

12/III-84



**сообщения
объединенного
института
ядерных
исследований
дубна**

1388/84

P10-83-827

А.В.Алфименков, Р.Вебер, В.М.Северьянов

**ДРАЙВЕР В ОПЕРАЦИОННОЙ СИСТЕМЕ IAS,
РЕАЛИЗУЮЩИЙ ПРОТОКОЛ
ИНФОРМАЦИОННОГО КАНАЛА ASTRA**

1983

1. КРАТКОЕ ОПИСАНИЕ ПРОТОКОЛА ASTRA

Протокол ASTRA - это байториентированный протокол информационного канала, отвечающий второму уровню иерархии сетевых протоколов Международной организации стандартов^{1/} и предназначенный для работы прежде всего с асинхронными последовательными линиями связи типа точка-точка, причем, как в дуплексном, так и в полудуплексном режимах. Протокол был разработан для локальной сети IBM SONET^{2/}, создаваемой в рамках измерительного центра Лаборатории нейтронной физики Объединенного института ядерных исследований^{3/}, однако его можно использовать и вне иерархии протоколов этой сети. При разработке учитывался опыт аналогичных протоколов DDCMP^{4/} и HDLC^{7,9/}.

Работа протокола организуется с помощью обмена кадрами, которые бывают двух типов: информационные /или кадры данных/ и управляющие /или командные/. Каждый кадр имеет четырехбайтовый заголовок и двухбайтовый хвостовик контрольной суммы. В информационных кадрах между заголовком и хвостовиком располагается поле данных произвольной длины. Длина поля данных передается в заголовке кадра, за счет этого достигается прозрачность информационного канала.

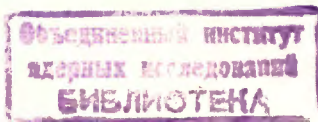
В заголовке кадра данных передается также номер этого кадра в потоке кадров и номер, подтверждающий правильность приема полученных кадров данных. Номера меняются циклически по модулю 8. Управляющие кадры не нумеруются.

На способ формирования контрольной суммы жестких требований не накладывается. Исправление обнаруженных ошибок производится с помощью повторной передачи кадров, полученных с ошибками. Для исключения "зависаний" канала используются сторожевые временные интервалы /или тайм-ауты/.

Протокол выполняет покадровую синхронизацию /побитовая и побайтовая синхронизация осуществляется аппаратно/, для чего используется посылка перед отправляемыми кадрами специальной синхропоследовательности, состоящей из байтов, полностью заполненных единицами.

В настоящее время в протоколе ASTRA используется 5 команд /или 5 типов управляющих кадров/:

1. START - запрос на соединение;
2. RESET - подтверждение соединения;
3. ACKNOWLEDGE - подтверждение правильного приема кадров данных;



- 4. REJECT - отказ от кадра или сообщение о потере ответа;
- 5. STOP - разъединение.

Более подробное описание протокола ASTRA дается отдельной публикацией.

К настоящему времени протокол реализован для последовательных асинхронных интерфейсов типа DL-11/10/ в операционных системах IAS и RT-11; в данной публикации рассматривается его реализация в IAS, выполненная в виде драйвера как часть программного обеспечения сети SONET.

2. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА В ОПЕРАЦИОННОЙ СИСТЕМЕ IAS

В операционной системе IAS/5/ подготовленная к выполнению программа /т.е. оттранслированная и построенная с помощью редактора связей/ называется задачей. Любая задача, желающая осуществить какую-либо операцию ввода-вывода, выдает системную макрокоманду QIO\$, которая означает буквально следующее: "В очередь к указанному периферийному устройству поставить запрос на указанную операцию ввода-вывода". Получив такую макрокоманду, супервизор операционной системы формирует элемент очереди, ставит его /в соответствии с приоритетом/ в очередь запросов ввода-вывода к драйверу указанного устройства и взводит флаг системного события, сообщаящий драйверу о том, что в его очереди появился новый запрос. Сам супервизор запросы на ввод-вывод никак не интерпретирует, а только передает их драйверу, т.е. задаче, управляющей определенным периферийным устройством.

Драйверы в операционной системе IAS близки к обычным задачам, но имеют дополнительные особенности: во-первых, как правило, они содержат подпрограммы обработки аппаратных прерываний, во-вторых, им разрешен доступ к привилегированным областям памяти.

После загрузки в память драйвер выполняет некоторую программу инициализации, после чего переходит в "бесконечный" цикл ожидания запросов. Запросы подразделяются на две категории: нормальные /т.е. собственно запросы на операции ввода-вывода, выдаваемые задачами/ и срочные /выдаваемые супервизором, например, для досрочного завершения операции или приостановки работы и выгрузки драйвера/. Получив запрос, драйвер выполняет его, после чего возвращается в цикл ожидания запросов /если, конечно, это не был срочный запрос на выгрузку самого драйвера/.

Задача, выдавшая запрос на ввод-вывод, как правило, получает управление обратно сразу же, как только запрос будет поставлен в очередь к драйверу. Однако она имеет возможность приостановить свое выполнение до завершения запрошенной операции ввода-вывода, ожидая установления флага события, или продолжать свою работу, но быть готовой к обработке асинхронного системного прерывания в момент завершения операции. Информацию о возможных ошибках

устройства драйвер передает задаче-заказчику через так называемый статусный блок /или блок состояний/ ввода-вывода, находящийся в адресном пространстве этой задачи.

В задаче драйвера периферийного устройства содержится 5 основных секций/6/:

1. Табличная область для связи с системными подпрограммами.
2. Код инициализации для однократного выполнения после загрузки.
3. Код для извлечения из очереди и исполнения запросов ввода-вывода.
4. Подпрограммы обработки прерываний /ПОП/.
5. Секция восстановления после сброса питания.

Взаимодействие с системными подпрограммами, которые предоставляют драйверам различного рода услуги, осуществляется через две таблицы: таблицу идентификации устройства, используемую для извлечения запросов из очереди, и таблицу диспетчеризации запросов.

Основные функции кода инициализации таковы:

- объявить задачу драйвера резидентной в оперативной памяти;
- подключиться к прерываниям;
- установить необходимую информацию для сбора статистики об ошибках;
- выполнить инициализацию, специфическую для данного класса устройств;
- установить системное асинхронное прерывание для восстановления после сброса питания.

Функции основного кода драйвера:

- ожидание завершения текущей операции или поступления в очередь нового запроса;
- выполнение необходимых действий по завершении операции ввода-вывода;
- попытка извлечь из очереди следующий запрос;
- если запрос извлечен, то проверка его правильности;
- диспетчеризация запроса на соответствующую подпрограмму;
- приостановка работы драйвера при получении срочного запроса на выгрузку.

3. ФУНКЦИОНАЛЬНАЯ СХЕМА ДРАЙВЕРА ASTRA

Функциональная схема драйвера, реализующего протокол ASTRA в операционной системе IAS, показана на рис.1. Чтобы не загромождать схему, на ней не изображены структуры, упомянутые в предыдущем разделе и присущие всем драйверам в IAS /код инициализации, таблицы идентификации и диспетчеризации, секция восстановления после сброса питания/. Основными функциональными элементами драйвера ASTRA являются:

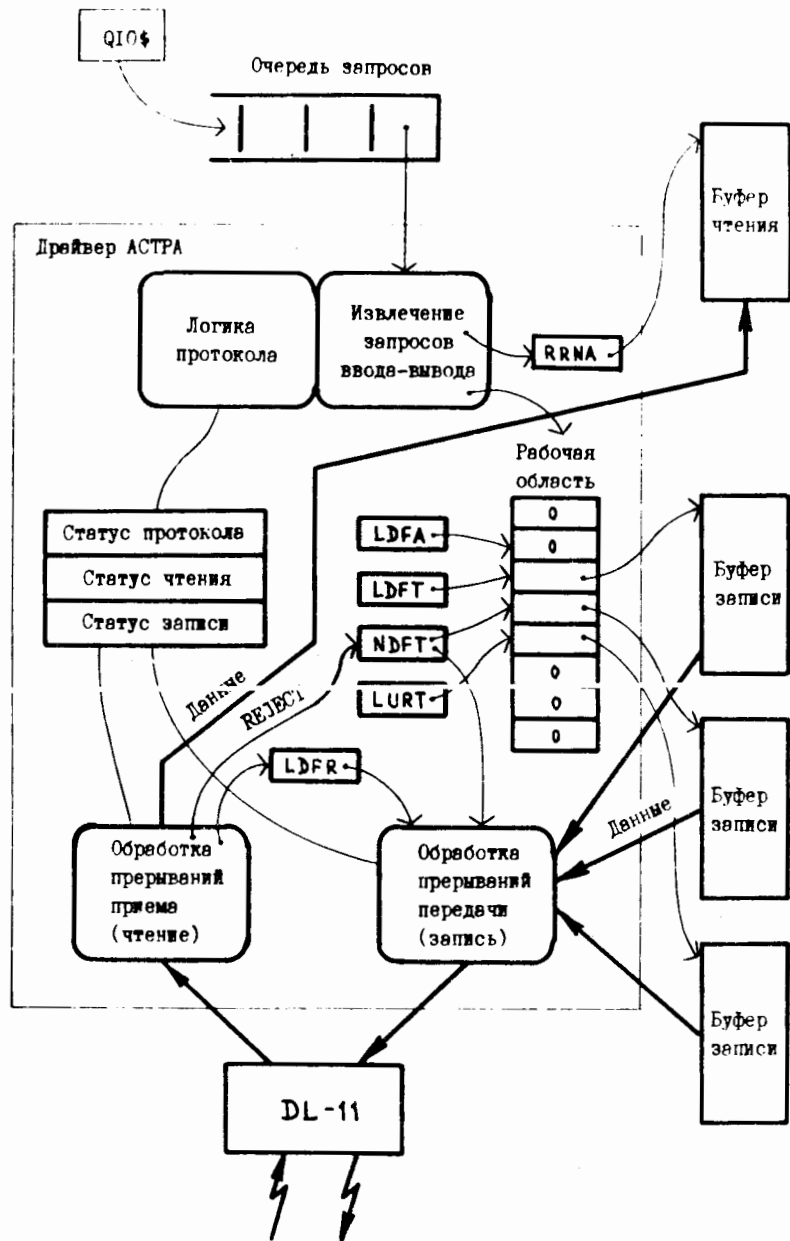


Рис.1. Функциональная схема драйвера ASTRA.

- программа извлечения запросов ввода-вывода;
- программа выполнения логики протокола;
- подпрограмма обработки прерываний приема /чтения/;
- подпрограмма обработки прерываний передачи /записи/;

Для взаимодействия между этими функциональными элементами /которые, по сути дела, представляют собой асинхронно и параллельно исполняемые процессы/ используются следующие структуры данных:

- три статусных слова /общий статус протокола, статус чтения и статус записи/;
- рабочая область для сохранения адресов для кадров, требующих подтверждения о правильном приеме;
- LURT - номер последнего извлеченного из очереди запроса на передачу;
- LDFR - номер последнего принятого кадра данных;
- LDFA - номер последнего подтвержденного кадра;
- NDFT - номер следующего кадра данных, подлежащего отправке в канал;
- LDFT - номер последнего переданного кадра данных.

Статусное слово протокола содержит общую информацию о состоянии канала, результатах извлечения запросов из очереди, операциях, находящихся в процессе выполнения или тех, которые предстоит выполнить на следующем этапе работы протокола. Через это статусное слово собственно и осуществляется диспетчеризация запросов, так как подпрограммы обработки прерываний именно из него узнают о тех операциях, которые им предстоит выполнять. Статусные слова чтения и записи хранят информацию о текущем состоянии подпрограммы обработки соответствующих прерываний /фактически это детальная информация о том, какие элементы формата кадров обрабатываются в данный момент/. Необходимость в статусных словах чтения и записи объясняется тем, что соответствующие подпрограммы обработки прерываний запускаются каждым принимаемым или отправляемым байтом.

Номер LURT увеличивается на единицу /все упомянутые выше номера изменяются циклически по модулю 8/ всякий раз, когда из очереди ввода-вывода извлекается запрос на передачу. Одновременно с этим устанавливается статус "Есть данные для передачи".

Номер каждого приходящего кадра данных должен быть на единицу больше, чем LDFR. Если это так, то LDFR увеличивается, чтобы отразить новый номер последнего принятого кадра данных; если нет, то драйвер отправляет в канал сообщение об ошибке в нумерации кадров, а полученный кадр выбрасывается.

Номер последнего подтвержденного кадра LDFA берется из пришедшего кадра данных /неявное подтверждение/ или из управляющего кадра явного подтверждения. Если выполняется соотношение $LDFA/старый/ < LDFA/новый/ \leq LURT$, то все кадры данных с номерами от $LDFA/старый/ + 1$ до $LDFA/новый/$ считаются принятыми правильно, после чего соответствующие запросы на передачу можно объявить

завершенными. Если LDFA/новый/ < LDFA/старый/, то пришедший из канала номер подтверждения просто игнорируется.

Номер следующего кадра, подлежащего передаче, NDFT, увеличивается на единицу при начале передачи каждого нового кадра данных. При получении из канала команды отказа REJECT этот номер получает значение LDFA/новый/ + 1, так как кадры, не подтвержденные командой отказа, подлежат повторной передаче.

Номер последнего переданного кадра данных LDFT получает значение NDFT до его увеличения.

Драйвер ASTRA буферизации транспортируемых данных не производит, он только формирует информационные и управляющие кадры и хранит лишь информацию, необходимую для его функционирования. Транспортируемые же данные буферизуются в адресном пространстве тех задач, которые являются пользователями драйвера ASTRA.

4. ОБРАБОТКА ОЧЕРЕДИ ЗАПРОСОВ

Для постановки запросов к драйверу ASTRA используются макрокоманды QIO\$ с кодами функций IO.RLB /"Чтение логического блока"/ и IO.WLB /"Запись логического блока"/. Макрокоманда чтения передает драйверу адрес приемного буфера и его максимальную длину, и макрокоманда записи - адрес и фактическую длину поля данных, подлежащего передаче.

В силу специфики иерархии программного обеспечения сети SONET, пользователи драйверов ASTRA являются программой коммутации пакетов, постоянно "слушающая" все обслуживаемые информационные каналы, для чего она раз за разом ставит запросы на чтение всем драйверам ASTRA, с которыми имеет дело. Это наложило некоторый отпечаток на построение драйвера и обработку им запросов. По этой причине и в силу особенностей самого протокола в процессе выполнения может находиться только один запрос на чтение, но может быть несколько /до семи/ запросов на передачу /поскольку циклическая нумерация позволяет посылать в канал несколько кадров данных без ожидания подтверждения на каждый кадр/.

Когда драйвер пытается извлечь очередной запрос из очереди, то он прежде всего проверяет, не обрабатывается ли в данный момент аналогичный запрос. Если да, то новый запрос остается в очереди до завершения текущего, а драйвер пытается извлечь следующий запрос в надежде, что он окажется работой другого типа. Если же новый запрос может быть обслужен, то драйвер передает адрес и длину буфера для соответствующей подпрограммы обработки прерываний и устанавливает отвечающий типу запроса статус /"Есть текущее чтение" или "Есть текущая запись"/.

Завершение обработки запросов на чтение и запись выполняется по-разному. По завершении запроса на чтение задаче, которая его послала, передается состояние по завершении операции, фактическая длина принятого массива данных и, в зависимости от парамет-

ров макрокоманды QIO\$, устанавливается флаг системного события или инициируется асинхронное системное прерывание.

По завершении текущей операции записи, т.е. после отправки кадра данных в канал, соответствующий узел из очереди запросов помещается во внутреннюю очередь кадров, ожидающих подтверждения, что дает возможность осуществлять повторную передачу в канал кадров, принятых с ошибками. В этот момент драйвер готов к извлечению из очереди и выполнению следующего запроса на запись. Пользователю же о завершении запроса на запись сообщается только после получения подтверждения правильного приема его данных, что делается так же, как и в случае завершения запроса на чтение.

5. ПОДПРОГРАММЫ ОБРАБОТКИ ПРЕРЫВАНИЙ

При поступлении из линии связи каждого байта запускается подпрограмма обработки прерываний чтения /ПОПЧ/. Она считывает принятый байт из входного буфера интерфейса DL-11 и обрабатывает в соответствии с информацией в статусном слове чтения, которая говорит о том, является ли принятый байт частью синхропоследовательности или какой части кадра он соответствует. Принимая информационные и управляющие кадры, ПОПЧ подсчитывает контрольную сумму, затем сравнивает ее с пришедшей из канала. При обнаружении ошибок ПОПЧ устанавливает в статусном слове протокола запрос для логики протокола на посылку в канал команды отказа и указывает тип ошибки в отведенном для этого байте. После правильного приема полного пакета ПОПЧ устанавливает статус окончания ввода, затем драйвер может завершить соответствующую заявку на чтение.

Подпрограмма обработки прерываний записи /ПОПЗ/ запускается логикой протокола путем установки бита разрешения прерывания передачи в статусном слове интерфейса. По статусному слову протокола ПОПЗ определяет, что ей надо передавать: первый байт синхропоследовательности или байт начала заголовка кадра. Она запрещает прерывание передачи, пересылает необходимый байт в выходной буфер интерфейса, вновь разрешает прерывание передачи и на этом заканчивается. Этот процесс повторяется для каждого байта кадра. При транспортировке кадра ПОПЗ подсчитывает для него контрольную сумму. Информация о том, какие части должны передаваться при очередном входе в ПОПЗ, находится в статусном слове записи. После передачи всех байтов кадра состояние "Есть текущая запись" в статусном слове протокола сбрасывается, а разрешения прерывания передачи больше не производится.

Описанным выше способом работа ведется в дуплексном режиме. В полудуплексном режиме /например, когда связь организована через полудуплексный модем/ перед выполнением каждой операции вывода проверяется ее возможность в данный момент, т.е. занят ли сейчас

канал вводом или свободен. Если ввод активен, то приходится ждать до его окончания.

6. ЛОГИКА ПРОТОКОЛА

Логика протокола реализуется драйвером в соответствии с графом состояний, изображенным на рис.2. Это ориентированный нагруженный граф, вершины которого представляют состояние протокола

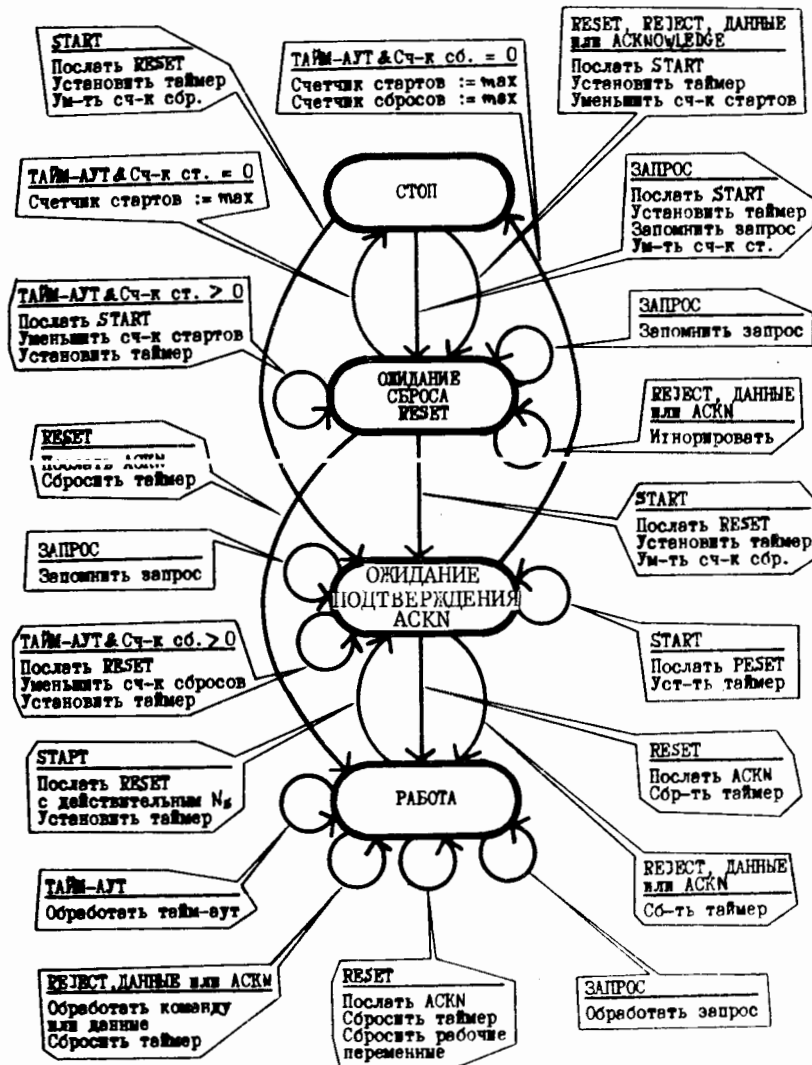


Рис.2. Граф состояний логики протокола.

/Точнее - состояние драйвера, реализующего протокол на одной стороне канала/, а ребра обозначают переходы между состояниями. На ребрах указана причина, вызвавшая переход, и приведены действия, которые необходимо выполнить при данном переходе. Причиной перехода может быть либо запрос пользователя, поступающий в драйвер со стороны операционной системы, либо командный или информационный кадр, приходящий из канала; указанные на графе действия выполняются драйвером ASTRA.

Как видно из графа, различаются 4 состояния логики протокола. В состоянии "Стоп" протокол оказывается сразу же после загрузки в оперативную память и запуска драйвера. Он остается в этом состоянии до тех пор, пока драйвер не получит какого-либо запроса на ввод-вывод или пока из канала не придет команда START. Только после этого, пройдя два промежуточных состояния "Ожидание сброса" и "Ожидание подтверждения", протокол переходит в состояние "Работа", в котором и производится обмен данными.

Чтобы не загромождать рисунок, детальная логика этой стадии протокола в графе состояний не показана. При разъединении связи протокол возвращается в состояние "Стоп". Мы не будем описывать логику протокола подробно, поскольку она отражена в графе состояний.

7. ЗАКЛЮЧЕНИЕ

драйвер ASTRA, выполненный в операционной системе IAS, написан на языке макроассемблера MACRO-11^{8/}. В течение длительного времени /с конца 1981 г./ он успешно работает как составная часть программного обеспечения сети SONET /хотя, конечно, его можно использовать и самостоятельно для решения любых проблем, требующих организации надежного информационного канала/. В основном он работает на дуплексных линиях связи, но был опробован и при работе через полудуплексный модем на связи по коммутируемым телефонным линиям.

ЛИТЕРАТУРА

1. Reference Model of Open Systems Architecture. ISO/TC97/SC16, No.227, Aug.1979.
2. Александрова И.В. и др. ОИЯИ, 10-82-407, Дубна, 1982.
3. Вагов В.А. и др. ОИЯИ, 10-82-351, Дубна, 1982.
4. Digital Data Communications Message Protocol. Specification. Version 4.0. Digital Equipment Corporation, Maynard, Massachusetts, 1978.
5. IAS Executive Facilities Reference Manual. Digital Equipment Corporation, Maynard, Massachusetts, 1978.

6. IAS Guide to Writing a Decive Handler Task. Digital Equipment Corporation, Maynard, Massachusetts, 1978.
7. Сипсер Р. Архитектура связи в распределенных системах. "Мир", М., 1981.
8. IAS/RSX-11 MACRO-11 Reference Manual. Digital Equipment Corporation, Maynard, Massachusetts, 1976.
9. Лайх Х. ОИЯИ, P11-80-384, Дубна, 1980.
10. DL-11 Asynchronous Line Interface Manual. Digital Equipment Corporation, Maynard, Massachusetts, 1973.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
D11-80-13	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
D4-80-271	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
D4-80-385	Труды Международной школы по структуре ядра. Алушта, 1980.	5 р. 00 к.
D2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	2 р. 50 к.
D10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.
D1,2-81-728	Труды VI Международного семинара по проблемам физики высоких энергий. Дубна, 1981.	3 р. 60 к.
D17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	5 р. 40 к.
D1,2-82-27	Труды Международного симпозиума по поляризаационным явлениям в физике высоких энергий. Дубна, 1981.	5 р. 20 к.
P18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.
D2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	1 р. 75 к.
D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D2,4-83-179	Труды XV Международной школы молодых ученых по физике высских энергий. Дубна, 1982.	4 р. 80 к.
	Труды УШ Всесоюзного совещания по ускорителям заряженных частиц. Протвино, 1982 /2 тома/	11 р. 40 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.

Рукопись поступила в издательский отдел
9 декабря 1983 года

Заказы на упомянутые книги могут быть направлены по адресу:
101000 Москва, Главпочтамт, п/я 79
Издательский отдел Объединенного института ядерных исследований

**ТЕМАТИЧЕСКИЕ КАТЕГОРИИ ПУБЛИКАЦИЙ
ОБЪЕДИНЕННОГО ИНСТИТУТА ЯДЕРНЫХ
ИССЛЕДОВАНИЙ**

Индекс	Тематика
1.	Экспериментальная физика высоких энергий
2.	Теоретическая физика высоких энергий
3.	Экспериментальная нейтронная физика
4.	Теоретическая физика низких энергий
5.	Математика
6.	Ядерная спектроскопия и радиохимия
7.	Физика тяжелых ионов
8.	Криогеника
9.	Ускорители
10.	Автоматизация обработки экспериментальных данных
11.	Вычислительная математика и техника
12.	Химия
13.	Техника физического эксперимента
14.	Исследования твердых тел и жидкостей ядерными методами
15.	Экспериментальная физика ядерных реакций при низких энергиях
16.	Дозиметрия и физика защиты
17.	Теория конденсированного состояния
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники
19.	Биофизика

Алфименков А.В., Вебер Р., Северьянов В.М. P10-83-827
 Драйвер в операционной системе IAS,
 реализующий протокол информационного канала ASTRA

Описывается программный драйвер, реализующий в рамках операционной системы IAS протокол информационного канала ASTRA, который был разработан как составная часть иерархии протоколов локальной сети SONET. Протокол является байториентированным, он соответствует второму уровню семиуровневой модели архитектуры открытых систем. Драйвер реализует этот протокол для последовательных асинхронных интерфейсов типа DL-11.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Alfimenkov A.V., Weber R., Severianov V.M. P10-83-827
 Driver in IAS Operating System
 Realizing the ASTRA Information Channel Protocol

A driver in operating system IAS, realizing the information channel protocol ASTRA, is described. The protocol has been designed as a part of the protocol hierarchy of the SONET local area network. This protocol is byte-oriented one. It corresponds to the second level of the Open Systems Interconnection Model. The driver realizes the protocol for sequential asynchronous interfaces like DL-11.

The investigation has been performed at the Laboratory of Neutron Physics, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1983

Перевод О.С.Виноградовой