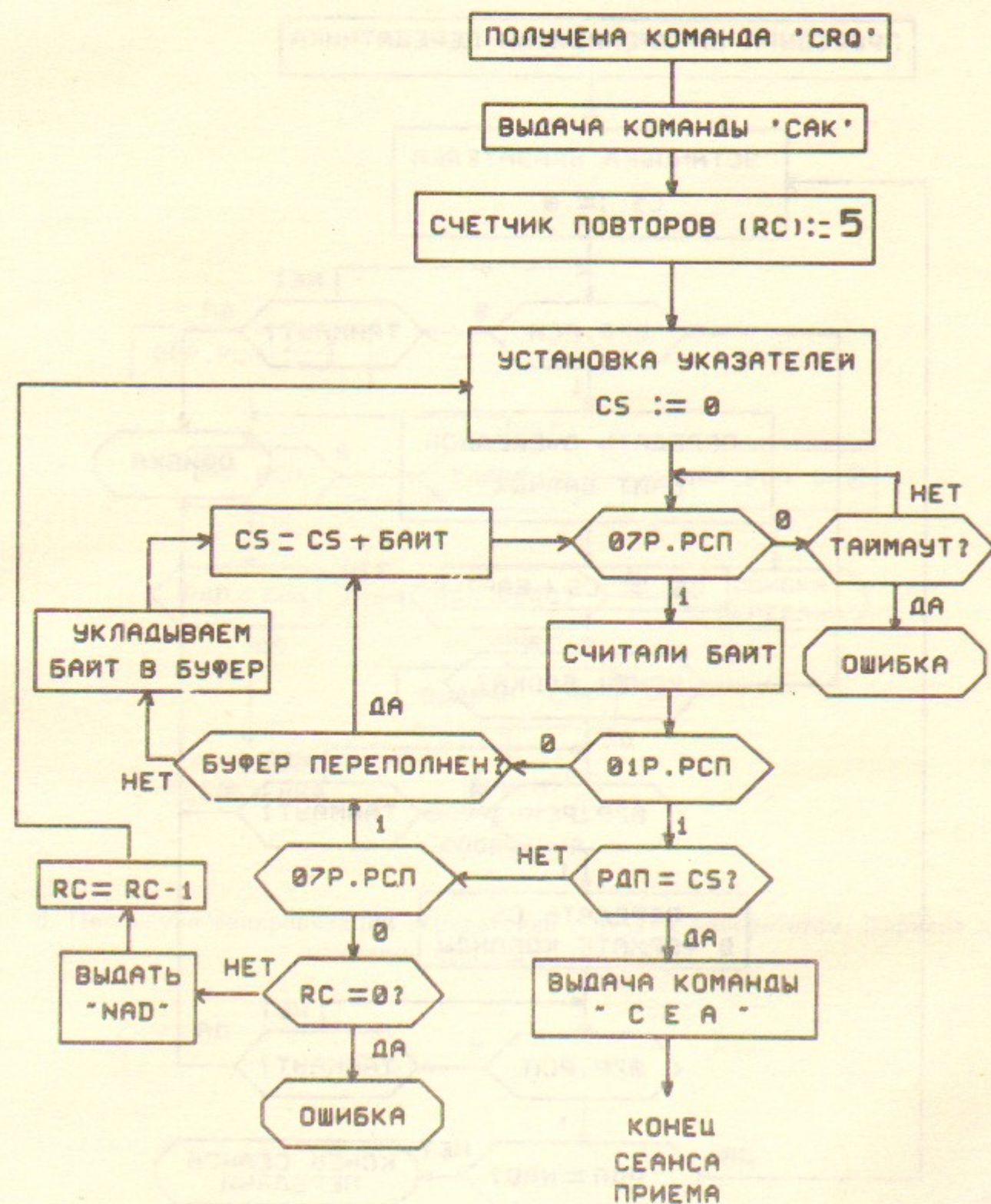


Рис. 8. Алгоритм приема сообщения.

сообщения начинается с выставления потенциального сигнала «запрос». Для интерфейсов, на которых этот сигнал отсутствует, передача сообщения начинается с выдачи команды CRQ — «запрос связи». Схема на рис. 8 использует для индикации конца принимаемых данных признак команды. Более защищенной (и единственно возможной для интерфейсов без аппаратной команды) является вариант определения конца данных по длине, указанной в поле H.LNG транспортного заголовка.

Команды, применяемые в протоколе линейного уровня

- CRQ = 222 ; Запрос сеанса связи (Connection request)
- САК = 351 ; Запрос принят (Connection acknowledgement)
- СЕА = 45 ; Конец сеанса принят (Connection end acknowl.)
- NAD = 155 ; Ошибка в данных (Negative acknowl. - data)
- АВО = 266 ; Аварийное завершение сеанса (Abort)

2. Типы и функции пользовательских протоколов

—	Тип сообщ.	Код функции
—	—	—

Рис. 9. Формат байта H.FNC.

Допустимые типы сообщений:

- BANK = 0 ; Архивный обмен
- TERM = 20 ; Терминальный обмен
- VDEV = 40 ; Виртуальные устройства
- INTSK = 60 ; Межзадачный обмен

Допустимые коды функций:

- 1) Функции архивного обмена
 - V.BEG = 0 ; Начало сеанса /только для запроса/
 - V.CNT = 1 ; Продолжение сеанса
 - V.END = 2 ; Конец сеанса /только для ответа/
- 2) Функции терминального обмена
 - T.CVT = 0 ; Занять вирт.терминал (Connect virtual term.)
 - T.RVT = 1 ; Ответ на T.CVT (Reply to connect term.)
 - ; ... H.SND=0 — нет свободных VT
 - T.DVT = 2 ; Освободить терминал (Disconnect virt. term.)

T.VTI =3 ;Байт с терминала (Terminal input)
 T.VTO =4 ;Байт на терминал (Terminal output)
 T.VTR =5 ;Готов от экрана терминала (Terminal ready)
 T.LPO =6 ;Байт на печать (Lineprinter output)
 T.LPR =7 ;Готов от печати (Lineprinter ready)
 T.OLN =12 ;Строка на терминал (Output line)
 T.LRD =13 ;Строковая готовность экрана (Line ready)
 T.BVT =14 ;Разрыв терм.соединения со стороны
 ; ... удаленного терминала (Break VT)

3) Функции управления виртуальным устройством

V.OKK =0 ;Успешное завершение запроса
 V.EOF =1 ;Ошибка — конец файла
 V.ERR =2 ;Аппаратная ошибка
 V.CON =3 ;Установление виртуального соединения
 V.DIS =4 ;Разрыв виртуального соединения
 V.SIZ =5 ;Запрос длины виртуального диска

4) Функции межзадачного обмена

I.RDD =0 ;Чтение (Read data)
 I.WRD =1 ;Запись (Write data)
 I.CNT =2 ;Установление логической линии (Connect task)
 I.RCT =3 ;Ответ на I.CNT (Reply to connect task)
 I.DCT =4 ;Разрыв логической линии (Disconnect task)
 I.REJ =10 ;Отказ на чтение/запись (Reject)

;+
 ; Значения байта H.NUM при H.FNC=I.RCT
 ; =0 — запрос связи принят, соединение установлено
 ; <0 — запрос связи отвергнут, H.NUM — код отказа

;—
 C.REJ = -1 ;Отказ задачи от соединения
 C.TNS = -2 ;Задача не установлена (Task not installed)

;+
 ; Коды отказов (байт H.NUM при H.FNC=I.REJ)

;—
 RJ.LNG =1 ;Превышение кредита (Длина записи больше
 ; ... длины чтения)
 RJ.CNF =2 ;Конфликт (Встречные чтения)

Для всех типов сообщений код функции 17 означает транспортный отказ.

3. Протокол управления удаленным устройством

Для каждой из описываемых функций протокола приводится описание содержательных полей транспортного заголовка (H.XXX) и данных (слова D.1, D.2, ...). Указанные в описании значения — восьмеричные. Значения остальных полей несущественны.

Функция: Установление виртуального соединения с удаленным устройством или файлом на удаленном устройстве

Запрос:

H.NUM (байт) = тип доступа к устройству
 H.FNC (байт) = 43 — VDEV!V.CON
 H.LNG = 4 — длина данных запроса
 H.RSV (байт) = 3 — номер подсистемы-адресата
 H.RSV + 1 (байт) = номер машины-адресата
 H.SND (байт) = 0 — номер процесса — инициатора запроса
 H.ADD = пароль устройства

Тип доступа к устройству: 0 — R/O (Read only), 1 — R/W (Read/Write), 2 — R/W прикрепление к занятому другой машиной устройству или файлу. Значение поля H.RSV + 1, равное нулю, используется для адресации запроса к смежному концентратору.

D.1 = имя устройства (RAD50)
 D.2 = имя файла 1-3 (RAD50)
 D.3 = имя файла 4-6 (RAD50)
 D.4 = тип файла (RAD50)

Ответ:

H.LNG = 4 — длина данных ответа
 H.RSV (байт) = 0 — номер процесса — инициатора запроса
 H.RSV + 1 (байт) = номер машины — инициатора запроса
 H.SND (байт) = 3 — номер процесса-ответчика
 H.SND + 1 (байт) = номер машины-ответчика

D.1 = если > 0 — номер канала, выделенного для доступа к устройству, < 0 — отказ (-1 — нет такого устройства; -2 — указанный файл не найден; -3 — нет свободного канала, переполнение системных таблиц на адресуемой машине; -4 — ошибка при чтении каталога устройства; -5 — устройство или файл уже прикреплены к другой машине; -6 — файл не RT-том; -7 — неверен пароль).

D.2 = номер машины, прикрепившей к себе устройство или файл (для кода отказа = -5).

Указание имени файла имеет смысл только для устройств с непосредственным (адресуемым) доступом. Для устройств с последовательным доступом невозможны отказы (-4, -6, -7).

Функция: Разрыв виртуального соединения с удаленным устройством или файлом

Запрос:

H.NUM (байт) = номер канала
H.FNC (байт) = 44 - VDEV!V.DIS
H.LNG = 0 - длина данных запроса
H.RSV (байт) = 3 - номер подсистемы-адресата
H.RSV + 1 (байт) = номер машины-адресата
H.SND (байт) = 0 - номер процесса - инициатора запроса

Ответ:

H.NUM (байт) = номер канала
H.FNC (байт) = 40 - VDEV!V.OKK
H.LNG = 0 - длина данных ответа
H.RSV (байт) = 0 - номер процесса - инициатора запроса
H.RSV + 1 (байт) = номер машины - инициатора запроса
H.SND (байт) = 3 - номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

Функция: Запрос размера устройства или файла

Запрос:

H.NUM (байт) = номер канала
H.FNC (байт) = 45 - VDEV!V.SIZ
H.LNG = 0 - длина данных запроса
H.RSV (байт) = 4 - номер подсистемы-адресата
H.RSV + 1 (байт) = номер машины-адресата
H.SND (байт) = 0 - номер процесса - инициатора запроса

Ответ:

H.FNC (байт) = 40 - VDEV!V.OKK
H.LNG = 1 - длина данных ответа
H.RSV (байт) = 0 - номер процесса - инициатора запроса
H.RSV + 1 (байт) = номер машины - инициатора запроса
H.SND (байт) = 4 - номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

D.1 = размер устройства (или файла) в блоках по 512. байтов.

Функция: Чтение удаленного устройства

Запрос:

H.NUM (байт) = номер канала
H.FNC (байт) = 40 - VDEV!V.OKK
H.LNG = <длина чтения в словах>!100000
H.RSV (байт) = 1 - номер подсистемы-адресата
H.RSV + 1 (байт) = номер машины-адресата
H.SND (байт) = 0 - номер процесса - инициатора запроса
H.ADD = адрес диска (номер блока - 0, 1, 2 ...)

Ответ:

H.FNC (байт) = 40 - VDEV!V.OKK
H.LNG = длина считанных данных в словах
H.RSV (байт) = 0 - номер процесса - инициатора запроса
H.RSV + 1 (байт) = номер машины - инициатора запроса
H.SND (байт) = 1 - номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

D.1, D.2, D.3 ... - считанные данные

Отказы:

H.FNC (байт) = 41 - VDEV!V.EOF (неверный адрес уст-ва)
H.LNG = 0
H.RSV (байт) = 0 - номер процесса - инициатора запроса
H.RSV + 1 (байт) = номер машины - инициатора запроса
H.SND (байт) = 1 - номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

H.FNC (байт) = 42 - VDEV!V.ERR (аппаратная ошибка)
H.LNG = 0
H.RSV (байт) = 0 - номер процесса - инициатора запроса
H.RSV + 1 (байт) = номер машины - инициатора запроса
H.SND (байт) = 1 - номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

Функция: Запись удаленного устройства

Запрос:

H.NUM (байт) = номер канала
H.FNC (байт) = 40 - VDEV!V.OKK
H.LNG = длина данных в словах

H.RSV (байт) = 1 — номер подсистемы-адресата
H.RSV + 1 (байт) = номер машины-адресата
H.SND (байт) = 0 — номер процесса — инициатора запроса
H.ADD = адрес диска (номер блока — 0, 1, 2 ...)

D.1, D.2, D.3 ... — записываемые данные

Ответ:

H.FNC (байт) = 40 — VDEV!V.OKK
H.LNG = <длина записанных данных в словах>!100000
H.RSV (байт) = номер процесса — инициатора запроса
H.RSV + 1 (байт) = 0 — номер машины — инициатора запроса
H.SND (байт) = 1 — номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

Отказы:

H.FNC (байт) = 41 — VDEV!V.EOF (неверный адрес уст-ва)
H.LNG = 0
H.RSV (байт) = 0 — номер процесса — инициатора запроса
H.RSV + 1 (байт) = номер машины — инициатора запроса
H.SND (байт) = 1 — номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

H.FNC (байт) = 42 — VDEV!V.ERR (аппаратная ошибка)
H.LNG = 0

H.RSV (байт) = 0 — номер процесса — инициатора запроса
H.RSV + 1 (байт) = номер машины — инициатора запроса
H.SND (байт) = 1 — номер процесса-ответчика
H.SND + 1 (байт) = номер машины-ответчика

Код отказа V.ERR означает либо аппаратную ошибку при записи, либо попытку записи на устройство, прикрепленное только для чтения. Длина данных, считываемых или записываемых за один запрос, не должна превышать 1К байт.

Э.Л. Неханевич, М.В. Яснев

Система ОРТ-локальная сеть мини- и микро-ЭВМ

Ответственный за выпуск С.Г. Попов

Работа поступила 11 июля 1988 г.
Подписано в печать 29.08. 1988 г. МН 08449
Формат бумаги 60×90 1/16 Объем 1,1 печ.л., 0,9 уч.-изд.л.
Тираж 290 экз. Бесплатно. Заказ № 112

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