



(19) RU (11) 2 155 372 (13) C2  
(51) МПК<sup>7</sup> G 06 F 9/445, H 04 Q 7/34

РОССИЙСКОЕ АГЕНТСТВО  
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: 97114887/09, 30.01.1996  
(24) Дата начала действия патента: 30.01.1996  
(30) Приоритет: 30.01.1995 US 380,794  
(46) Дата публикации: 27.08.2000  
(56) Ссылки: WO 9320524 A1, 14.10.1993. SU 1347080 A1, 23.10.1987. WO 9102313 A1, 21.02.1991. US 4730187 A, 08.03.1988. US 5008814 A, 16.04.1991. SU 1562912 A1, 07.05.1990.  
(85) Дата перевода заявки РСТ на национальную фазу: 01.09.1997  
(86) Заявка РСТ:  
SE 96/00104 (30.01.1996)  
(87) Публикация РСТ:  
WO 96/24231 (08.08.1996)  
(98) Адрес для переписки:  
129010, Москва, ул. Большая Спасская 25,  
стр.3, ООО "Городисский и Партнеры",  
Емельянову Е.И.

(71) Заявитель:  
ТЕЛЕФОНАКТИЕБОЛАГЕТ ЛМ ЭРИКСОН  
(пабл.) (SE)  
(72) Изобретатель: Матс Хокан ДАЛИН (SE),  
Матс Эрланд ЭРИКСОН (SE), Леннарт Нильс  
Адольф ЛЕФГРЕН (SE)  
(73) Патентообладатель:  
ТЕЛЕФОНАКТИЕБОЛАГЕТ ЛМ ЭРИКСОН  
(пабл.) (SE)

(54) СПОСОБ ГИБКОЙ ЗАГРУЗКИ ПРОГРАММНЫХ СРЕДСТВ И УСТРОЙСТВО ДЛЯ ОСУЩЕСТВЛЕНИЯ СПОСОБА

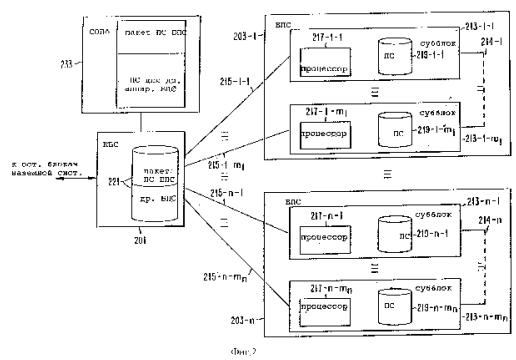
(57)  
Изобретение относится к системе и способу для загрузки файлов программных средств в системе связи. Техническим результатом является уменьшение объема кодовых данных программ базовых приемопередающих станций. В сетевой системе, содержащей первый узел, связанный со вторым узлом, способ передачи одного или более желательных файлов программных средств во второй узел включает вызов в первом узле меню идентификаторов файлов программных средств, в котором каждый идентификатор файла программных средств идентифицирует соответствующий файл из множества файлов. Идентификаторы файлов программных средств затем передаются ко второму узлу, где каждый из них анализируется и для каждого формируется отклик. Отклик, который указывает, является ли соответствующий файл из множества файлов желательным или

нежелательным, передается назад к первому узлу. Первый узел анализирует отклик и вызывает желательные файлы программных средств. Вызванные программные средства затем передаются из первого узла во второй узел. В другом варианте изобретения индикация "нежелательно" может быть расширена и включает индикацию того, что предоставляемый файл "уже загружен" или "запрещен для загрузки". Этапы представления меню, приема отклика и передачи желательных файлов повторяются в каждой из двух фаз, так что на первой фазе второй узел может запрашивать загрузку более сложной программы самозагрузки и запуска, затем в течение второй фазы загрузки более сложная программа загрузки, запускаемая во втором узле, обеспечивает принятие решения о том, какие файлы необходимо передать из первого узла. 4 с. и 12 з.п.ф-лы, 9 ил.

R  
U  
2  
1  
5  
5  
3  
7  
2  
  
C  
2

C  
2  
?  
1  
5  
5  
3  
7  
2

R U ? 1 5 5 3 7 2 C 2



R U 2 1 5 5 3 7 2 C 2



(19) RU (11) 2 155 372 (13) C2  
(51) Int. Cl. 7 G 06 F 9/445, H 04 Q 7/34

RUSSIAN AGENCY  
FOR PATENTS AND TRADEMARKS

## (12) ABSTRACT OF INVENTION

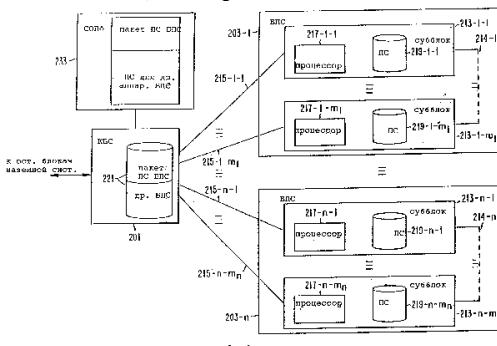
- (21), (22) Application: 97114887/09, 30.01.1996  
(24) Effective date for property rights: 30.01.1996  
(30) Priority: 30.01.1995 US 380,794  
(46) Date of publication: 27.08.2000  
(85) Commencement of national phase: 01.09.1997  
(86) PCT application:  
SE 96/00104 (30.01.1996)  
(87) PCT publication:  
WO 96/24231 (08.08.1996)  
(98) Mail address:  
129010, Moskva, ul. Bol'shaja Spasskaja 25,  
str.3, OOO "Gorodisskij i Partnerny",  
Emel'janovu E.I.

- (71) Applicant:  
TELEFONAKTIEBOLAGET LM EHRIKSSON  
(pabl.) (SE)  
(72) Inventor: Mats Khokan DALIN (SE),  
Mats Ehrland EHRIKSSON (SE), Lennart Nil's  
Adolf LEFGREN (SE)  
(73) Proprietor:  
TELEFONAKTIEBOLAGET LM EHRIKSSON  
(pabl.) (SE)

## (54) METHOD FOR FLEXIBLE LOADING OF SOFTWARE APPLICATIONS AND DEVICE WHICH IMPLEMENTS SAID METHOD

(57) Abstract:  
FIELD: computer, communication equipment.  
SUBSTANCE: method to be used in network system, which has first unit, which is connected to second unit, involves call of software applications files identifiers in first unit, so that each identifier of file of software applications identifies respective file in a set of files. Identifiers of software applications files are then transmitted to second unit, which analyzes each file and produces response, which informs whether respective file from set of files is acceptable or unacceptable and sends the response back to first unit. First unit analyzes response and calls acceptable files of software applications. Called software applications are then transmitted from first unit to second unit. Another claim of invention discloses method which involves extended unacceptability indication in order to inform that transmitted file is already loaded or its loading is forbidden. Steps for menu presentation, response reception and

transmission of acceptable files are repeated in each of the two phases, so that at first phase second unit may request loading of more complex self-loading and starting application, while during second phase more complex loading application, which is started in second unit provides possibility to make decision which files should be transmitted from first unit.  
EFFECT: decreased amount of code data for applications of base transmitting-receiving stations. 16 cl, 13 dwg



R  
U  
2  
1  
5  
5  
3  
7  
2  
C  
2

C  
2  
?  
1  
5  
5  
3  
7  
2

RU ? 1 5 5 3 7 2 C 2

Настоящее изобретение относится к системе и способу для загрузки файлов программных средств из одного узла в системе связи в один или более других узлов и, более конкретно, к системе и способу загрузки файлов программных средств от контроллера базовой станции в базовые приемопередающие станции с системе сотовой связи.

Принципы работы мобильных систем связи хорошо известны. По существу система подразделяется на наземную подсистему и подсистему радиосвязи. Мобильные абоненты осуществляют связь с любой базовой приемопередающей станцией (БПС) из некоторого числа БПС посредством интерфейса радиосвязи. БПС территориально распределены так, что в процессе перемещений между зонами обслуживания (определенном понятием "роуминг") мобильных абонентов соответствующий мобильный абонент будет находиться в зоне обслуживания по меньшей мере одной БПС системы. Если в процессе осуществления вызова мобильный абонент будет выходить из зоны относящейся к нему БПС и входить в зону соседней БПС, то система сотовой телефонной связи будет перераспределять мобильного абонента и выделять для его обслуживания соседнюю БПС. Данная процедура определяется как "передача связи". Управление процедурой передачи связи, а также другими процедурами, связанными с инициированием и завершением вызова, обрабатываются различными компонентами наземной подсистемы. Наземная подсистема также несет ответственность за маршрутизацию вызовов между мобильными абонентами и за передачу вызовов в коммутируемую телефонную сеть общего пользования.

Множество функций наземной подсистемы выполняется не одним компонентом аппаратных средств, а напротив, распределяется среди ряда ее компонентов, которые связаны между собой коммуникационной сетью. К таким компонентам относятся вышеупомянутые БПС, которые обеспечивают канал радиосвязи между наземной подсистемой и мобильными абонентами.

Наземная подсистема также может включать в себя контроллер базовых станций (КБС), предназначенный для управления операциями высокого уровня сотни и более БПС. Коммутация, передача вызовов и другие функции выполняются центром коммутации мобильных абонентов (ЦКМА), который связан с КБС в системе.

Как показано на фиг. 1(а), сеть в типовом случае состоит из БПС 103-1, ..., 103- $n$  (далее будут обозначаться ссылочной позицией 103). БПС 103 не обязательно должны быть идентичными одна другой; в типовом случае они будут образованы аппаратурой разных поколений, разных типов, модификаций, что является результатом периодической модернизации и/или наращивания системы с введением в нее новых усовершенствованных блоков аппаратуры. Нарашивание аппаратных средств может быть связано с добавлением полностью новых БПС 103 или новых их частей, например приемопередатчиков. Модернизация может быть связана с заменой

отдельных частей, например, приемопередатчиков или блоков синхронизации.

Как показано на фиг. 1(а), БПС 103 в типовом случае состоит из различных функциональных блоков 113-1-1, ..., 113-1- $m_1$ , ..., 113- $n$ -1, ..., 113- $n$ - $m_1$  (далее будут обозначаться ссылочной позицией 113), например, таких как приемопередатчики (ПРД) и устройство управления группой ПРД для выполнения функций, общих для БПС 103 в целом, таких как так называемые "центральные функции базовой станции" (ЦФБС). Обычно каждый из блоков 113 управляет своим собственным компьютером 117, имеющим память. Блоки 113 на самом высоком уровне, как упомянуто выше, имеют соответствующий канал сигнализации 115-1-1, ..., 115-1- $m_1$ , ..., 115- $n$ -1, ..., 115- $n$ - $m_1$  (далее обозначаются ссылочной позицией 115), с помощью которых они непосредственно управляются от КБС 101. Канал сигнализации 115 может быть образован на основе выделенного временного слота (интервала) с импульсно-кодовой модуляцией (ИКМ). Как вариант, канал сигнализации 115 может быть реализован как канал передачи данных уровня 2 взаимных соединений открытых систем (стандарт OSI), использующий временной слот с ИКМ (для использования в качестве физического уровня). (SOI представляет собой стандарт Международной организации по стандартизации (ISO)). Канал передачи данных уровня 2 стандарта OSI может мультиплексироваться или иным образом уплотняться с другими каналами передачи данных уровня 2 стандарта OSI для других блоков в одном и том же временном слоте с ИКМ. Такой канал или специализированный временной слот с ИКМ рассматривается как прямой канал сигнализации 115 между КБС 101 и блоком 113. Помимо использования канала сигнализации 115 для управления блоками 113, он также используется для передачи сигналов управления между КБС 101 и мобильными станциями (не показаны), которые связаны с БПС 103 по радиоканалам.

В системе, как описано выше, БПС 103 и составляющие их блоки 113 в типовом случае управляются с использованием программных средств (ПС) 119, которые загружаются от КБС 101. Загрузка может выполняться, например, после запуска системы. Другие события также могут потребовать загрузки ПС. Например, новые БПС 103, а также новые части в составе БПС 103 могут потребовать новых прикладных ПС 119, которые отличаются от тех, которые ранее использовались в БПС 103. Эти новые ПС 119 должны загружаться в новые или модифицированные БПС 103 от КБС 101. Кроме того, даже если никакие из аппаратных средств (АС) системы не изменились, однако вся сеть или части сети периодически обновляются за счет загрузки новых (обновленных) прикладных ПС 119. Во всех таких ситуациях, требующих загрузки ПС (например, при запуске, при замене АС или при обновлении ПС), различные версии ПС 119 должны загружаться в различные БПС 103.

Модернизация системы в типовом случае связана с заменой или модифицированием

R U ? 1 5 5 3 7 2 C 2

некоторых из блоков 113 при сохранении остальных неизменными. В результате ПС 119, которые должны загружаться в одни БПС 103, могут различаться от ПС 119, связанных с другими БПС 103.

В обычных системах прикладные ПС для такой структуры, как БПС 103, представляют собой пакет программ 121, состоящий из нескольких файлов (далее называемых субфайлами), по одному для каждого блока. Загрузка БПС может выполняться по каналам сигнализации 115. В некоторых вариантах реализации все каналы используются для передачи возможно одних и тех же ПС по всем каналам сигнализации 115. Как вариант, к блокам 113 могут передаваться различающиеся ПС.

В других сетевых конфигурациях, например, как показано на фиг.1(б), используется только один канал сигнализации 115 для каждой БПС 103. Данный канал сигнализации может представлять собой канал, завершающийся в блоке 113-х-1 ( $1 \leq x \leq n$ ), который выполняет вышеупомянутую функцию ЦФБС. Последующая пересылка ПС 119 к соответствующим остальным блокам 113-х-у ( $u > 1$ ) в БПС 103-х выполняется затем с помощью внутренней коммуникационной сети 123-х. Такая коммуникационная сеть 123-х может иметь различные ограничения, так что вопрос о том, каким образом распределять субфайлы ПС от КБС 101, зависит от конкретного выполнения системы. Оптимальный выбор может зависеть от типа и варианта БПС 103 и может изменяться по мере введения новых модернизированных версий ПС.

Эти известные способы загрузки программных средств имеют ряд недостатков. Рассмотрим, например, способ, иллюстрируемый на фиг.1(б), в котором КБС 101 передает пакет ПС 121 на БПС 103 посредством одного канала сигнализации 115. КБС 101 должен следить за тем, какие БПС должны загружаться какими пакетами ПС 121. Для осуществления обновления должна иметься одна новая версия пакета ПС 121 для каждого варианта БПС 103, даже если только один субблок 113 претерпел изменения. Кроме того, весь пакет ПС 121 в соответствующем варианте полностью пересыпается по всем возможным каналам сигнализации 115. Поскольку имеется множество субблоков 113 и прикладных задач, которые могут потребовать обновленных ПС более или менее независимо, количество версий пакетов может увеличиваться весьма значительно. Таким образом, данный известный способ приводит к большим непроизводительным затратам на пересылку и, следовательно, к увеличению времени загрузки. Данный способ также приводит к непроизводительным затратам объема памяти и к увеличению объема ручной обработки носителей с записью файлов.

В способе, показанном на фиг. 1 (а), каждый из субблоков 113 имеет специализированный канал сигнализации 115 для приема соответствующих субфайлов, причем КБС 101 имеет еще более трудоемкую задачу отслеживания того, какой из пакетов 121 должен передаваться к какому из субблоков 113. Каждый субблок 113 в сети

обычно имеет свой уровень версии, зарегистрированный в КБС 101, который, с учетом этого варианта, должен обеспечивать соответствующую модернизацию.

Зависимости между версиями для различных субблоков 113 также должны отслеживаться либо вручную, либо системой обработки версий в составе КБС 101, или в особой системе обеспечения процедур функционирования (СОПФ), что также известно из предшествующего уровня техники. С учетом такого большого объема информации такого рода следует, что ручной ввод данных, который по необходимости требуется в некоторые моменты процедуры обработки, создает источник потенциальных ошибок.

#### Сущность изобретения

В соответствии с вышеизложенным задачей изобретения является создание устройства и способа для распределения программных средств (ПС) от контроллера базовых станций (КБС) к одной или более базовых приемопередающих станций (БПС), которые позволяют уменьшить объем кодовых данных программ БПС, который должен храниться в КБС.

Задачей изобретения является также создание способа загрузки ПС, который позволяет уменьшить объем кодовых данных программ БПС, которые должны пересыпаться на БПС, путем пересылки на БПС только тех их частей, которые строго необходимы. Это включает ситуацию (но не ограничивается этой ситуацией), когда часть ПС уже хранится в памяти БПС.

Также задачей изобретения является создание способа загрузки ПС, который обладает общностью и гибкостью в условиях непрерывного обновления ПС, направляемого в совокупности БПС различных вариантов и/или типов, при наличии или без наличия значительной унифицированности потребностей в отдельных частях этих ПС, причем на обновления могут влиять от небольшого объема ПС до всех ПС полностью, причем обновления должны быть адаптированными к типу БПС и/или их варианту. Кроме того, задачей изобретения является создание такого способа для условий использования, когда типы и варианты, а также версии файлов ПС размножаются.

Еще одной задачей изобретения является создание способа загрузки ПС, который исключает риск ошибок оператора в представлении информации, которая контролирует то, какие ПС должны поступать на какие БПС и на какие субблоки БПС, а также управляет выбором каналов сигнализации для использования при такой загрузке. Кроме того, задачей изобретения является исключение риска в такой системе, связанного с загрузкой несовместимых субфайлов ПС.

В соответствии с одним из аспектов изобретения вышеуказанные результаты достигаются в сетевой системе, содержащей первый узел, связанный с вторым узлом, которые могут представлять собой, например, одну БПС, связанную с другой БПС в сотовой системе связи. Способ и устройство, соответствующие изобретению, обеспечивают передачу одного или более желательных файлов ПС к второму узлу

R U ? 1 5 5 3 7 2 C 2

следующим образом. В первом узле вызывается меню, которое содержит перечень идентификаторов файлов программных средств, причем каждый идентификатор файла программных средств идентифицирует один из множества файлов программных средств. Затем перечень идентификаторов файлов программных средств пересыпается с первого узла на второй узел. Во втором узле каждый из идентификаторов файлов программных средств анализируется. Формируется отклик, который содержит для каждого из идентификаторов файлов программных средств индикацию того, является ли соответствующий файл из множества файлов желательным или нежелательным. В соответствии с другим аспектом изобретения индикация того, является ли файл программных средств нежелательным, может включать альтернативные индикации того, что данный файл программных средств уже загружен или что он запрещен к загрузке. В случае, когда первый узел принимает индикацию того, что конкретный файл программных средств уже загружен, может осуществляться дальнейшая обработка для определения того, следует ли загружать этот файл программных средств во второй узел, несмотря на индикацию о том, что данные программные средства уже загружены (т.е. для определения того, следует ли принудительно загружать данный файл программных средств во второй узел).

Отклик затем пересыпается с второго узла в первый узел, где он анализируется. Первый узел затем вызывает те из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные. В соответствии с одним из аспектов изобретения, если первый узел представляет собой БПС в сотовой сети связи, то желательные файлы программных средств вызываются из памяти, входящей в состав контроллера базовых станций (КБС). В другом варианте КБС запрашивает желательные файлы программных средств из системы обеспечения процедур функционирования (СОПФ), где осуществляется их локальное хранение. После вызова этих файлов из локальной памяти СОПФ затем пересыпает желательные файлы программных средств в КБС.

После получения желательных файлов программных средств первый узел пересыпает их второму узлу.

В другом аспекте изобретения этап передачи перечня идентификаторов файлов программных средств второму узлу включает передачу первым узлом идентификаторов файлов программных средств второму узлу по одному в каждый данный момент времени и ожидание отклика на каждый запрос, прежде чем послать следующий идентификатор файла программных средств.

Еще в одном из аспектов изобретения этапы предоставления меню (т.е. идентификаторов файлов программных средств) второму узлу, приема отклика в первом узле и передачи желательных файлов от первого узла второму узлу повторяются в каждой из двух фаз, так что в первой фазе второй узел может запросить, чтобы более

сложная программа самозагрузки была загружена и запущена. Затем во второй фазе загрузки более сложная программа самозагрузки, запущенная во втором узле, принимает решение о том, какие последующие файлы принимать от первого узла.

Краткое описание чертежей

Задачи и преимущества изобретения поясняются в последующем подробном описании, иллюстрируемом чертежами, на которых показано следующее:

Фиг. 1(а) и 1(б) - блок-схемы компонентов сотовых систем связи, имеющих отношение к загрузке программных средств в соответствии со способом, известным из предшествующего уровня техники.

Фиг. 2 - блок-схема компонентов сотовой системы связи, включая пакеты программных средств, соответствующая настоящему изобретению.

Фиг. 3 - блок-схема высокого уровня, иллюстрирующая последовательность операций загрузки, осуществляемых в контроллере базовой станции в соответствии с настоящим изобретением.

Фиг. 4(а) и 4(б) - более детальная блок-схема для первой фазы операций загрузки, которые выполняются в контроллере базовой станции в соответствии с настоящим изобретением.

Фиг. 5 (а) и (б) - более детальная блок-схема второй фазы операций загрузки, которые выполняются в контроллере базовой станции в соответствии с настоящим изобретением.

Фиг. 6 - блок-схема последовательности операций загрузки, которые выполняются в базовой приемопередающей станции субблоком в состоянии самозагрузки в соответствии с настоящим изобретением.

Фиг. 7 - блок-схема последовательности операций загрузки, которые выполняются в базовой приемопередающей станции субблоком в состоянии запуска в соответствии с изобретением.

Фиг. 8 - диаграмма состояний перехода для состояний "самозагрузки" и "запуска" для субблока базовой приемопередающей станции в соответствии с настоящим изобретением.

Фиг. 9(а) и (б) - компоненты пакета программных средств, предназначенных для использования в соответствии с изобретением.

Описание предпочтительных вариантов осуществления изобретения

Различные признаки изобретения будут описаны ниже со ссылками на чертежи, на которых одинаковые элементы обозначены одинаковыми ссылочными позициями. Хотя изобретение описано в применении к сотовым системам связи, однако оно не ограничивается таким использованием. Напротив, приведенный пример осуществления показан только для иллюстрации.

На фиг. 2 иллюстрируются те компоненты сотовой системы связи, которые имеют отношение к настоящему изобретению. Как показано на чертеже, КБС 201 соединен с некоторым числом п БПС 203-1..., 203-п (далее обозначены ссылочной позицией 203). Каждая из БПС 203 включает в себя ряд субблоков: в БПС 203-1 это субблоки 213-1-1,

R U ? 1 5 5 3 7 2 C 2

R U

213-1-2,..., 213-1- $m_1$ ; в БПС 203-п это субблоки 213-п-1, 213-п-2,..., 213-п- $m_n$ . В общем случае количество субблоков 213-х в конкретной БПС 203-х не зависит от количества субблоков в любой другой БПС и поэтому выражено значением  $m_x$ . Субблоки будут обозначаться ссылочной позицией 213, но это не означает, что каждый из этих субблоков выполняет одни и те же функции. Каждый из субблоков 213 включает процессор 217-х-у ( $1 \leq x \leq n; 1 \leq u \leq m_n$ ) и память для хранения субпрограммы 219-х-у. Как упоминалось ранее, субблоки 213-х в одной и той же БПС 203-х имеют, в общем случае, полностью различные функции и конструкцию и поэтому определяются как относящиеся к различным классам. Программный код, загружаемый в любой конкретный субблок 213 должен соответствовать классу субблока. Например, субблок 213-1-1 может относиться к блоку ЦФБС, а субблоки 213-1-2,..., 213-1- $m_1$  могут относиться к классу приемопередатчиков. Субблоки 213 одного и того же класса в различных БПС 203 не обязательно идентичны по своей конструкции, так что загружаемые ПС 219 должны также соответствовать типу и варианту субблока. Могут существовать различные версии ПС для идентичных конструкций субблоков, что обеспечивает субблоки с различными функциональными возможностями.

В типовом случае более высокий уровень версии ПС обеспечивает более высокие функциональные возможности. Уровень версии ПС в одном субблоке 213 может быть или не быть совместимым с версиями ПС в других субблоках в той же самой БПС 203. Это применимо как для ПС одного и того же класса, так и к ПС разных классов. Кроме того, могут иметься субверсии одной версии, обеспечивающей эквивалентные возможности для класса, но с различными свойствами, так что выбор субверсии предоставляется оператору КБС 201 или СОПФ 233 или программе оптимизации СОПФ. Субблоки 213 БПС 203 могут взаимно соединяться посредством шины или системы шин 214, так что они могут обмениваться данными друг с другом более или менее ограниченным способом. В некоторых БПС 203 данная шина 214 может использоваться в целях пересылки загружаемых ПС от одного субблока к другому. Однако эти функциональные возможности могут отсутствовать в других БПС 203.

В соответствии с одним из аспектов изобретения каждый из субблоков 213 работает в одном из по меньшей мере двух состояний: состояние самозагрузки 801 и состояние запуска 803. На фиг. 8 представлена диаграмма перехода состояний, которая показывает, каким образом различные операции, описанные ниже, влияют на состояние субблока 213.

В рассматриваемом варианте осуществления КБС 201 непосредственно соединен с каждым из субблоков 213-1-1,..., 213-1- $m_1$ ,..., 213-n-1,..., 213-n- $m_n$  посредством соответствующих каналов сигнализации 215-1-1,..., 215-1- $m_1$ ,..., 215-n-1,..., 215-n- $m_n$ . Канал сигнализации 215 может представлять собой соединение Уровня 1 (т.е физическое соединение), например временной слот с ИКМ с использованием или без

использования дополнительного мультиплексирования. Как вариант, канал сигнализации 215 может представлять собой канал передачи данных Уровня 2 стандарта OSI на физическом соединении. Как хорошо известно из предшествующего уровня техники, различные каналы передачи данных Уровня 2 могут выполняться на одном и том же физическом соединении. Конкретное выполнение канала сигнализации 215 не существенно для настоящего изобретения и поэтому далее не будет рассматриваться более подробно и на фиг.2 не показано. Конкретный выбор протокола для использования при реализации Уровня 2 и канала связи также не существенны для изобретения, которое применимо к любому протоколу или конфигурации, которые обеспечивают логически независимые каналы сигнализации между КБС 201 и каждым субблоком 213 каждой БПС 203. Примерами хорошо известных уровней каналов передачи данных могут служить LAPD в системе ETS 300 125 (Цифровая сеть с комплексными услугами - ISDN); Применение Рекомендаций МККТТ Q.920/I.440 и Q.921/I.441 к Спецификациям Уровней каналов передачи данных интерфейса "пользователь-сеть"; LAPB в протоколе X.25 МККТТ или Уровень 2, представленный в Системе сигнализации N7. Логически, путем комбинирования селекции (коммутации) Уровня 1 и адресации Уровня 2 в каналах сигнализации 215 все субблоки 213 соединяются непосредственно и одинаково с КБС 201 в конфигурации типа звезды. Однако настоящее изобретение не ограничено данной конфигурацией, а включает в себя все возможные схемы, посредством которых КБС 201 осуществляет точное управление тем из субблоков 213, с которым он осуществляет связь (и, разумеется, той БПС 203, в которой находится данный субблок 213).

В соответствии с настоящим изобретением БПС 203 должна быть заранее обеспечена информацией, идентифицирующей, какой пакет ПС 221 должен быть загружен в каждую из БПС 203. Кроме того, либо КБС 201, либо СОПФ 233 должны быть заранее обеспечены и хранить пакеты ПС 221 для БПС 203, предположительно по одному пакету ПС 221 для каждого полностью отличающегося поколения БПС 203 или для каждого типа БПС 203 соответствующих изготовителей.

Пакет ПС 221, который используется в рассматриваемом варианте осуществления, показан детально на фиг.9(а) и 9(б). Как можно видеть из фиг.9(а), пакет ПС 221 состоит из ряда субфайлов, включая файл заголовка 911, один или более других субфайлов 921-1,..., 921-k, которые представляют собой субпрограммы, которые должны загружаться в один или более субблоков 213. Каждый из субфайлов 921-1,..., 921-k идентифицирован номером пересмотра файла ("FR"). Например, как показано на фиг.9(а), первый субфайл содержит субпрограмму, идентифицированную как FR<sub>1</sub>, последний субфайл идентифицирован как FR<sub>k</sub>.

Подобно большинству программных средств, субфайлы БПС регулярно обновляются, иногда для того, чтобы учсть изменения в аппаратных средствах, иногда для введения усовершенствований,

R U ? 1 5 5 3 7 2 C 2

связанных исключительно с программными средствами. Такие изменения, которые приводят к новой версии ПС, могут означать изменения в одном или нескольких субфайлах, которые должны работать совместно. Также изменения в числе субфайлов, которые компонуются вместе, могут представлять собой уровень новой версии. Для того чтобы отслеживать эти различные версии, номер пересмотра файла также включает в себя номер версии. В одном из возможных применений такого номера он обновляется не только в измененных субфайлах, но и в неизмененных субфайлах, которые компонуются вместе с измененными субфайлами. Это обеспечивает то, что все субфайлы, которые соответствуют конкретному типу и варианту БПС, будут иметь один и тот же номер версии.

Кроме того, в соответствии с настоящим изобретением файл заголовка 911 содержит меню 930, которое содержит перечень всех идентификаторов пересмотра файлов 931-1, ..., 931-к в соответствии с номерами пересмотра файлов для субфайлов 921-1, ..., 921-к, содержащихся в пакетах ПС 221 для БПС (см. фиг. 9(b)). Каждый из идентификаторов пересмотра файлов 931-1, ..., 931-к всеобщим и уникальным образом идентифицирует субфайл 921, так что соответствующий файл может быть локализован и считан с использованием соответствующей системы хранения файлов в КБС 201 или в СОПФ 233.

В соответствии с возможным вариантом осуществления изобретения загрузка осуществляется следующим образом, иллюстрируемым на фиг. 3-7. В возможных других вариантах части этапов или все этапы полностью, описываемые здесь как выполняемые в КБС 201, могут также выполняться в СОПФ 233, о чём будет производиться сигнализация в БПС 203 через КБС 201.

В соответствии с фиг. 3 загрузка инициируется (этап 301) одним из описанных ниже возможных событий:

Выдача команды от оператора КБС, требующей, чтобы операция загрузки была нацелена на одну БПС 203-х;

Выдача команды от оператора КБС, требующей, чтобы операция загрузки была нацелена на выбранную группу БПС 203;

Выдача команды от оператора КБС, требующей, чтобы операция загрузки была нацелена на все БПС 203, соединенные с КБС 201;

Выдача команды от оператора СОПФ, требующей, чтобы операция загрузки была нацелена на одну БПС 203-х;

Выдача команды от оператора СОПФ, требующей, чтобы операция загрузки была нацелена на выбранную группу БПС 203;

Выдача команды от оператора СОПФ, требующей, чтобы операция загрузки была нацелена на все БПС 203, соединенные с КБС 201;

Прием КБС 201 сообщения от БПС 203, указывающего, что предшествующий отказ устранен (инициирование загрузки может быть выполнено автоматически КБС 201 после приема такого сообщения);

Обнаружение КБС 201 признаков определенных сбоев в БПС, например, отсутствие ответа на некоторые команды.

Для каждой БПС 203-х ( $1 \leq x \leq n$ ), являющейся потенциальным приемником одной или более субпрограмм 921-1, ..., 921-к, локализуется соответствующий файл заголовка 911 и считывается в компьютерную память БПС (этап 303). Затем первый субблок 213-х-i ( $i=1$ ), связанный с БПС 203, выбирается для обработки. Можно видеть, что проверка на этапе 315 ("Все субблоки обработаны?"), за которой следует выполнение этапа 317 ("выбрать следующий субблок 213-х-(i+1) для обработки") создает цикл обработки, так что этапы 307-313 выполняются для каждого из субблоков 213-х-i, ..., 213-х-m<sub>x</sub> в БПС 203-х. При выполнении этапов 307-313 для конкретного субблока 213-х-i ( $1 \leq i \leq m_x$ ) осуществляется обмен сообщениями с субблоком 213-х-i с использованием соответствующего канала сигнализации 215-х-i. Хотя в рассматриваемом примере каждый субблок 213 обрабатывается один раз последовательно (в процессе выполнения этапов 307-317), это не является обязательным требованием для изобретения. Напротив, специалистам в данной области техники должно быть ясно, что способы, описанные здесь, могут быть применены для обработки всех блоков 213 одновременно.

В рассматриваемом примере осуществления обработка, выполняемая в КБС 210 на этапах 307-313, спроектирована для осуществления двухфазной операции загрузки. На первой фазе выполнения каждому субблоку 213-х-i предоставляется возможность использовать резидентную программу самозагрузки (или, как вариант, выполняемую прикладную программу, которая эмулирует программу самозагрузки) для выбора субфайлов для загрузки и запуска. В соответствии с изобретением субфайлы, переносимые в процессе выполнения этой фазы, сами могут быть программами самозагрузки, которые могут выполнять более сложную операцию загрузки программы, чем операция, выполняемая резидентной программой самозагрузки. Поэтому за первой фазой следует вторая фаза, на которой каждый субблок 213-х-i вновь получает возможность выбрать субфайлы, подлежащие загрузке и запуску. Однако в течение второй фазы именно программа, которая была загружена и запущена в течение первой фазы, принимает решения. Разумеется, возможные варианты осуществления не ограничиваются двумя фазами, а могут быть распространены на любое необходимое число фаз, которое соответствует конкретным условиям использования.

Ниже будут описаны две фазы для рассматриваемого варианта осуществления изобретения. На этапе 307 КБС 201 вызывает процедуру "меню запроса фазы самозагрузки в файле заголовка". Данная процедура (описанная ниже более подробно со ссылками на фиг. 4(a) и (b)) проходит через файл заголовка 911 и определяет для каждого перечисленного идентификатора пересмотра файла 931-1, ..., 931-к, должен ли соответствующий субфайл 921-1, ..., 921-к передаваться к субблоку 213-х-i. Выполнение на этапе 307 также гарантирует, что выбранные субфайлы загружаются в субблок 213-х-i. В соответствии с предпочтительным

R U ? 1 5 5 3 7 2 C 2

вариантом осуществления изобретения выбор субфайла и последующая загрузка выполняются путем поочередного запроса блока 213-х-*i* относительно субфайла 921-*j* ( $1 \leq j \leq k$ ) и затем в зависимости от полученного ответа и возможной другой информации передачи субфайла 921-*j* к субблоку 213-х-*i*.

Следует отметить, что поскольку процедура "меню запроса фазы загрузки в файле заголовка" может выполняться без осуществления загрузки вообще (т.е. субблок 213-х-*i* может отказаться от всех предоставляемых файлов 921), то субблок 213-х-*i* неизбежно должен вводиться в состояние самозагрузки 801 перед выполнением этапа 307. Предпочтительно избегать введения субблока 213-х-*i* в состояние запуска и выведения его из этого состояния, поскольку переходы между состояниями могут привести к запуску других посторонних процедур, требующих времени, например, связанных с тестированием аппаратных средств. Поэтому в рассматриваемом варианте осуществления такие переходы между состояниями минимизированы. В случае, если субблок 213-х-*i* находится в состоянии запуска 803 в момент начала выполнения этапа 307, то процедура "меню запроса фазы самозагрузки в файле заголовка" будет вынуждать субблок 213-х-*i* перейти в состояние самозагрузки 801, если субфайл 921-*j* действительно выбран для загрузки.

После обработки на этапе 307 КБС 201 использует соответствующий канал сигнализации 215-х-*i* для передачи команды "установка в состояние запуска" на субблок 213-х-*i*. Это вынуждает субблок 213-х-*i* выйти из состояния самозагрузки 801, если он в нем находился, и перейти в состояние запуска 803. Любые субфайлы 921-*j*, которые были загружены в результате выполнения этапа 307, начнут выполнять. В соответствии с одним из аспектов изобретения эти загруженные субфайлы 921-*j* могут сами создавать программу самозагрузки, которая будет управлять тем, какие субфайлы 921-*j* должны загружаться в течение следующей фазы самозагрузки (описано ниже).

Затем на этапе 311 КБС 201 переходит на вторую фазу процесса загрузки путем вызова процедуры "меню запроса фазы запуска в файле заголовка" (описано более детально ниже со ссылками на фиг.5(а) и (б)). Подобно процедуре, вызванной на этапе 307, процедура "меню запроса фазы запуска в файле заголовка" проходит через файл заголовка 911 и определяет для каждого из перечисленных идентификаторов пересмотра файлов 931-1,..., 931-*k*, должен ли соответствующий субфайл 921-1, ..., 921-*k* передаваться к субблоку 213-х-*i*. Выполнение этапа 311, таким образом, гарантирует, что выбранные субфайлы 921 загружаются в субблок 213-х-*i*.

После завершения вызова этой процедуры передачи субфайлов КБС 201 передает метку "конец пакета" в субблок 213-х-*i*. Как будет пояснено ниже, прием метки "конец пакета" субблоком 213-х-*i* обеспечивает то, что субблок 213-х-*i* сначала определяет, какой из загруженных файлов должен быть запущен и затем запускает эти файлы.

Затем как поясняено выше, на этапе 315 определяется, следует ли выполнять процедуру загрузки для каждого из субблоков 213-х-*i*, ..., 213-х-*m*<sub>x</sub> в БПС 203-х. Если да, то обработка завершается в блоке 319. В противном случае на этапе 317 выбирается следующий субблок 213-х-*(i-1)* и повторяется обработка на этапах 307-313.

Таким образом, для каждого субблока 213-х-*i* КБС 201 выполняет соответствующую процедуру запроса файлов дважды: один раз на этапе 307 и затем снова на этапе 311. Различные файлы могут быть загружены в процедуру "запроса фазы запуска..." по сравнению с процедурой "запроса фазы самозагрузки...", поскольку принимающий субблок 213-х-*i* находится в различном состоянии и выполняет разные программы при каждом из этих двух вызовов. Таким образом, при выполнении этапа 307 субблок 213-х-*i* может быть в состоянии самозагрузки 801, выполняя резидентную программу самозагрузки (или в состоянии запуска 803, выполняя прикладную программу, которая эмулирует резидентную программу самозагрузки). В противоположность этому при выполнении этапа 311 субблок 213-х-*i* гарантированно находится в состоянии запуска 803 в результате передачи команды "установка запуска" в субблок 213-х-*i* (этап 309). В состоянии запуска 803 субблок 213-х-*i* будет использовать либо программу самозагрузки, которая была только что загружена, либо прикладную программу, которая была загружена ранее, в целях принятия решений о загрузке на второй фазе процесса загрузки.

На фиг. 4(а) и 4(б) более детально показаны операции процедуры "меню запроса фазы самозагрузки в файле заголовка" для КБС, вызываемой на этапе 307. Эта процедура передачи субфайлов на фазе 1 начинается при ее вызове (этап 4). На этапе 403 выполняется инициализация процедуры путем выбора идентификатора пересмотра первого файла 931-*j* (*j*=1) из меню 930. Затем на этапе 405 КБС 201 использует канал сигнализации 215-х-*i* для передачи сообщения в субблок 213-х-*i*, запрашивающего о том, является ли субпрограмма, идентифицированная идентификатором пересмотра файла 931-*j*, "желательной", "уже загруженной" или "запрещенной" для загрузки.

Субблок 213-х-*i* генерирует отклик в соответствии с методами, описанными ниже со ссылками на фиг.6 и 7, и затем возвращает отклик в КБС 201. Отклик принимается КБС 201 на этапе 407 и затем на этапе 409 выделяется "ответ состояния самозагрузки". Поскольку субблок 213-х-*i* может находиться в состоянии запуска 803, то отсутствует требование, чтобы он обеспечивал "ответ состояния самозагрузки" в своем отклике, хотя это является допустимым действием на этапе запуска 803. Таким образом, необходимо на этапе 411 определить, присутствует ли действительно в принятом отклике "ответ состояния самозагрузки".

Если в отклике отсутствует "ответ состояния самозагрузки", то необходимо принудительно перевести субблок 213-х-*i* в состояние самозагрузки 801, так чтобы запрос мог быть повторен в отношении данной конкретной субпрограммы. Соответственно на

R U 2 1 5 5 3 7 2 C 2

RU 2 1 5 3 7 2 C 2

этапе 413 канал сигнализации 215-х-и используется для передачи команды "установка в состояние самозагрузки" на субблок 213-х-и. Затем выполнение обработки переходит на этап 405 и процедура повторяется.

Если "ответ состояния самозагрузки" был успешно выделен из отклика, то выполнение обработки переходит на этап 415 для определения сущности ответа. Если субблок 213-х-и указал, что субпрограмма, идентифицированная идентификатором пересмотра программы 931-ј, "запрещена" к загрузке, то осуществляется переход на этап 427, где выполняется проверка того, все ли идентификаторы пересмотра файлов 931-ј в меню 930 были обработаны. Если да, то на этапе 431 процедура "меню запроса фазы самозагрузки в файле заголовка" возвращается к точке вызова процедуры (т.е. возобновляется выполнение обработки на этапе 309). В противном случае на этапе 429 выбирается субпрограмма, идентифицированная следующим идентификатором пересмотра файла 931-(ј+1), и выполнение обработки продолжается на этапе 405.

Если на этапе 415 определено, что субблок 213-х-и указал, что субпрограмма, идентифицированная идентификатором пересмотра файла 931-ј, является либо "желательной", либо "уже загруженной", но подлежащей загрузке в любом случае (т.е. осуществляется переход по ветви "вынужденная загрузка" с этапа 417), то осуществляется переход на этап 419. Информация от оператора КБС или от СОПФ может быть использована для определения того, следует ли осуществить принудительную загрузку субпрограммы 921-ј в субблок 213-х-и. Примером ситуации, в которой необходима принудительная загрузка, является то, когда номер субверсии для модуля ПС, который должен проверяться, не распознается как более высокий уровень версии, чем у модуля, который он должен заменить. В этом случае существующий модуль должен сформировать отклик, указывающий, что он "уже загружен". Другие ситуации также могут обусловить желательность принудительной загрузки субпрограммы 921-ј в субблок 213-х-и. Поскольку такие ситуации являются специфическими для каждой прикладной задачи, то их перечисление не входит в объем изобретения.

Если субпрограмма идентифицирована идентификатором пересмотра файла 931-ј как "уже загруженная" и не требует принудительной загрузки, то выполнение обработки просто продолжается на этапе 427, как описано выше.

При выполнении этапа 419 определяется, что должна выполняться операция загрузки. Т.к. субблок 213-х-и должен быть в соответствующем состоянии (т.е. состоянии самозагрузки 801) для приема субфайлов 921 в течение этой фазы загрузки, то КБС 201 проверяет (этап 419) состояние субблока 213-х-и (в предпочтительном варианте осуществления КБС 201 постоянно отслеживает текущее состояние каждого субблока 213), и, если субблок 213-х-и находится в состоянии запуска 803, то КБС 201 сначала использует канал сигнализации

215-х-и для передачи команды "установка самозагрузки" субблоку 213-х-и. Данный этап пропускается, если субблок 213-х-и уже находится в состоянии самозагрузки 801.

Затем на этапе 423 КБС 201 вызывает субфайл 921-ј, который соответствует идентификатору пересмотра файла 931-ј. Это может выполнятся путем считывания этих файлов из памяти в составе КБС 201. В соответствии с другим вариантом осуществления изобретения это может выполняться путем подачи запроса на СОПФ 233, которая содержит память для хранения этих файлов. В последнем случае СОПФ 233 вызывает запрошенный субфайл 921-ј из своей локальной памяти и затем пересыпает его на КБС 201.

Затем на этапе 425 КБС 201 использует канал сигнализации 215-х-и для передачи вызванного субфайла 921-ј на субблок 213-х-и. Выполнение обработки затем переходит к этапу 427 проверки для определения того, все ли позиции меню обработаны. Если да, то процедура заканчивается. В противном случае обработка повторяется.

На фиг.5(а) и 5(б) более детально показаны особенности операции, осуществляющейся БКС по процедуре "меню запроса фазы запуска в файле заголовка", которая вызывается на этапе 311. Данная процедура пересылки субфайлов в фазе 2 аналогична процедуре "меню запроса фазы самозагрузки в файле заголовка" и начинается при ее вызове (этап 501). На этапе 503 выполняется инициализация процедуры путем выбора идентификатора пересмотра первого файла 931-ј (ј=1) из меню 930. Затем на этапе 505 КБС 201 использует канал сигнализации 215-х-и для передачи сообщения в субблок 213-х-и, запрашивающего о том, является ли субпрограмма, идентифицированная идентификатором пересмотра файла 931-ј, "желательной", "уже загруженной" или "запрещенной" для загрузки.

Субблок 213-х-и генерирует отклик в соответствии с методами, описанными ниже со ссылками на фиг. 7 (в результате выполнения этапа 309 субблок 213-х-и гарантированно находится в состоянии запуска 803), и затем возвращает отклик в КБС 201. Отклик принимается КБС 201 на этапе 507 и затем на этапе 509 выделяется "ответ состояния запуска". Поскольку субблок 213-х-и находится в состоянии запуска 803, то всегда присутствует "ответ состояния запуска".

Затем выполнение обработки переходит на этап 515 для определения сущности ответа. Если субблок 213-х-и указал, что субпрограмма, идентифицированная идентификатором пересмотра программы 931-ј, "запрещена" к загрузке, то осуществляется переход на этап 519, где выполняется проверка того, все ли идентификаторы пересмотра файлов 931-ј в меню 930 были обработаны. Если да, то на этапе 523 процедура "меню запроса фазы запуска в файле заголовка" возвращается к точке вызова процедуры (т.е. возобновляется выполнение обработки на этапе 313). В противном случае на этапе 521 выбирается субпрограмма, идентифицированная следующим идентификатором пересмотра

R U ? 1 5 5 3 7 2 C 2

файла 931-(j+1), и выполнение обработки продолжается на этапе 505.

Если на этапе 511 определено, что субблок 213-х-i указал, что субпрограмма, идентифицированная идентификатором пересмотра файла 931-j, является либо "желательной", либо "уже загруженной", но подлежащей загрузке в любом случае (т.е. осуществляется переход по ветви "вынужденная загрузка" с этапа 513), то осуществляется переход на этап 515. Информация от оператора КБС или от СОПФ может быть использована для определения того, следует ли осуществить принудительную загрузку субпрограммы 921-j в субблок 213-х-i. Как упоминалось выше, примером ситуации, в которой необходима принудительная загрузка, является то, когда номер субверсии для модуля ПС, который должен проверяться, не распознается как более высокий уровень версии, чем у модуля, который он должен заменить.

Если субпрограмма идентифицирована идентификатором пересмотра файла 931-j как "уже загруженная" и не требует принудительной загрузки, то выполнение обработки просто продолжается на этапе 427, как описано выше.

При выполнении этапа 515 определяется, что осуществляется операция загрузки. Для КБС нет необходимости направлять команду изменения состояния, поскольку уже известно, что субблок 213-х-i находится в состоянии запуска 803. Поэтому в этот момент КБС 201 вызывает субфайл 921-j, который соответствует идентификатору пересмотра файла 931-j. Как и ранее, это может выполняться путем считывания этих файлов из памяти в составе КБС 201. В соответствии с другим вариантом осуществления изобретения это может выполняться путем направления запроса в СОПФ 233, которая имеет память для хранения этих файлов. В последнем случае СОПФ 233 вызывает запрошенный субфайл 921-j из своей локальной памяти и посыпает его в КБС 201.

Затем на этапе 517 КБС 201 использует канал сигнализации 215-х-i для передачи вызванного субфайла 921-j на субблок 213-х-i. Выполнение обработки затем переходит к этапу 519 проверки для определения того, все ли позиции меню обработаны. Если да, то процедура заканчивается. В противном случае выбирается следующий идентификатор пересмотра файла 931-(j+1) и цикл обработки повторяется.

Для обеспечения функционирования с использованием КБС 201, выполняющего вышеописанную процедуру, субблок 213 БПС выполняет программу, которая будет описана со ссылками на фиг.6-8. Как упоминалось ранее, субблок 213 может находиться в одном из двух состояний, имеющих отношение к процедуре загрузки: в состоянии самозагрузки 801 и в состоянии запуска 803 (фиг. 8). В состоянии самозагрузки 801 субблок 213 выполняет резидентную программу самозагрузки (т. е. запомненную в энергонезависимой памяти или выполняемую прикладную программу, которая эмулирует программу самозагрузки). В состоянии запуска 803 субблок 213 запускает прикладную программу, которая загружается в его память. Как упомянуто выше, эта

прикладная программа может быть более сложной программой самозагрузки, для определения того, какие файлы должны быть загружены на второй фазе процедуры загрузки. На фиг. 8 представлена диаграмма перехода состояний, которая иллюстрирует то, каким образом прием определенных сообщений влияет на состояние субблока 213.

На фиг. 6 показана блок-схема операций для субблока 213, который находится в состоянии самозагрузки 801. На этапе 601 субблок 213 ожидает приема сообщения от КБС 201. Когда КБС 201 начинает выполнять вышеописанную процедуру "меню запроса фазы загрузки в файле заголовка" (фиг.4(а) и 4(б)), субблок 213 получает сообщение запроса о том, идентифицирована ли субпрограмма, идентифицированная идентификатором пересмотра файлов 931-j как "желательная", "уже загруженная" или "запрещенная" для загрузки. Это приводит к выполнению этапа 611, где запрос принимается субблоком 213.

Затем на этапе 613 субблок 213 подготавливает "ответ состояния самозагрузки". Это может выполняться, например, путем использования любого хорошо известного способа (такого как вычисление контрольной суммы циклического избыточного кода (ЦИК)) для определения того, действительный ли программный файл уже загружен в память. Если да, то идентификатор пересмотра файла 931-j, который был принят в запросе, сравнивается с идентификатором загруженного файла, за исключением его части, относящейся к субверсии. Если оба они совпадают, то ответ "уже загружено" пересыпается в ответном сообщении (этап 615). Если действительный программный файл в данный момент не загружен в память субблока или если имеет место рассогласование при сравнении, то идентификатор пересмотра файла 931-j анализируется для того, чтобы проверить, является ли он файлом ПС для определенного типа БПС, варианта субблока и класса субблока, которые действительны для данного конкретного субблока 213 в состоянии самозагрузки 801. Если да, то ответ "желательно" направляется в ответном сообщении (этап 615).

Если не генерируется ни один из вышеуказанных ответов, то в ответном сообщении направляется ответ "запрещено" (этап 615). Субблок 213 затем переходит на этап 601, где он вновь ожидает прихода следующих сообщений.

Следующее сообщение может быть новым запросом, и описанная процедура обработки повторяется. Если следующее сообщение является субфайлом 921-j, оно принимается на этапе 607. Затем на этапе 609 принятый субфайл 921-j обрабатывается в соответствии с любым числом хорошо известных способов, например декомпрессии, выполнения суммирования с использованием ЦИК, дешифрования или проверки длины. Наконец, обработанный файл запоминается в памяти субблока, из которой он может вызываться для выполнения.

Рассмотрим ситуацию, когда субблок 213 ожидает сообщения от КБС 201 (этап 601). Следующее сообщение может быть командой перехода в состояние запуска 803. В этом

R U ? 1 5 5 3 7 2 C 2

случае он принимает команду (этап 605) и затем начинает выполнение программного файла, который был либо запомнен, либо находился в памяти в состоянии самозагрузки 801 (этап 605). После начала выполнения программы состояние субблока 213 изменяется на состояние запуска 803, как показано переходом в точку 1 на фиг. 7.

На фиг. 7 представлена блок-схема последовательности операций субблока 213 после перехода его в состояние запуска 803. На этапе 701 он ожидает приема сообщения от КБС 201. (Разумеется, субблок 213 также выполняет другие задачи, как предписано выполняемой программой. Программа, которая осуществляется теперь, является той, которая была загружена в состоянии самозагрузки 801 (или той, которая была определена как резидентная на первой фазе процесса загрузки). Это означает, что анализ сообщений запроса теперь осуществляется в загружаемой программе. Такой анализ может поэтому строиться по сложным процедурам

и может модифицироваться простым изменением ПС. Короче говоря, он отображает всю гибкость динамически загружаемой управляющей программы.

В других отношениях диаграмма, представленная на фиг.7, аналогична показанной на фиг.6. Первое сообщение, приходящее после команды установки в состояние запуска (этап 309), является, таким образом, запросом, принимаемым на этапе 711. На этапе 713 субблок 213 дополнительно подготавливает "ответ состояния самозагрузки" аналогично тому, как описано выше для этапа 613. Это делается таким образом, чтобы программа, выполняемая в состоянии запуска 803, имела возможность выдачи ответов для использования на первой фазе процесса загрузки.

Затем на этапе 715 субблок 213 подготавливает "ответ состояния запуска". Это действие не является факультативным. Подготовка "ответа состояния запуска" полностью зависит от программы, которая загружалась (или была определена как уже загруженная) на этапе самозагрузки. Этот этап может выполнять любую функцию для получения решения о том, следует ли принимать предлагаемый субфайл 921-*j* ("ответ состояния запуска" = "желательно"), или необходимо показать, что предлагаемый субфайл 921-*j* "уже загружен" (может быть загружен вновь) или "запрещен" к загрузке. Например, субблок 213 может сформировать "ответ состояния запуска" как функцию действительности субфайла, принимая во внимание тип БПС, вариант субблока и класс субблока. Субблок 213 может, кроме того, учитывать номер версии предлагаемого субфайла 921-*j*, в частности, равен ли он, или ниже, или выше, чем номер уже загруженной версии (если она есть). Другая возможная информация, которая может быть использована, заключается в том, является ли предоставляемая версия совместимой с версиями других субфайлов или субфайлов, расположенных в других субблоках БПС 203-*x* (внутренняя сеть 214-*x* может использоваться для определения этой информации).

После подготовки "ответа состояния

запуска" обработка переходит на этап 717, где "ответ состояния запуска" и дополнительный "ответ состояния самозагрузки" кодируются в ответном сообщении, которое пересыпается к КБС 201. В соответствии с предпочтительным вариантом осуществления изобретения ответное сообщение кодируется для формирования первого слота (интервала), выделенного для ответа для "текущего" состояния, и второго слота, выделенного для ответа для "другого" состояния. Тем самым, поскольку субблок 213 находится в состоянии запуска 803, то "ответ состояния запуска" будет кодироваться в слоте ответного сообщения, выделенном для "текущего" состояния, а любой "ответ состояния самозагрузки" будет помещаться в соответствующий слот, выделенный для "другого" состояния.

Затем субблок 213 возвращается к ожиданию сообщений (этап 701) и повторяет обработку, как описано выше для состояния самозагрузки 801 (см. фиг.6). Т. е., если приходит субфайл 921-*j*, то субблок 213 обрабатывает и запоминает его (этапы 707 и 709), как описано выше для этапов 607 и 609.

Если приходит новое сообщение запроса, то оно обрабатывается на этапах 711-717, как описано выше.

Если приходит метка "конец пакета" (присыпаемая КБС 201 на этапе 313), то она принимается, и субблок 213 использует специфическую для прикладной задачи программу для определения того, какая из программ, загружаемых в память, должна быть запущена (этапы 703-704). Состояние субблока 213 остается неизменным, и независимо от того, какая программа выполняется при завершении этапа 706, гарантируется, что этап 701 выполняется для возобновления ожидания следующего сообщения. Т.е. любая новая программа, которая принимает на себя управление от текущей выполняемой программы, должна иметь возможность выполнять этапы, показанные на фиг. 7.

Если приходит команда "установить самозагрузку", то субблок 213 принимает команду (этап 705) и затем запускает выполнение резидентной программы самозагрузки (этап 706). Как вариант субблок 213 может начать выполнение прикладной программы, которая эмулирует резидентную программу самозагрузки. В любом случае состояние субблока 213 изменяется на состояние самозагрузки 801, как показано переходом в точку 2 на фиг.6.

В другом аспекте изобретения только один из субблоков 213-*x* в БПС 203-*x* будет разрешать загружаться какому-либо из субфайлов 921-*j*. Остальные субблоки 213-*x*-*y* ( $2 \leq y \leq m_x$ ) всегда формируют ответ "запрещено" в ответном сообщении на любой запрос, независимо то того, находятся ли они в состоянии загрузки 801 или в состоянии запуска 803. Первый субблок 213-*x*-*i* в этом случае будет нести ответственность за прием всех субфайлов 921 для БПС 203-*x* и распределение их другим субблокам 213-*x*-*y* с помощью внутренней сети 214-*x*.

В соответствии с еще одним из аспектов изобретения число фаз процесса загрузки может изменяться. Т.е. оно может быть сокращено до одного процессы запроса (один

R U 2 1 5 5 3 7 2 C 2

R U ? 1 5 5 3 7 2 C 2

запрос на субфайл 921), или оно может быть увеличено до трех и более фаз. Заметим, что изобретение охватывает возможность эффективно сокращать число фаз до одной путем выполнения всех переносов файлов за одну фазу, и затем субблоки 213 будут отказываться от всех предлагаемых субфайлов 921-ј на всех других фазах загрузки.

Кроме того, число фаз может изменяться динамически, основываясь на ответной информации, принятой от субблоков 213 БПС. Оно также может базироваться на информации, которую возможно добавлять посредством идентификатора пересмотра файла 931-ј в файле заголовка.

В другом варианте осуществления изобретения субблоки 213-х-и не должны обрабатываться последовательно. В противоположность этому КБС 201 может посыпать запросы нескольким или всем субблокам 213 параллельно.

В еще одном варианте осуществления изобретения КБС не должен ожидать отклика от субблоков 213-х по отношению к первому субфайлу 921-ј перед передачей запроса по отношению к следующему субфайлу 921-(j+1). Вместо этого КБС 201 может передать весь перечень идентификаторов пересмотра файлов 931 на субблок 213-х-и сразу. Преимущество данного варианта состоит в том, что субблок 213-х-и может анализировать весь перечень перед тем, как осуществлять прием конкретных субфайлов 921. Специалисты в данной области техники могут легко внести изменения в примеры осуществления, показанные на фиг. 3-7, для получения такого альтернативного варианта. Например, можно создать программу, которая обеспечит передачу от КБС 201 каждого из идентификаторов пересмотра файлов 931 субблоку 213-х-и с последующей меткой "конец пакета", чтобы субблок 213-х-и мог знать, что он принял все идентификаторы пересмотра файлов 931. После приема метки "конец пакета" субблок 213-х-и мог бы анализировать полученный перечень, осуществлять свой выбор и соответственно кодировать выбранные файлы в ответном сообщении. КБС 201 после приема ответного сообщения мог бы вызвать и передать запрошенные субфайлы 921, как описано выше.

Как описано выше, настоящее изобретение имеет ряд преимуществ по сравнению со способами загрузки ПС, известными из предшествующего уровня техники. Например, оно учитывает тот факт, что пакет ПС может состоять из различных субфайлов, а также возможность того, что загрузка может осуществляться по различным каналам пересылки, каждый из которых завершается в различном субблоке 213-х-и БПС 203-х.

Кроме того, настоящее изобретение не требует полного взаимного соответствия между субблоками и субфайлами. Поскольку способ, соответствующий изобретению, обеспечивает возможность субблоку отказаться от предоставляемого субфайла или принять его, то субфайлы могут быть сгруппированы в пакеты, которые не зависят от требований, предъявляемых какими-либо из субблоков.

Изобретение было описано на примере

конкретного варианта осуществления. Однако специалистам в данной области техники должно быть ясно, каким образом можно реализовать данное изобретение в конкретных формах, отличающихся от предпочтительного варианта, описанного выше. Это может быть осуществлено без изменения сущности изобретения.

Предпочтительный вариант осуществления является иллюстративным и не накладывает никаких ограничений. Объем изобретения определяется пунктам формулы изобретения, а не предшествующим описанием, и все эквиваленты и изменения, которые находятся в пределах пунктов формулы изобретения, должны считаться включенными в него.

### Формула изобретения:

1. Способ передачи одного или нескольких файлов программных средств второму узлу в сетевой системе, содержащей первый узел, связанный со вторым узлом, отличающийся тем, что содержит этапы: вызова в первом узле меню, содержащего перечень идентификаторов файлов программных средств, причем каждый идентификатор файла программных средств идентифицирует соответствующий файл из множества файлов программных средств, передачи перечня идентификаторов файлов программных средств из первого узла во второй узел, анализа во втором узле каждого из идентификаторов файлов программных средств и формирования отклика, который включает в себя для каждого из идентификаторов файлов программных средств индикацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, передачи отклика от второго узла к первому узлу, анализа в первом узле отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были индицированы вторым узлом как желательные, и передачи вызванных файлов программных средств из первого узла во второй узел.

2. Способ по п.1, отличающийся тем, что индикация того, что соответствующий файл из множества файлов программных средств является нежелательным, представляет собой либо индикацию того, что соответствующий файл из множества файлов программных средств уже загружен во второй узел, либо индикацию того, что соответствующий файл из множества файлов программных средств запрещен для загрузки во второй узел.

3. Способ по п.1, отличающийся тем, что сетевая система представляет собой систему сотовой телефонной связи, первый узел представляет собой контроллер базовых станций, а второй узел представляет собой базовую приемопередающую станцию, контроллер базовой станции содержит средство для хранения множества файлов программных средств, при этом осуществляемый в первом узле этап вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные, включает этап вызова из указанного средства хранения тех файлов из

R U 2 1 5 5 3 7 2 C 2

множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные.

4. Способ по п.1, отличающийся тем, что сетевая система представляет собой систему сотовой телефонной связи, первый узел представляет собой контроллер базовых станций, а второй узел представляет собой базовую приемопередающую станцию, при этом осуществляемый в первом узле этап вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные, включает этапы: передачи из первого узла в систему обеспечения процедур функционирования запроса тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные, вызова в системе обеспечения процедур функционирования запрошенных файлов программных средств и передачи вызванных файлов программных средств из системы обеспечения процедур функционирования в первый узел.

5. Способ по п.1, отличающийся тем, что этап передачи перечня идентификаторов файлов программных средств во второй узел включает этапы: (а) выбора из перечня идентификаторов файлов программных средств первого идентификатора для использования в качестве текущего идентификатора файла программных средств, (б) передачи текущего идентификатора файла программных средств из первого узла во второй узел, (в) ожидания отклика от второго узла, (г) после приема отклика от второго узла выбора из перечня идентификаторов файлов программных средств следующего идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств и (д) повторение этапов с (б) по (г) до тех пор, пока каждый из идентификаторов файлов программных средств в перечне идентификаторов файлов программных средств не будет передан от первого узла во второй узел.

6. Способ по п. 5, отличающийся тем, что этап (г) включает операции: после приема отклика от второго узла ожидания того, пока отклик будет проанализирован в первом узле, ожидания, когда те файлы из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны во втором узле как желательные, будут вызваны в первом узле, ожидания, когда вызванные файлы программных средств будут переданы из первого узла во второй узел, и выбора из перечня идентификаторов файлов программных средств следующего идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств.

7. Способ передачи одного или нескольких желательных файлов программных средств второму узлу в сетевой системе, содержащей первый узел, связанный со вторым узлом, отличающейся тем, что содержит этапы: вызова в первом узле меню, содержащего

перечень идентификаторов файлов программных средств, причем каждый идентификатор файла программных средств идентифицирует соответствующий файл из множества файлов программных средств, передачи перечня идентификаторов файлов программных средств из первого узла во второй узел, анализа во втором узле с использованиемрезидентной программы каждого из идентификаторов файлов программных средств и формирования первого отклика, который включает в себя для каждого из идентификаторов файлов программных средств первую индикацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, передачи первого отклика от второго узла к первому узлу, анализа в первом узле первого отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были индицированы вторым узлом как желательные, и передачи вызванных файлов программных средств из первого узла во второй узел, начала выполнения во втором узле файлов программных средств, которые были приняты от первого узла, передачи второй раз перечня идентификаторов файлов программных средств из первого узла во второй узел, использования во втором узле выполняемых файлов программных средств, которые были получены от первого узла, для анализа каждого из идентификаторов файлов программных средств и формирования второго отклика, который содержит для каждого идентификатора файла программных средств вторую индикацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, передачи второго отклика от второго узла к первому узлу, анализа в первом узле второго отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом во втором отклике как желательные, и передачи из первого узла второму узлу файлов программных средств, которые были вызваны в результате анализа в первом узле второго отклика.

8. Способ по п.7, отличающийся тем, что каждая из упомянутых первой и второй индикаций, что соответствующие файлы из множества файлов программных средств являются нежелательными, представляют собой индикацию того, что соответствующие файлы из множества файлов программных средств уже загружены во второй узел, или индикацию того, что соответствующие файлы из множества файлов программных средств запрещены к загрузке во второй узел.

9. Устройство для передачи одного или нескольких желательных файлов программных средств второму узлу в сетевой системе, содержащей первый узел, связанный со вторым узлом, отличающееся тем, что содержит в первом узле средство для вызова меню, содержащего перечень идентификаторов файлов программных средств, причем каждый идентификатор файла программных средств идентифицирует соответствующий файл из множества файлов

R U ? 1 5 5 3 7 2 C 2

R U

программных средств, и средство для передачи перечня идентификаторов файлов программных средств из первого узла во второй узел, во втором узле средство для анализа каждого из идентификаторов файлов программных средств и формирования первого отклика, который включает в себя для каждого из идентификаторов файлов программных средств индикацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, и средство для передачи первого отклика от второго узла к первому узлу, при этом первый узел дополнительно содержит средство для анализа отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были индицированы вторым узлом как желательные, и средство для передачи вызванных файлов программных средств из первого узла во второй узел.

10. Устройство по п.9, отличающееся тем, что индикация того, что соответствующий один из множества файлов программных средств является нежелательным, представляет собой индикацию того, что соответствующий один из множества файлов программных средств уже загружен во второй узел, или индикацию того, что соответствующий один из множества файлов программных средств запрещен к загрузке во второй узел.

11. Устройство по п.9, отличающееся тем, что сетевая система представляет собой систему сотовой телефонной связи, первый узел представляет собой контроллер базовых станций, а второй узел представляет собой базовую приемопередающую станцию, контроллер базовой станции содержит средство для хранения множества файлов программных средств, при этом содержащееся в первом узле средство вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные, содержит средство для вызова из локального средства хранения тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные.

12. Устройство по п.9, отличающееся тем, что сетевая система представляет собой систему сотовой телефонной связи, которая включает в себя систему обеспечения процедур функционирования, связанную с первым узлом, первый узел представляет собой контроллер базовых станций, второй узел представляет собой базовую приемопередающую станцию, содержащееся в первом узле средство вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом как желательные, содержит средство для передачи из первого узла системы обеспечения процедур функционирования запроса тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были

указаны вторым узлом как желательные, а система обеспечения процедур функционирования содержит средство для вызова запрошенных файлов программных средств и средство для передачи вызванных файлов программных средств из системы обеспечения процедур функционирования к первому узлу.

13. Устройство по п.9, отличающееся тем, что средство для передачи идентификаторов файлов программных средств ко второму узлу содержит средство для выбора из перечня идентификаторов файлов программных средств первого идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств, средство для передачи текущего идентификатора файла программных средств от первого узла второму узлу, средство для ожидания отклика от второго узла, средство, реагирующее на прием отклика от второго узла, для выбора из перечня идентификаторов файлов программных средств следующего идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств средством для передачи текущего идентификатора файла программных средств из первого узла во второй узел.

14. Устройство по п.9, отличающееся тем, что средство для передачи идентификаторов файлов программных средств ко второму узлу содержит средство для выбора из перечня идентификаторов файлов программных средств первого идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств, средство для передачи текущего идентификатора файла программных средств от первого узла второму узлу, средство для ожидания, связанное со средством для анализа и вызова и со средством для передачи вызванных файлов программных средств из первого узла ко второму узлу, для ожидания тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны как желательные вторым узлом для вызова и передачи из первого узла ко второму узлу, и для формирования выходного сигнала при завершении ожидания, средство, реагирующее на прием выходного сигнала средства ожидания, для выбора из перечня идентификаторов файлов программных средств следующего идентификатора файла программных средств для использования в качестве текущего идентификатора файла программных средств средством для передачи текущего идентификатора файла программных средств из первого узла во второй узел.

15. Устройство для передачи одного или нескольких желательных файлов программных средств второму узлу в сетевой системе, содержащей первый узел, связанный со вторым узлом, отличающееся тем, что содержит в первом узле средство вызова меню, содержащего перечень идентификаторов файлов программных средств, причем каждый идентификатор файла программных средств идентифицирует соответствующий файл из множества файлов программных средств, и средство для

передачи перечня идентификаторов файлов программных средств из первого узла во второй узел, во втором узле средство для анализа с использованием резидентной программы каждого из идентификаторов файлов программных средств и формирования первого отклика, который включает в себя для каждого из идентификаторов файлов программных средств первую идентификацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, и средство для передачи первого отклика от второго узла к первому узлу, причем первый узел дополнительно содержит средство первого отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были индицированы вторым узлом как желательные, средство для передачи вызванных файлов программных средств из первого узла во второй узел, и средство для передачи во второй раз перечня идентификаторов файлов программных средств от первого узла ко второму узлу, второй узел дополнительно содержит средство для начала выполнения файлов программных средств, которые были приняты от первого узла, средство использования выполняемых файлов программных средств, которые были получены от первого узла, для

анализа каждого из идентификаторов файлов программных средств и формирования второго отклика, который содержит для каждого идентификатора файла программных средств вторую индикацию того, является ли соответствующий файл из множества файлов программных средств желательным или нежелательным, и средство для передачи второго отклика от второго узла к первому узлу, причем первый узел дополнительно содержит средство для анализа второго отклика и вызова тех файлов из множества файлов программных средств, которые соответствуют идентификаторам файлов программных средств, которые были указаны вторым узлом во втором отклике как желательные, и средство для передачи из первого узла второму узлу файлов программных средств, которые были связаны в результате анализа в первом узле второго отклика.

16. Устройство по п.15, отличающееся тем, что каждая из упомянутых первой и второй индикаций, что соответствующие файлы из множества файлов программных средств являются нежелательными, представляют собой индикацию того, что соответствующие файлы из множества файлов программных средств уже загружены во второй узел, или индикацию того, что соответствующие файлы из множества файлов программных средств запрещены к загрузке во второй узел.

30

35

40

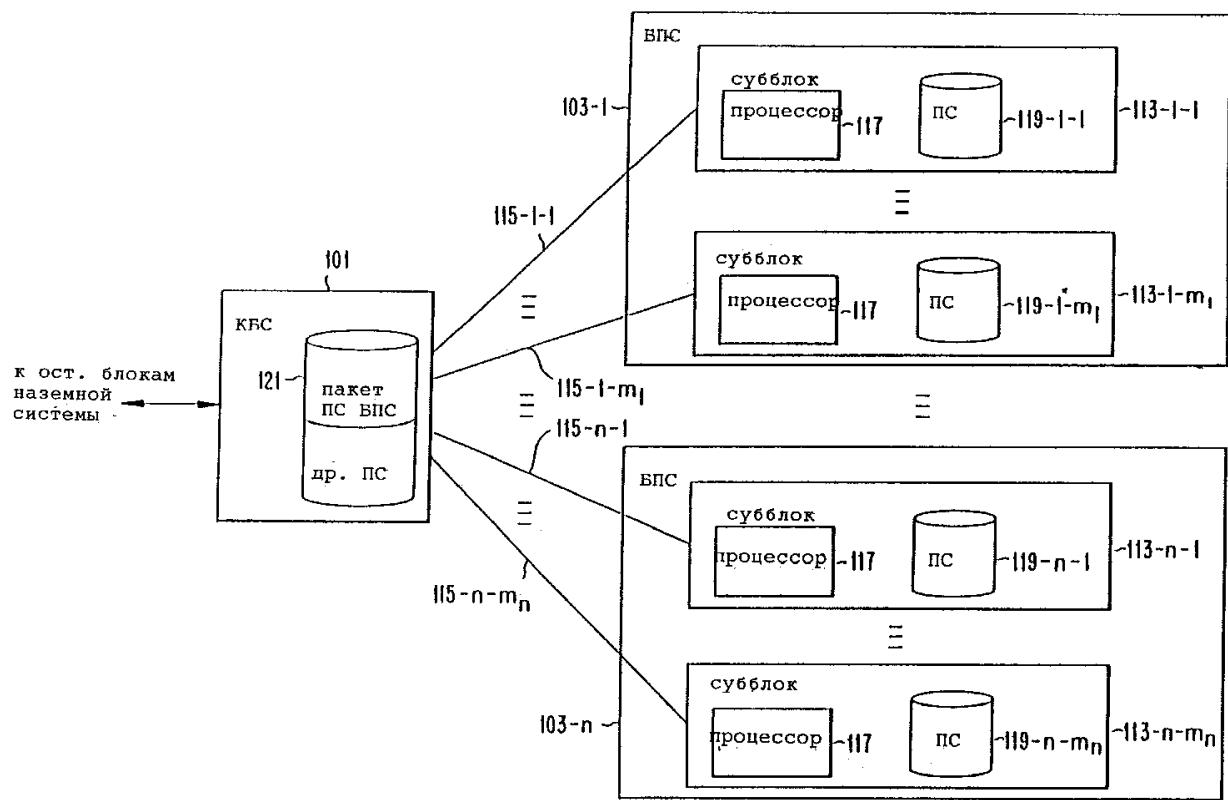
45

50

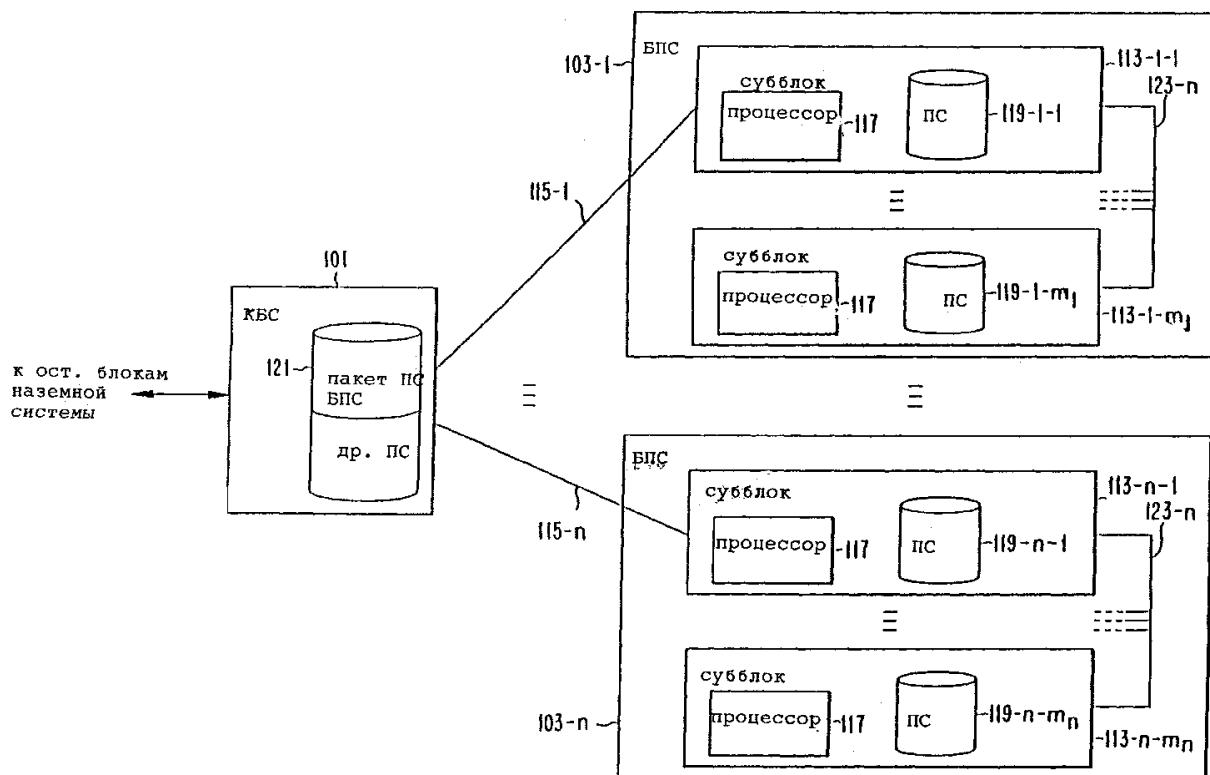
55

60

R U ? 1 5 5 3 7 2 C 2

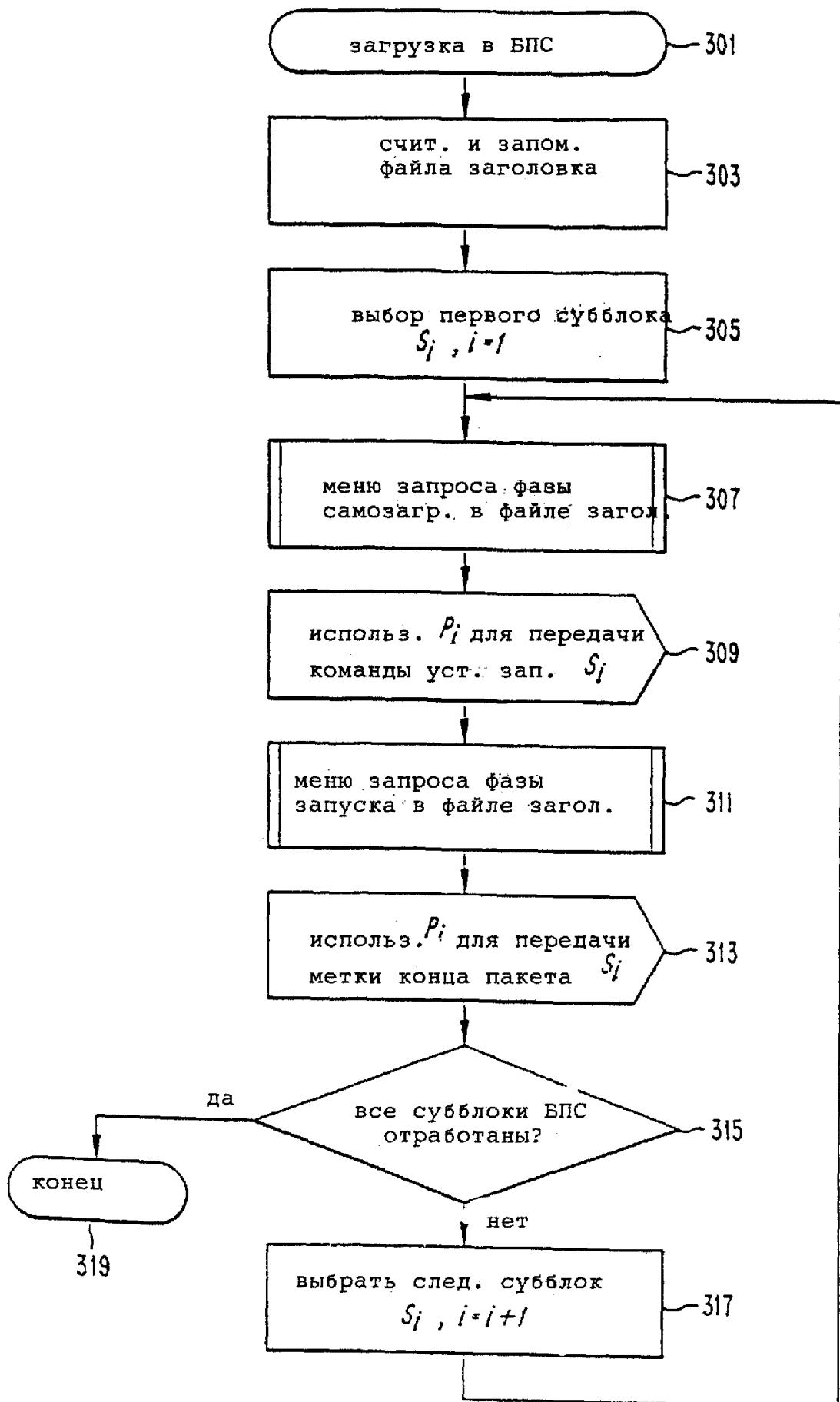


Фиг.1(а)

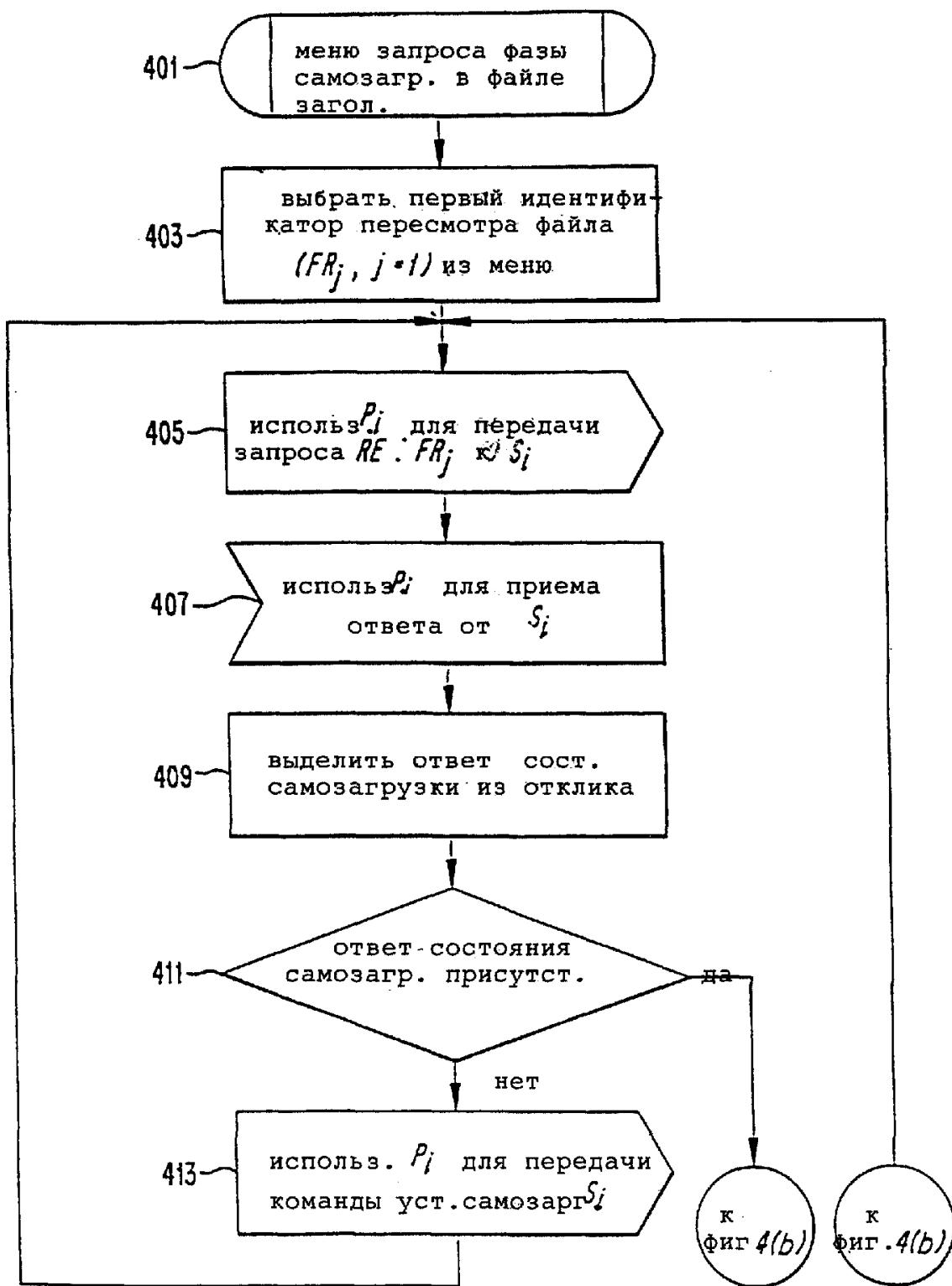


Фиг.1(б)

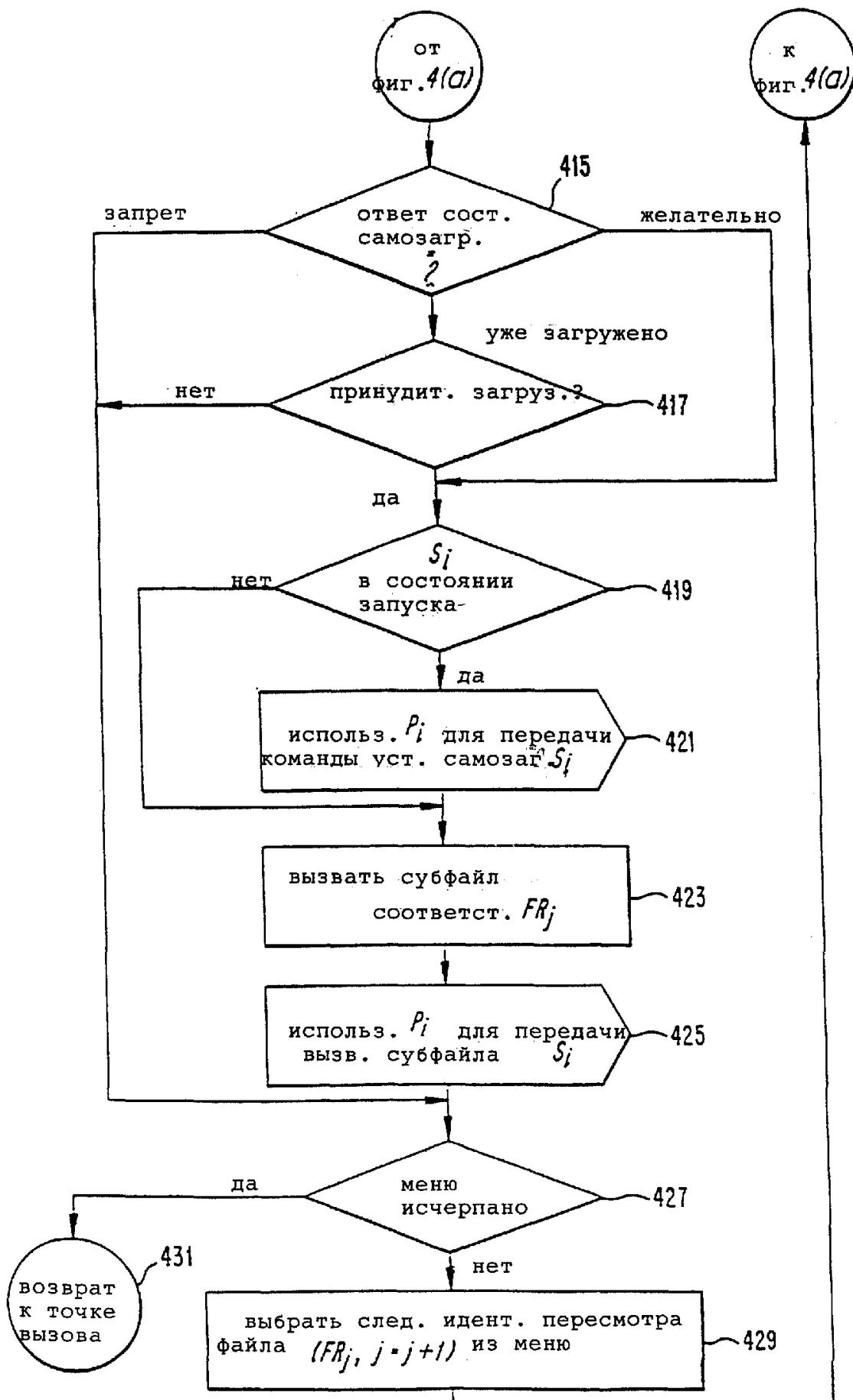
R U 2 1 5 5 3 7 2 C 2



Фиг. 3

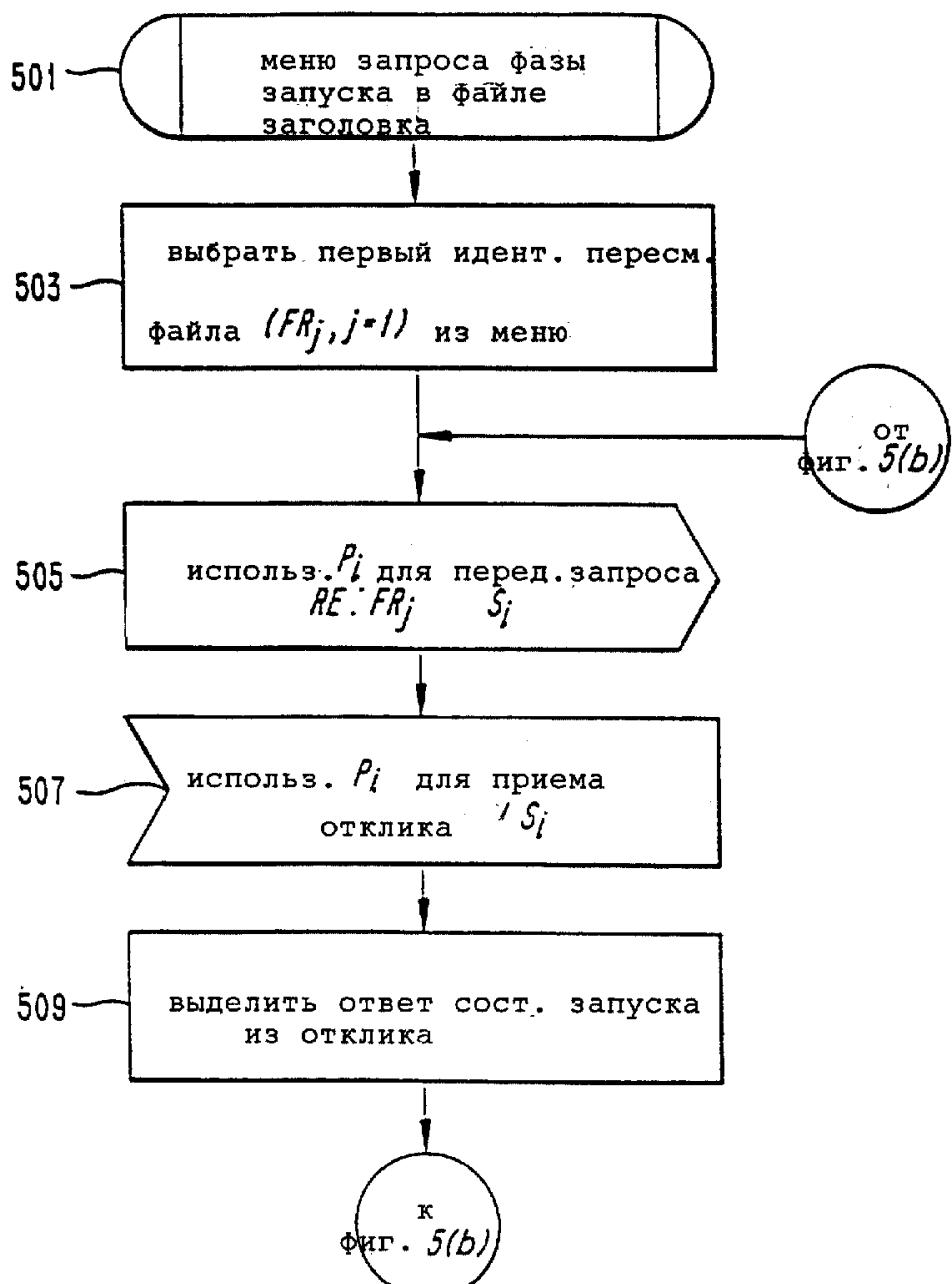


Фиг. 4(а)



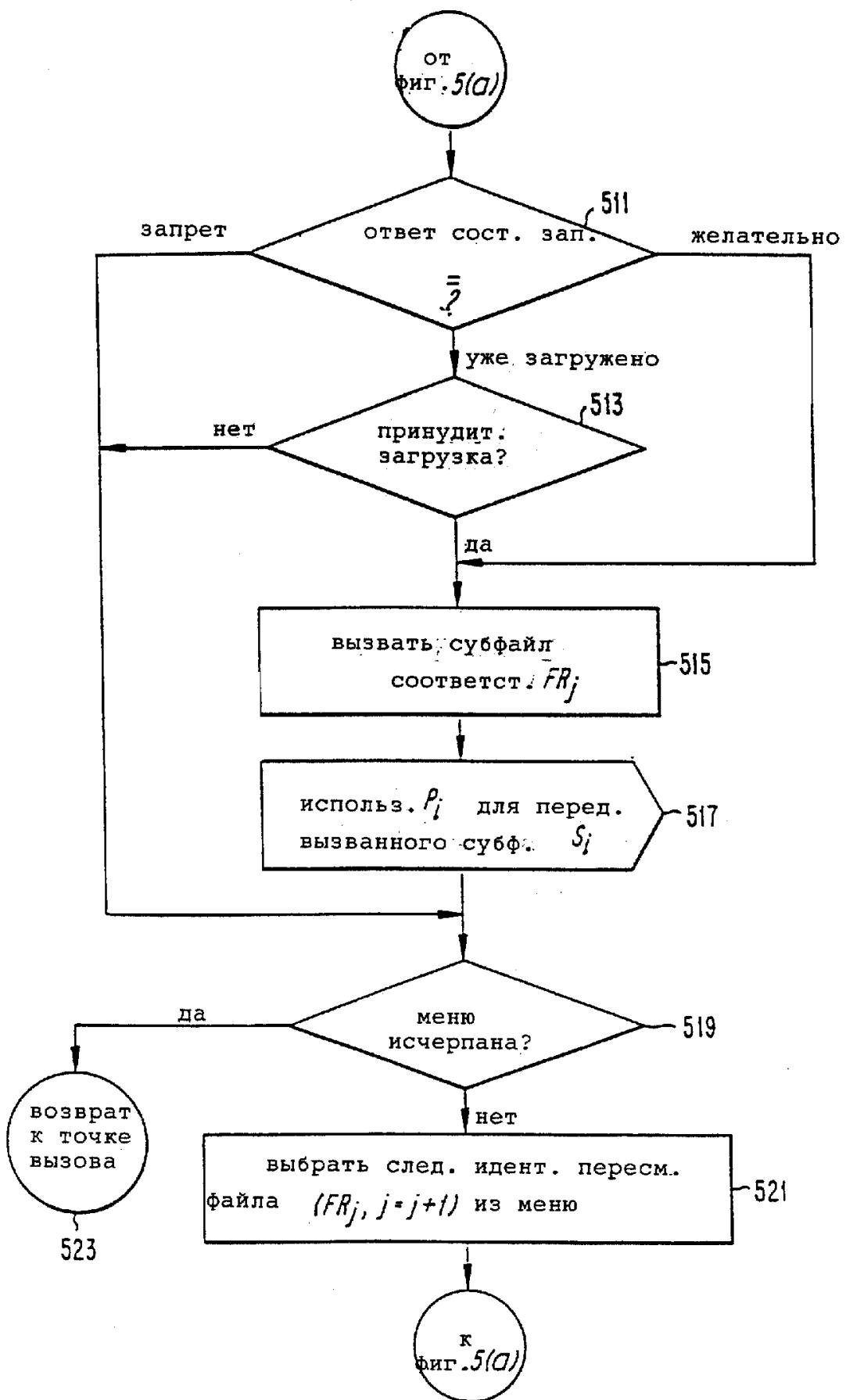
Фиг.4(б)

R U ? 1 5 5 3 7 2 C 2



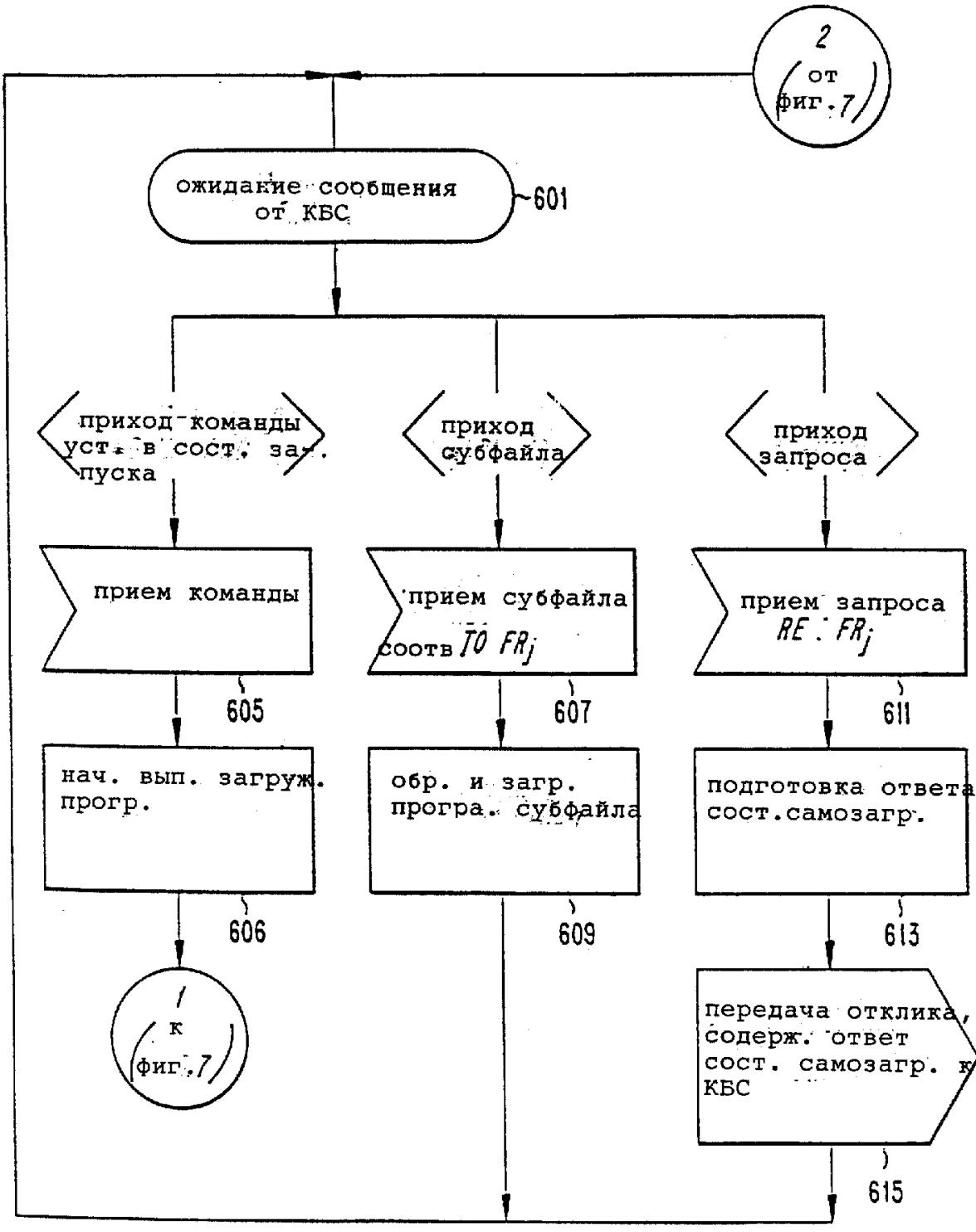
Фиг. 5(а)

R U 2 1 5 5 3 7 2 C 2



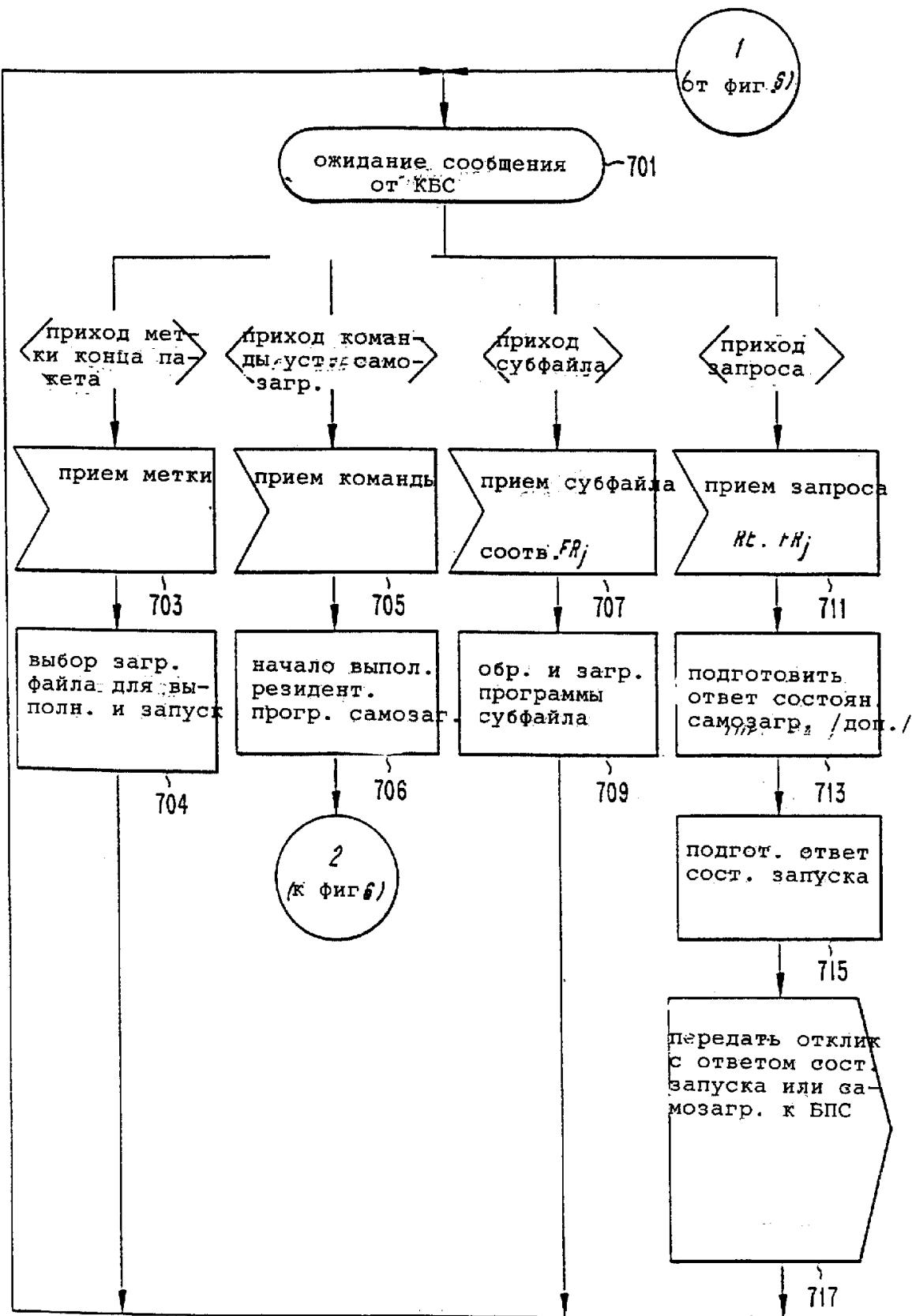
Фиг. 5(b)

R U ? 1 5 5 3 7 2 C 2



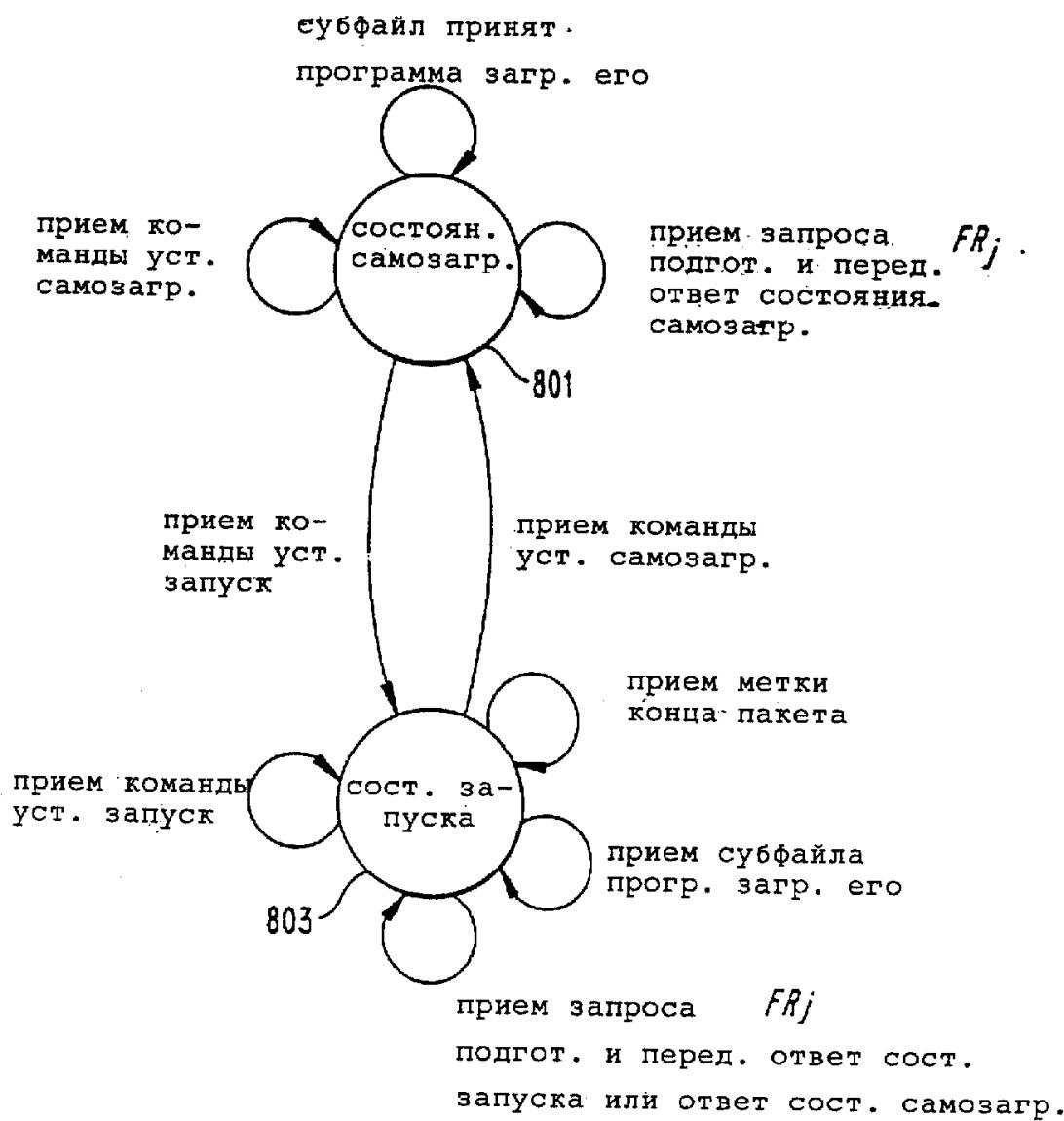
Фиг. 6

R U 2 1 5 5 3 7 2 C 2



Фиг. 7

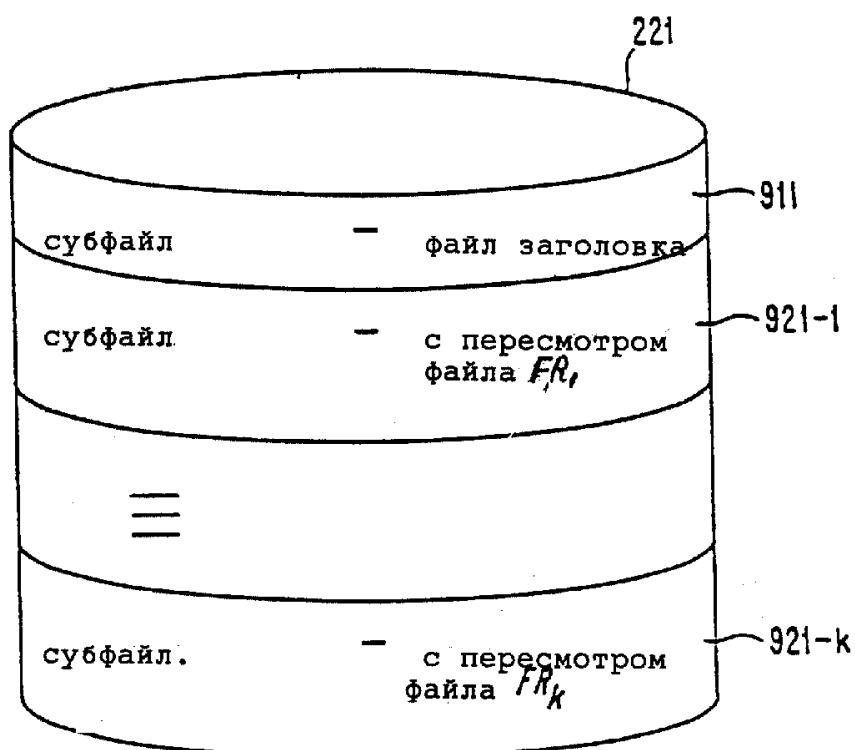
R U ? 1 5 5 3 7 2 C 2



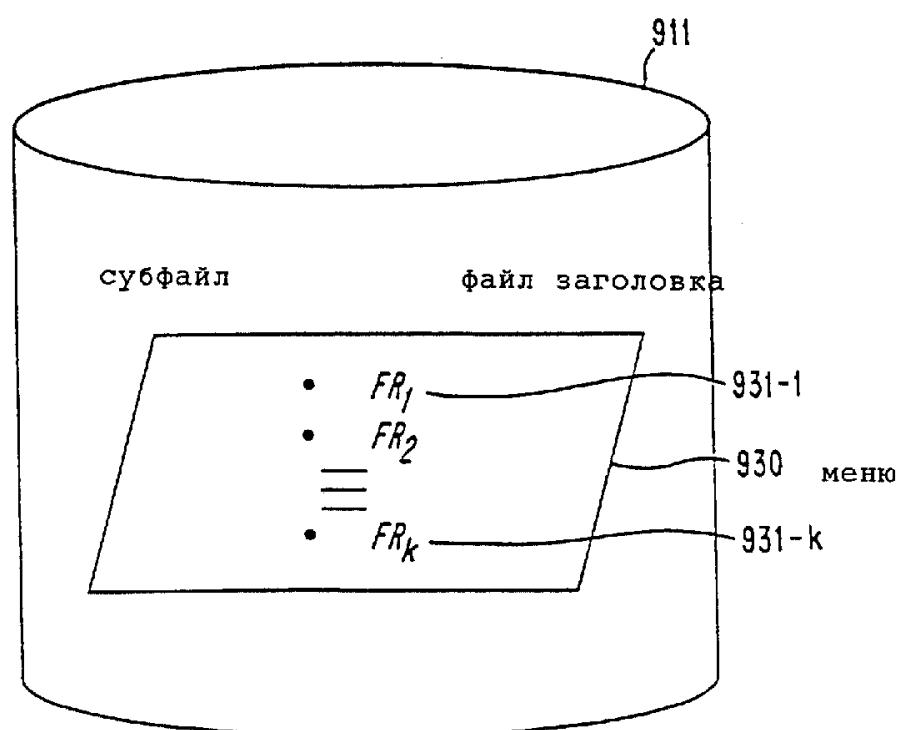
Фиг. 8

R U 2 1 5 5 3 7 2 C 2

R U ? 1 5 5 3 7 2 C 2



Фиг.9(а)



Фиг.9(б)

R U 2 1 5 5 3 7 2 C 2