РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 408 060** (13) **C2**

(51) МПК
*G06F 12/08* (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

# (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2005133383/09, 28.10.2005**

(24) Дата начала отсчета срока действия патента:
**28.10.2005**

(30) Конвенционный приоритет:
**30.11.2004 US 11/000,430**

(43) Дата публикации заявки: **10.05.2007**

(45) Опубликовано: **27.12.2010** Бюл. № **36**

(56) Список документов, цитированных в отчете о поиске: **RU 2159953 C1, 27.11.2000. US 20010020245 A1, 06.09.2001. US 6408298 B1, 18.06.2002. US 6708186 B1, 16.03.2004. US 6473806 B1, 29.10.2002. US 6477544 B1, 05.11.2002. EP 1209556 A2, 29.05.2002.**

Адрес для переписки:
**129090, Москва, ул. Б.Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595**

(72) Автор(ы):
**КРИСТИАНСЕН Нил Р. (US),**
**ТХИНД Равиндер С. (US),**
**ХАВЕВАЛА Сарош Сирус (US)**

(73) Патентообладатель(и):
**МАЙКРОСОФТ КОРПОРЕЙШН (US)**

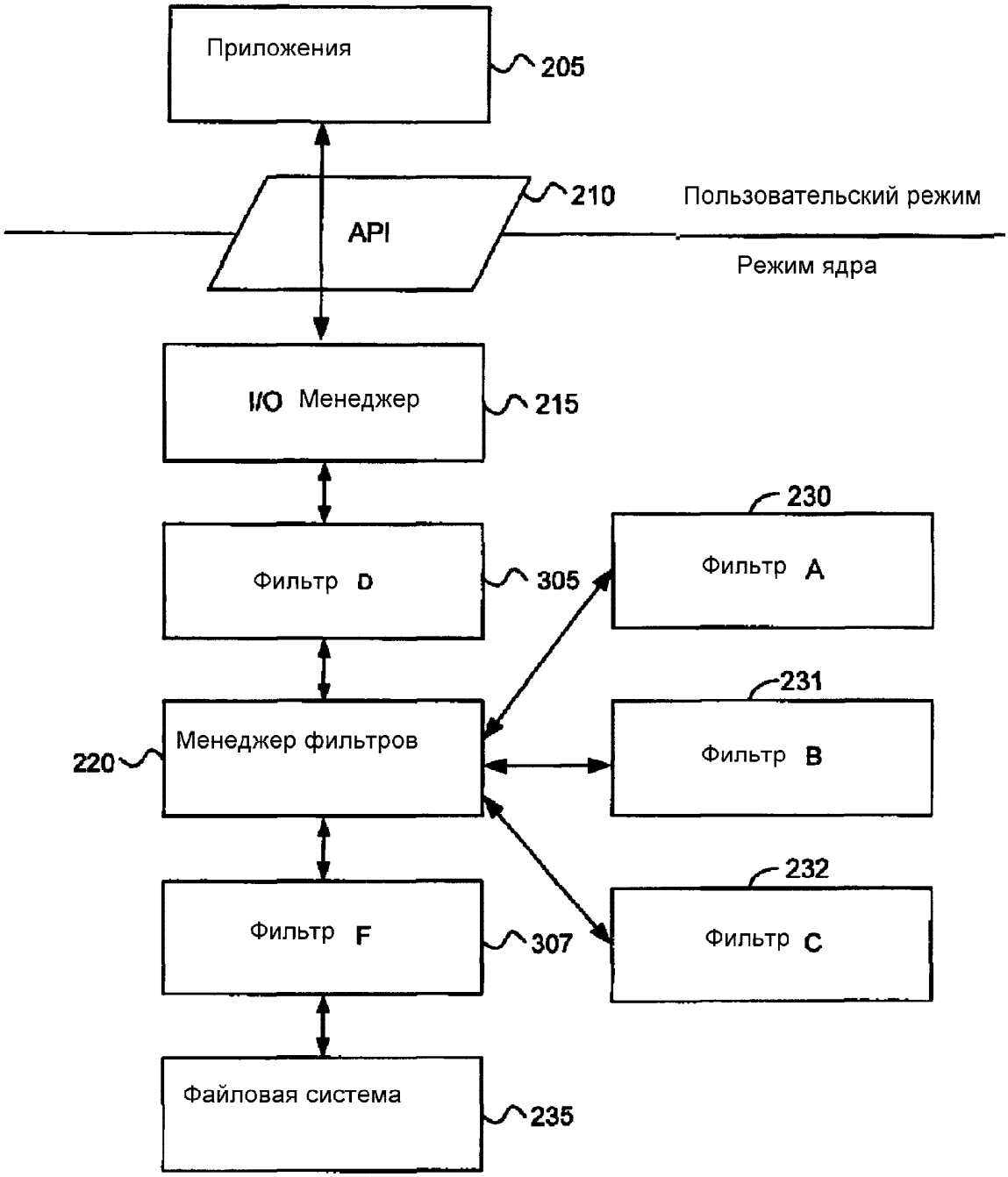
RU 2408060 C2

## (54) СПОСОБ И СИСТЕМА ДЛЯ ПОДДЕРЖАНИЯ СОГЛАСОВАННОСТИ ПРОСТРАНСТВА ИМЕН С ФАЙЛОВОЙ СИСТЕМОЙ

(57) Реферат:

Изобретение относится к вычислительной технике, в частности к способу выполнения фильтром отслеживания запросов ввода/вывода. Техническим результатом является увеличение быстродействия компьютерной системы. Способ включает: создание экземпляра фильтра файловой системы, ассоциированного с файловой системой, регистрируют фильтр с помощью механизма регистрации, реализуемого в средстве управления фильтрами; уведомляют посредством фильтра средство управления фильтрами о по меньшей мере одном типе запроса ввода/вывода, в котором он заинтересован; задают посредством фильтра, должен ли фильтр быть уведомлен о предварительных обратных вызовах для упомянутого по меньшей мере одного типа запроса ввода/вывода; задают посредством фильтра, должен ли фильтр быть уведомлен о последующих обратных вызовах; поддерживают пространство имен, ассоциированное с фильтром, отличное от пространства имен файловой системы; принимают запрос ввода/вывода; определяют, что объект является объектом, представляющим интерес для фильтра; определяют действие фильтра, ассоциированное с принятым запросом ввода/вывода, выполняют это действие; и обновляют пространство имен, ассоциированное с фильтром, на основе принятого запроса ввода/вывода. 2 н. и 20 з.п. ф-лы, 15 ил.

RU 2408060 C2




ФИГ.4

RU 2408060 C2

RU 2408060 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 408 060** (13) **C2**

(51) Int. Cl.
*G06F 12/08* (2006.01)

## (12) ABSTRACT OF INVENTION

(21), (22) Application: **2005133383/09, 28.10.2005**

(24) Effective date for property rights:
**28.10.2005**

(30) Priority:
**30.11.2004 US 11/000,430**

(43) Application published: **10.05.2007**

(45) Date of publication: **27.12.2010 Bull. 36**

Mail address:
**129090, Moskva, ul. B.Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595**

(72) Inventor(s):
**KRISTIANSEN Nil R. (US),**
**TKhIND Ravinder S. (US),**
**KhAVEVALA Sarosh Sirus (US)**

(73) Proprietor(s):
**MAJKROSOFT KORPOREJShN (US)**

RU 2408060 C2

## (54) METHOD AND SYSTEM FOR MAINTAINING CONFORMITY OF NAME SPACE WITH FILE SYSTEM

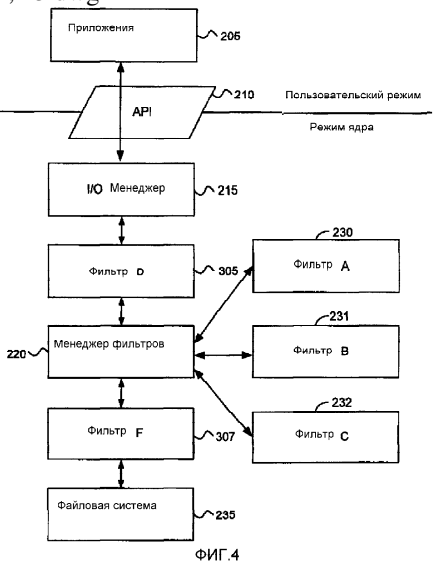
(57) Abstract:

FIELD: information technology.

SUBSTANCE: method involves creation of a copy of a file system filter associated with the file system, recording the filter using a recording mechanism, realised in a filter control apparatus, the filter control apparatus is notified through the filter on at least one type of input/output request, in which it is interested; whether the filter must be notified on preliminary return calls for the said at least one type of input/output request is determined through the filter; whether the filter must be notified on the next return calls is determined through the filter; the name space associated with the filter, different from the name space of the file system, is supported; an input/output request is received; it is determined whether an object is of interest to the filter; action of the filter associated with the received input/output request is determined, and this action is performed; and the name space associated with the filter is updated based on the received input/out request.

EFFECT: high-speed operation of the computer system.

22 cl, 15 dwg




ФИГ.4

RU 2408060 C2

Область техники, к которой относится изобретение

Это изобретение относится в общем к компьютерам и более конкретно к файловым системам.

Предшествующий уровень техники

В современных операционных системах, таких как операционная система Windows XP корпорации Microsoft Corporation, с лежащей в основе файловой системой, такой как Windows NTFS (файловая система Windows NT), FAT, CDFS, редиректорная файловая система SMB или файловые системы WebDav, один или более драйверов фильтров файловой системы могут вставляться между средством управления вводом/выводом (I/O менеджером), которое принимает пользовательские I/O запросы, и драйвером файловой системы. В общем, драйверы фильтров (иногда называемые здесь просто как фильтры) - это процессы, которые усиливают лежащую в основе файловую систему посредством выполнения различных относящихся к файлам вычислительных задач, которые нужны пользователям, включая задачи, такие как передача I/O запросов и данных файловой системы через антивирусное программное обеспечение, средства обеспечения квот файловой системы, репликаторы файлов и продукты шифрования/сжатия.

Например, антивирусные продукты предоставляют фильтр, который наблюдает за I/O в отношении определенных типов файлов (.exe, .doc и подобных), осуществляя поиск сигнатур вирусов, в то время как продукты репликации файлов выполняют зеркальное отражение на уровне файловой системы. Другие типы драйверов фильтров файловой системы нацелены на восстановление системы (которое сохраняет системные файлы, когда вот-вот должны делаться изменения, так что пользователь может возвратиться в исходное состояние), навязывание дисковых квот, резервное копирование открытых файлов, отмену удаления удаленных файлов, шифрование файлов и так далее. Таким образом, посредством установки драйверов фильтров файловой системы пользователи компьютера могут выбирать признаки файловой системы, которые они хотят и которые им нужны, способом, который делает возможным обновления, замену, вставку и удаление компонентов без изменения реально существующей операционной системы или кода драйвера файловой системы.

Фильтр файловой системы может поддерживать внутренние метаданные для файлов и директорий в томе. Изменения в отношении тома, с которым ассоциирован некоторый фильтр, могут стать причиной того, что внутренние метаданные фильтра станут несинхронизированными с состоянием этого тома. Это может служить причиной того, что фильтр будет вести себя некорректно или указывать, что он неспособен выполнять свою требуемую функцию.

То, что необходимо - это способ и система для поддержания согласованности пространства имен между выбранными объектами, поддерживаемыми файловой системой, и фильтром, ассоциированным с ними.

Сущность изобретения

Кратко, настоящее изобретение предоставляет способ и систему для поддержания согласованности пространства имен между выбранными объектами, поддерживаемыми файловой системой, и фильтром, ассоциированным с ней. Процесс запрашивает доступ к некоторому объекту файловой системы. Фильтр отслеживает выбранные типы запросов (или операций, ассоциированных с ними) и определяет, находится ли этот объект внутри пространства имен, ассоциированного с этим фильтром. Пространство имен, ассоциированное с фильтром, обновляется, базируясь на изменении в отношении объекта.

В одном аспекте этого изобретения запрос содержит операцию переименования или удаления. Объект находится внутри пространства имен, ассоциированного с фильтром, если этот объект имеет ассоциированную с ним политику или если этот объект является предком такого объекта.

В другом аспекте этого изобретения операция переименования может служить причиной того, что объект перемещается и/или переименовывается.

В другом аспекте этого изобретения к объекту осуществляется доступ посредством открытия описателя для этого объекта. Описатель содержит информацию, которая идентифицирует канал связи, используемый для доступа к объекту. Может быть установлен флаг, который показывает, что объект должен быть удален, когда будут закрыты все описатели для этого объекта. Если этот флаг остается установленным после того, как будет закрыт последний описатель, объект удаляется.

В одном аспекте этого изобретения поддержание согласованности пространства имен позволяет фильтрам применять политики к объектам, когда объекты переименовываются, и останавливать применение политик к объектам, которые удалены.

Другие аспекты станут видны из последующего подробного описания, взятого в связи с чертежами.

Перечень чертежей

Фиг.1 - блок-схема, представляющая компьютерную систему, в которой может быть реализовано настоящее изобретение;

фиг.2 - блок-схема, представляющая иллюстративную компоновку компонентов системы, в которой настоящее изобретение может работать, в соответствии с различными аспектами этого изобретения;

фиг.3 - блок-схема, представляющая другую иллюстративную компоновку компонентов системы, в которой настоящее изобретение может работать, в соответствии с различными аспектами этого изобретения;

фиг.4 - блок-схема, представляющая другую иллюстративную компоновку компонентов системы, в которой настоящее изобретение может работать, в соответствии с различными аспектами этого изобретения;

фиг.5 - блок-схема, представляющая структуру данных, которая может использоваться, чтобы отслеживать объекты, которые имеют применяемые к ним политики, в соответствии с различными аспектами этого изобретения;

фиг.6 - диаграмма последовательности операций, которая в общем представляет действия, которые могут происходить для поддержания согласованности между пространством имен фильтра и пространством имен файловой системы для интересующих объектов, в соответствии с различными аспектами этого изобретения;

фиг.7 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 630 по фиг.6, которые могут происходить, когда принимается операция переименования, в соответствии с различными аспектами этого изобретения;

фиг.8 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 745 по фиг.7, которые могут происходить при обновлении структуры данных, в соответствии с различными аспектами этого изобретения;

фиг.9 - диаграмма последовательности операций, в общем представляющая действия, которые соответствуют этапу 815 по фиг.8, которые могут происходить при добавлении узлов, когда объект перемещается, в соответствии с различными

аспектами этого изобретения;

фиг.10 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 825 по фиг.8, которые могут происходить при обновлении структуры данных для удаления префиксных узлов, ассоциированных со старым именем объекта, в соответствии с различными аспектами этого изобретения;

фиг.11 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 640 по фиг.6, которые могут происходить для поддержания согласованности между пространством имен фильтра и пространством имен файловой системы в течение относящихся к удалению операций, в соответствии с различными аспектами этого изобретения;

фиг.12 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1115 по фиг.11, которые могут происходить, когда принимается операция создания для удаления при закрытии, в соответствии с различными аспектами этого изобретения;

фиг.13 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 632 по фиг.6, которые могут происходить, когда принимается операция переименования, в соответствии с различными аспектами этого изобретения;

фиг.14 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1135 по фиг.11, которые могут происходить, когда принимается операция задания состояния, в соответствии с различными аспектами этого изобретения; и

фиг.15 - диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1145 по фиг.11, которые могут происходить, когда принимается операция очистки, в соответствии с различными аспектами этого изобретения.

Подробное описание

Иллюстративная операционная среда

Фиг.1 показывает пример подходящей среды 100 вычислительной системы, в которой может быть реализовано это изобретение. Среда 100 вычислительной системы является только одним примером подходящей вычислительной среды и не предназначается, чтобы предлагать какое-либо ограничение в отношении объема использования или функциональных возможностей этого изобретения. Вычислительная среда 100 не должна интерпретироваться как имеющая какую-либо зависимость или требование, относящиеся к какому-либо одному компоненту или комбинации компонентов, показанных в иллюстративной операционной среде 100.

Это изобретение является работоспособным с многочисленными другими средами или конфигурациями вычислительных систем общего назначения или специального назначения. Примеры широко известных вычислительных систем, сред и/или конфигураций, которые могут быть подходящими для использования с этим изобретением, включают в себя, но не ограничены этим, персональные компьютеры, серверные компьютеры, ручные или портативные устройства, многопроцессорные системы, базирующиеся на микроконтроллерах системы, приставки к телевизору, программируемую бытовую электронику, сетевые PC, миникомпьютеры, компьютеры-мэйнфреймы, распределенные вычислительные среды, которые включают в себя любые из вышеперечисленных систем или устройств, и подобное.

Это изобретение может описываться в общем контексте машиноисполняемых инструкций, таких как программные модули, которые исполняются компьютером. В

общем, программные модули включают в себя процедуры, программы, объекты, компоненты, структуры данных и так далее, которые выполняют конкретные задачи или реализуют определенные абстрактные типы данных. Это изобретение может также применяться на практике в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки данных, которые соединены через сеть связи. В распределенной вычислительной среде программные модули могут располагаться как в локальных, так и в удаленных компьютерных носителях данных, включая запоминающие устройства.

Со ссылкой на фиг.1, иллюстративная система для реализации этого изобретения включает в себя вычислительное устройство общего назначения в форме компьютера 110. Компоненты компьютера 110 могут включать в себя, но не ограничены этим, устройство 120 обработки данных, системную память 130 и системную шину 121, которая соединяет различные компоненты системы, включая системную память, с устройством 120 обработки данных. Системная шина 121 может относиться к любому из нескольких типов структур шин, включая шину памяти или контроллер памяти, периферийную шину и локальную шину, используя любую из многообразия архитектур шин. В качестве примера, и не ограничения, такие архитектуры включают в себя шину промышленной стандартной архитектуры (ISA), шину микроканальной архитектуры (MCA), шину расширенной ISA (EISA), локальную шину ассоциации по стандартам в области видеоэлектроники (VESA) и шину межсоединения периферийных компонент (PCI), также известную как шина расширения.

Компьютер 110 обычно включает в себя многообразие машиночитаемых носителей. Машиночитаемые носители могут быть любыми доступными носителями, к которым компьютер 110 может осуществлять доступ, и включают в себя как энергозависимые, так и энергонезависимые носители, как съемные, так и несъемные носители. В качестве примера, и не ограничения, машиночитаемые носители могут содержать компьютерные носители данных и среды передачи данных. Компьютерные носители данных включают в себя как энергозависимые, так и энергонезависимые, съемные и несъемные носители, реализованные любым способом или технологией для хранения информации, такой как машиночитаемые инструкции, структуры данных, программные модули или другие данные. Компьютерные носители данных включают в себя, но не ограничены этим, ОЗУ, ПЗУ, электрически стираемое программируемое ПЗУ (EEPROM), флеш-память или память другой технологии, ПЗУ на компакт диске (CD-ROM), универсальные цифровые диски (DVD) или другое оптическое дисковое хранилище, магнитные кассеты, магнитную ленту, магнитное дисковое хранилище или другие магнитные хранящие устройства, или любой другой носитель, который может использоваться для хранения нужной информации и к которому компьютер 110 может осуществлять доступ. Среды передачи данных обычно воплощают машиночитаемые инструкции, структуры данных, программные модули или другие данные в модулированном информационном сигнале, таком как несущее колебание, или другом транспортном механизме и включают в себя любые среды доставки информации. Термин «модулированный информационный сигнал» означает сигнал, одна или более характеристик которого установлены или изменены таким образом, чтобы кодировать информацию в этом сигнале. В качестве примера, и не ограничения, среды передачи данных включают в себя проводные среды, такие как проводная сеть или прямое проводное соединение, и беспроводные среды, такие как акустические, радиочастотные, инфракрасные и другие беспроводные среды.

Комбинации любых из вышеперечисленных сред и носителей также охватываются понятием «машиночитаемый носитель».

Системная память 130 включает в себя компьютерные носители данных в форме энергозависимой и/или энергонезависимой памяти, такой как постоянное запоминающее устройство (ПЗУ) 131 и оперативное запоминающее устройство (ОЗУ) 132. Базовая система 133 ввода/вывода (BIOS), содержащая базовые процедуры, которые помогают переносить информацию между элементами внутри компьютера 110, как в течение запуска, обычно хранится в ПЗУ 131. ОЗУ 132 обычно содержит данные и/или программные модули, которые являются немедленно доступными для устройства по обработке данных и/или в настоящем обрабатываются им. В качестве примера, и не ограничения, фиг.1 показывает операционную систему 134, прикладные программы 135, другие программные модули 136 и данные 137 программ.

Компьютер 110 может также включать в себя другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители данных. Только в качестве примера, фиг.1 показывает накопитель 140 на жестких дисках, который осуществляет считывание или запись в отношении несъемных, энергонезависимых магнитных носителей, магнитный дисковод 151, который осуществляет считывание или запись в отношении съемного, энергонезависимого магнитного диска 152, и оптический дисковод 155, который осуществляет считывание или запись в отношении съемного, энергонезависимого оптического диска 156, такого как CD-ROM или другие оптические носители. Другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители данных, которые могут использоваться в иллюстративной операционной среде, включают в себя, но не ограничены этим, кассеты магнитной ленты, платы флеш-памяти, универсальные цифровые диски, цифровую видеоленту, твердотельное ОЗУ, твердотельное ПЗУ и подобное. Накопитель 141 на жестких дисках обычно подсоединяется к системной шине 121 через интерфейс несъемной памяти, такой как интерфейс 140, а магнитный дисковод 151 и оптический дисковод 155 обычно подсоединяются к системной шине 121 посредством интерфейса съемной памяти, такого как интерфейс 150.

Дисководы и накопители и их ассоциированные компьютерные носители данных, обсуждавшиеся выше и показанные на фиг.1, предоставляют хранилище машиночитаемых инструкций, структур данных, программных модулей и других данных для компьютера 110. На фиг.1, например, накопитель 141 на жестких дисках показан как хранящий операционную систему 144, прикладные программы 145, другие программные модули 146 и данные 147 программ. Заметим, что эти компоненты могут быть либо такими же как, либо отличающимися от операционной системы 134, прикладных программ 135, других программных модулей 136 и данных 137 программ. Операционной системе 144, прикладным программам 145, другим программным модулям 146 и данным 147 программ здесь даны другие ссылочные позиции, чтобы показать, что, по меньшей мере, они являются другими копиями. Пользователь может вводить команды и информацию в компьютер 20 через устройства ввода, такие как клавиатура 162 и координатно-указательное устройство 161, обычно упоминаемое как мышь, шаровой манипулятор или сенсорная панель. Другие устройства ввода (не показаны) могут включать в себя микрофон, джойстик, игровую приставку, спутниковую параболическую антенну, сканер, чувствительный к прикосновениям экран ручного PC или другого записывающего планшета или подобное. Эти и другие устройства ввода часто подсоединяются к

устройству 120 обработки данных через интерфейс 160 пользовательского ввода, который соединен с системной шиной, но могут подсоединяться посредством других структур интерфейсов и шин, таких как параллельный порт, игровой порт или универсальная последовательная шина (USB). Монитор 191 или другой тип устройства отображения также подсоединяется к системной шине 121 через некоторый интерфейс, такой как видеоинтерфейс 190. В дополнение к монитору компьютеры могут также включать в себя другие периферийные устройства вывода, такие как громкоговорители 197 и принтер 196, который может подсоединяться через периферийный интерфейс 190 вывода.

Компьютер 110 может работать в сетевой среде, используя логические соединения с одним или более удаленными компьютерами, такими как удаленный компьютер 180. Удаленный компьютер 180 может быть персональным компьютером, сервером, маршрутизатором, сетевым PC, одноранговым устройством или другим общим сетевым узлом и обычно включает в себя многие или все из элементов, описанных выше по отношению к компьютеру 110, хотя только запоминающее устройство 181 показано на фиг.1. Логические соединения, изображенные на фиг.1, включают в себя локальную сеть (LAN) 171 и глобальную сеть (WAN) 173, но могут также включать в себя другие сети. Такие сетевые среды являются обычным явлением в офисах, компьютерных сетях масштаба предприятия, внутренних сетях и сети Интернет.

При использовании в сетевой среде LAN компьютер 110 подсоединяется к LAN 171 через сетевой интерфейс или адаптер 170. При использовании в сетевой среде WAN компьютер 110 обычно включает в себя модем 172 или другое средство для установления связи через WAN 173, такую как сеть Интернет. Модем 172, который может быть внутренним или внешним, может подсоединяться к системной шине 121 через интерфейс 160 пользовательского ввода или другой соответствующий механизм. В сетевой среде программные модули, изображенные по отношению к компьютеру 110, или их части могут храниться в удаленном запоминающем устройстве. В качестве примера, и не ограничения, фиг.1 показывает удаленные прикладные программы 185 как постоянно находящиеся в запоминающем устройстве 181. Следует принять во внимание, что показанные сетевые соединения являются иллюстративными и может использоваться другое средство установления линии связи между компьютерами.

Поддержание согласованности пространства имен

Фиг.2 - это блок-схема, представляющая иллюстративную компоновку компонентов системы, в которой может работать настоящее изобретение, в соответствии с различными аспектами этого изобретения. Эти компоненты включают в себя одно или более приложений 205, интерфейс 210 прикладного программирования (API), менеджер 215 ввода/вывода (I/O менеджер), менеджер 220 фильтров, файловую систему 225 и один или более фильтров 230-232.

Приложения 205 могут делать запросы файловой системы (например, через вызовы функции/метода) через API 210 к I/O менеджеру 215. I/O менеджер 215 может определять, какой I/O запрос или запросы должны выдаваться, чтобы исполнить каждый запрос и отправлять каждый I/O запрос менеджеру 220 фильтров. I/O менеджер 215 также может возвращать данные приложениям 205 по мере того, как операции, ассоциированные с запросами файловой системы, продолжаются, завершаются или преждевременно прекращаются.

В одном варианте осуществления фильтры содержат объекты или подобное, которые при создании их экземпляров регистрируются (например, в течение

процедуры их инициализации) с помощью механизма регистрации в менеджере 220 фильтров. Для эффективности каждый фильтр обычно регистрируется только для запросов файловой системы, в которых он может быть заинтересован при обработке. С этой целью, в качестве части регистрации, каждый фильтр уведомляет менеджер 220 фильтров о типах I/O запросов, в которых он заинтересован (например, создания, чтения, записи, закрытия, переименования и так далее). Например, фильтр шифрования может регистрироваться для чтения и записи вводов/выводов, но не для других, причем данные не должны шифроваться или дешифроваться. Аналогично, фильтр квот может интересоваться только созданием объектов и записью объектов.

В дополнение к определению типов I/O запросов, в которых он заинтересован, фильтр может дополнительно определять, должен ли этот фильтр уведомляться в отношении предварительных обратных вызовов и последующих обратных вызовов для каждого из типов I/O. Предварительный обратный вызов вызывается, когда данные, ассоциированные с I/O запросом, проходят от I/O менеджера 215 к файловой системе 225, в то время как последующий обратный вызов вызывается в течение завершения I/O запроса, когда данные, ассоциированные с этим I/O запросом, проходят из файловой системы 225 к I/O менеджеру 215.

Из каждого I/O запроса менеджер 220 фильтров может создавать структуру данных в едином формате, подходящем для использования фильтрами 230-232. Ниже эта структура данных иногда указывается как данные обратного вызова. Затем менеджер 220 фильтров может вызывать и передавать данные обратного вызова каждому фильтру, который зарегистрировался, чтобы принимать обратные вызовы для типа I/O, принятого менеджером 220 фильтров. Любые фильтры, зарегистрированные, чтобы принимать обратные вызовы для типа вводов/выводов, принятых менеджером фильтров, иногда указываются как зарегистрированные фильтры.

Обычно менеджер 220 фильтров передает данные обратного вызова, ассоциированные с конкретным типом I/O запроса, каждому зарегистрированному фильтру последовательно в порядке, в котором зарегистрированные фильтры упорядочены. Например, если фильтры 230 и 232 зарегистрированы, чтобы принимать обратные вызовы для всех I/O запросов чтения, и упорядочены так, что фильтр 230 находится перед фильтром 232 при обработке таких запросов, то после приема прочитанного I/O менеджер 220 фильтров может сначала вызвать и передать данные обратного вызова фильтру 230, и после того, как фильтр 230 обработает эти данные обратного вызова, менеджер 220 фильтров затем может вызвать и передать эти данные обратного вызова (как модифицированные, если вообще передает) фильтру 232.

Фильтр может быть прикреплен к одному или более томам. То есть фильтр может быть зарегистрирован, чтобы вызываться и принимать данные обратного вызова для вводов/выводов, относящихся только к одному или более чем одному тому.

Фильтр может сгенерировать свой собственный I/O запрос, который затем может передаваться другим фильтрам. Например, антивирусный фильтр может желать прочитать файл до его открытия. Фильтр может остановить I/O запрос от дальнейшего прохождения и может инструктировать менеджер фильтров докладывать код состояния (например, успех или неудачу) для этого I/O запроса. Фильтр может хранить данные в памяти и сохранять эти данные на диск. В общем, фильтр может создаваться, чтобы выполнять любое множество действий, которые могут выполняться процессом режима ядра или пользовательского режима, и может

являться реагирующим (например, ждать, когда он примет I/O запросы до осуществления действий) и/или упреждающим (например, инициировать свои собственные I/O запросы или выполнять другие действия асинхронно с I/O запросами, обрабатываемыми I/O менеджером 215).

В одном варианте осуществления фильтры могут быть организованы стековым способом, как показано на фиг.3, которая является блок-схемой, представляющей другую иллюстративную компоновку компонентов системы, в которой настоящее изобретение может работать, в соответствии с различными аспектами этого изобретения. В этом варианте осуществления каждый из фильтров 305-307 может обрабатывать I/O запросы и передавать эти запросы (модифицированные или немодифицированные) другому фильтру или другому компоненту в стеке. Например, в ответ на запрос чтения, принятый от одного из приложений 205, I/O менеджер 215 может выдать I/O запрос и отправить этот запрос фильтру 305. Фильтр 305 может проанализировать этот I/O запрос и определить, что фильтр 305 не интересуется этим I/O запросом, и затем передать этот I/O запрос неизмененным фильтру 306. Фильтр 306 может определить, что фильтр 306 выполнит некоторое действие, базируясь на этом I/O запросе, и затем может передать этот I/O запрос (измененный или неизмененный) фильтру 307. Фильтр 307 может определить, что фильтр 307 не интересуется этим I/O запросом, и передать этот I/O запрос файловой системе 235.

После того как файловая система 235 обслужит I/O запрос, она передает результаты фильтру 307. Обычно результаты передаются в порядке, обратном тому, в котором I/O запрос проходил (например, сначала к фильтру 307, затем к фильтру 306 и затем к фильтру 305). Каждый из фильтров 305-307 может анализировать результаты, определить, заинтересован ли фильтр в этих результатах, и может выполнять действия, базируясь на этом, до передачи результатов (измененных или неизмененных) далее другому фильтру или компоненту.

В другом варианте осуществления этого изобретения фильтры могут быть организованы стековым/управляемым способом, как показано на фиг.4, которая является блок-схемой, представляющей другую иллюстративную компоновку компонентов системы, в которой настоящее изобретение может работать, в соответствии с различными аспектами этого изобретения. В этой конфигурации некоторые фильтры ассоциированы с менеджером фильтров, в то время как другие фильтры нет. Менеджер 220 фильтров помещается в стек с другими фильтрами (например, фильтрами 305 и 307).

Должно быть понятно, что фильтры могут быть реализованы во многих других конфигурациях без отхода от сущности или объема этого изобретения. В некоторых вариантах осуществления фильтр содержит любой объект, который анализирует I/O между приложением и файловой системой и который способен изменять, совершать или преждевременно прекращать I/O или выполнять другие действия, базирующиеся на этом анализе.

Возвращаясь к фиг.2, файловая система 235 может включать в себя один или более томов, которые могут располагаться локально или удаленно для машины или машин, на которых исполняются приложения 205.

Некоторые фильтры отслеживают файлы в определенных директориях. Например, фильтр квот может навязывать квоты в конкретных директориях. В качестве другого примера, фильтр-экран данных может отказывать в создании определенных типов файлов (например, файлов, которые могут охраняться авторским правом, что показывается с помощью расширения, как например .MP3, .JPG и подобных) в

определенных директориях. Эти фильтры могут включать в себя пространство имен, которое идентифицирует вовлеченные директории.

Определенные запросы файловой системы (например, переименования и удаления) могут изменять пространство имен файловой системы для объектов, отслеживаемых фильтром. Например, фильтр квот может быть сконфигурирован так, чтобы навязывать квоту, равную одному гигабайту для объекта, называемого C:\DIR\QUOTA.

Имя объекта и квота, применяемая к этому объекту, могут храниться в файле метаданных, который сохраняется в энергонезависимом хранилище. Приложение может переименовать объект C:\DIR в C:\DIR2 или может переместить C:\DIR\QUOTA в C:\QUOTA. Чтобы продолжать навязывать квоту для объекта QUOTA, фильтр квот отслеживает переименования и обновляет свой файл метаданных каждый раз, когда переименование затрагивает какой-либо объект, для которого этот фильтр квот навязывает какую-либо квоту.

В некоторых операционных системах операция переименования может переименовывать объект и/или перемещать объект. Таким образом, посредством отслеживания операций переименования, фильтр может фиксировать либо переименование имени, либо перемещение какого-либо объекта, или и то, и другое. В операционных системах, в которых переименование файла и перемещение файла являются отдельными операциями, фильтр может иметь необходимость отслеживать обе эти операции, чтобы поддерживать согласованность пространства имен с пространством имен файловой системы.

В дополнение, приложение может удалить объект, например, C:\DIR или C:\DIR\QUOTA. Чтобы избежать отслеживания объектов, которые были удалены, не следует навязывать политики, когда существующий объект удаляется, и на его месте создается новый объект с тем же именем, и чтобы уменьшить размер метаданных, используемых в отслеживании таких объектов, фильтр может отслеживать удаления и обновлять свои метаданные соответственно.

Фиг.5 - это блок-схема, представляющая структуру данных, которая может использоваться, чтобы отслеживать объекты, которые имеют применяемые к ним политики, в соответствии с различными аспектами этого изобретения. Как здесь указывается, отслеживаемые объекты содержат директории, файлы, все что угодно, что может сохраняться в файловой системе, и подобное.

Структура данных включает в себя узел для каждого интересующего объекта и иногда указывается как префиксное дерево. Представленные иллюстративные объекты включают в себя корневую директорию \, \DIR1, \DIR1\QUOTA1, \DIR1\QUOTA1\QUOTA2, \DIR2, \DIR2\DIR3 и \DIR2\DIR3\QUOTA3. Объекты \DIR1\QUOTA1, \DIR1\QUOTA1\QUOTA2 и \DIR2\DIR3\QUOTA3 имеют применяемые к ним политики P1, P2 и P3 соответственно. Политика может содержать квоту на дисковое пространство, используемое каким-либо объектом и его потомками (например, поддиректориями), файлами, разрешенными в объекте и его потомках, что позволяется и/или не позволяется по отношению к объекту и его потомкам и подобное. Когда в файловой системе интересующий объект переименовывается или удаляется, может требоваться обновить структуру данных, показанную на фиг.5. Такие обновления могут включать в себя удаление узлов, добавление дополнительных узлов и изменение содержимого узлов, как описывается более детально ниже.

В общем, объект представляет интерес, если переименование или удаление этого

объекта повлияет на обнаружение этого объекта в структуре данных или применение некоторой политики. Например, если \DIR2 и ее поддиректории были удалены, политика P3 больше не должна применяться к \DIR2\DIR3\QUOTA3. Более того, узлы DIR2, DIR3 и QUOTA3 больше не должны применять политику P3 и могут быть удалены, чтобы сохранить пространство и уменьшить время поиска для любых оставшихся политик.

В качестве другого примера, если в файловой системе \DIR1\QUOTA1\QUOTA2 переименовывается в \QUOTA5, это может повлиять на поиск нового объекта в структуре данных. Это важно, так как перед тем, как фильтр применит некоторую политику к конкретной операции, этот фильтр должен знать, может ли операция затронуть какие-либо объекты, для которых были установлены политики. Чтобы найти новый объект в структуре данных (например, чтобы фильтр мог определить, должен ли он навязывать какую-либо политику для некоторой конкретной операции), может создаваться и ассоциироваться с этим новым объектом новый узел, затем политики, ассоциированные со старым узлом QUOTA2 (например, P2), могут копироваться в этот новый узел, и затем старый узел QUOTA2 может быть удален. Затем для этого нового объекта могут навязываться политики P2. Добавление и удаление узлов в ответ на изменения в отношении объектов описываются более детально в связи с фиг.8-10.

Приложение (например, служба или что-либо другое) может сообщать фильтру, какие объекты имеют ассоциированные с ними политики. Приложение может предоставлять список таких объектов или может добавлять и/или удалять политики, насколько необходимо. После приема такого списка или инструкции фильтр может создать новую структуру данных или модифицировать существующую структуру данных, чтобы учесть интересующие объекты.

Структура данных может реконструироваться во время начальной загрузки или запуска экземпляра фильтра, который использует эту структуру данных. Это может делаться при использовании метафайла, описанного выше, который хранится в энергонезависимом хранилище (например, диске).

Структура данных может совместно использоваться между более чем одним фильтром. Например, два или более фильтра могут навязывать политики объектам, которые могут переименовываться или удаляться. Вместо поддержания отдельной копии структуры данных для каждого фильтра, эти два или более фильтров могут совместно использовать структуру данных. Один или более из этих фильтров могут назначаться для содержания структуры данных в синхронизме с файловой системой.

Более того, даже хотя фильтры могут совместно использовать структуру данных, они могут не желать отслеживать все одни и те же объекты. С этой целью могут поддерживаться дополнительные данные, которые показывают, в каких узлах каждый фильтр заинтересован.

Фиг.6 - это диаграмма последовательности операций, которая в общем представляет действия, которые могут происходить, чтобы поддерживать согласованность между пространством имен фильтра и пространством имен файловой системы для интересующих объектов, в соответствии с различными аспектами этого изобретения. На этапе 605 процесс начинается. На этапе 610 фильтр принимает интересующую I/O операцию. В случае управляемых фильтров, фильтр может предварительно зарегистрироваться, чтобы принимать I/O операции принятого типа. В случае стековых фильтров фильтр может определять, что I/O является тем, который может затрагивать пространство имен файловой системы. В одном варианте

осуществления некоторые I/O операции, которые могут затрагивать пространство имен файловой системы, включают в себя переименование и удаление.

На этапе 615 делается определение в отношении того, относится ли эта I/O операция к интересующему объекту. Если это так, процесс ответвляется на этап 625; иначе, процесс ответвляется на этап 620, где процесс завершается. Ссылаясь на фиг.5, интересующий объект - это объект, представленный любым узлом структуры данных. После приема данных, ассоциированных с интересующей I/O операцией, фильтр затем может выполнить поиск по структуре данных, чтобы определить, затрагивает ли эта I/O операция какие-либо узлы. Если нет (например, поиск терпит неудачу), фильтр может выдать результат.

На этапе 625 делается определение в отношении того, является ли I/O операция операцией переименования. Если это так, процесс ответвляется на этап 630; иначе, процесс ответвляется на этап 635. В другом варианте осуществления действия, ассоциированные с этапами 615 и 610, меняют направление на обратное. То есть сначала делается определение в отношении того, что I/O операция включает в себя интересующий объект, и затем делается определение в отношении того, является ли эта I/O операция интересующей. В этом варианте осуществления, если I/O не затрагивает интересующий объект или если этот I/O не является интересующим I/O, процесс завершается.

На этапе 630 пространство имен фильтра обновляется по обстоятельствам, как описывается более детально в связи с фиг.7-10. На этапе 632 контекст потока, ассоциированный с объектом, обновляется, если контекст потока существует, как описано более детально в связи с фиг.13. Контекст потока содержит данные, в которых может храниться и извлекаться информация об объекте. Контекст потока может создаваться для объекта, например, когда объект был отмечен для удаления. Даже хотя объект может быть отмечен для удаления, он может переименовываться до его удаления. Чтобы отследить имя объекта сквозь переименования, новое имя объекта сохраняется в контексте потока. Каждый раз как объект переименовывается, контекст потока обновляется, чтобы включать в себя новое имя объекта. Затем, когда объект в конечном счете удаляется, фильтр ищет наиболее недавнее имя этого удаленного объекта в контексте потока и удаляет соответствующий узел или узлы из пространства имен фильтра.

На этапе 635 делается определение в отношении того, является ли эта операция операцией, относящейся к удалению. Если это так, процесс ответвляется на этап 640; иначе, процесс ответвляется на этап 645. Относящаяся к удалению операция может содержать операцию создания для удаления при закрытии, операцию задания состояния или операцию очистки, как описано более детально в связи с фиг.11. На этапе 640 пространство имен фильтра обновляется по обстоятельствам, как описано более детально в связи с фиг.10.

На этапе 645 могут выполняться любые другие действия фильтра. Например, фильтр квот может навязывать квоту посредством отказа или разрешения I/O запросу продолжаться. В некоторых вариантах осуществления действия, ассоциированные с этапом 645, происходят сразу после этапа 610. Это может позволить фильтру отказать операции в исполнении и остановить изменение любого объекта. На этапе 620 процесс завершается. Процесс, описанный в связи с фиг.6, может повторяться каждый раз, когда принимается интересующая I/O операция.

Фиг.7 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 630 на фиг.6, которые могут

происходить, когда принимается операция переименования, в соответствии с различными аспектами этого изобретения. На этапе 705 процесс начинается. На этапе 710 делается определение в отношении того, находится ли эта операция переименования в стадии предшествующего обратного вызова или последующего обратного вызова. Если эта операция находится в стадии предшествующего обратного вызова (т.е. до того, как этот I/O запрос был передан файловой системе), процесс ответвляется на этап 715; иначе, так как операция находится в стадии последующего обратного вызова, обработка ответвляется на этап 730. В сущности, перед обновлением структуры данных процесс ожидает, когда операция достигнет успеха.

На этапе 715 контекст обратного вызова выделяется и ассоциируется с операцией. Контекст обратного вызова может включать в себя любые данные и предоставляет средство для передачи относящихся к операции данных стадии последующего обратного вызова. Когда имеет место стадия последующего обратного вызова для конкретной операции, к контексту обратного вызова, ассоциированному с этой операцией, может осуществляться доступ.

На этапе 720 имя объекта до выполнения операции (т.е. старое имя) сохраняется в контексте обратного вызова. На этапе 725 реализуется идущий после операции обратный вызов, что по существу означает, что фильтр показывает, что он желает быть вызванным после того, как операция будет выполнена в файловой системе, и перед тем, как результаты I/O будут возвращены приложению.

На идущей после операции стадии на этапе 730 делается определение в отношении того, потерпела ли операция неудачу. Если это так, процесс ответвляется на этап 750; иначе, процесс ответвляется на этап 735. Если операция потерпела неудачу, нет необходимости обновлять структуру данных.

На этапе 735 старое имя объекта извлекается из контекста обратного вызова. Это старое имя используется для нахождения узла в структуре данных, которая поддерживает пространство имен фильтра. На этапе 740 файловая система запрашивается о новом имени объекта, которое может включать в себя новое имя объекта и/или путь объекта. На этапе 745 структура данных обновляется, как описывается более детально в связи с фиг.8. На этапе 750 контекст обратного вызова освобождается. На этапе 755 процесс завершается.

Фиг.8 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 745 по фиг.7, которые могут происходить при обновлении структуры данных, в соответствии с различными аспектами этого изобретения. На этапе 805 процесс начинается. На этапе 810 делается определение в отношении того, был ли объект перемещен. Если это так, процесс ответвляется на этап 815; иначе, процесс ответвляется на этап 830. Если объект был перемещен, старые узлы могут быть удалены и созданы новые узлы, чтобы обновить пространство имен фильтра и политики, как показано с помощью этапов 815, 820 и 825. Если изменилось только имя объекта, а не его путь, имя объекта, которое указывается в узле, может быть изменено на новое имя объекта, как показано с помощью этапа 830.

На этапе 835 узлы-потомки перемещенного узла могут быть обновлены по обстоятельствам. В одном варианте осуществления полный путь и имя объекта хранятся в каждом узле. Например, ссылаясь на фиг.5, узел QUOTA2 может хранить \DIR1\QUOTA1\QUOTA2. Это может делаться для ускорения выполнения или по другим причинам. Когда имя объекта и/или путь узла изменяются, каждый из

узлов, которые являются потомками этого узла, могут обновляться, чтобы включить в себя новый префиксный путь, который включает в себя новое имя объекта и/или путь, ассоциированный с измененным узлом. В других вариантах осуществления (например, где полный путь и имя объекта не хранятся в узле и не должны обновляться в узлах-потомках) этап 835 может быть пропущен.

На этапе 840 процесс завершается.

Фиг.9 - это диаграмма последовательности операций, в общем представляющая действия, которые соответствуют этапу 815 по фиг.8, которые могут происходить при добавлении узлов, когда объект перемещается, в соответствии с различными аспектами этого изобретения. На этапе 905 процесс начинается. На этапе 910 выбирается первый компонент нового пути объекта. Как используется в пути, компонент указывает на часть пути, которая соответствует объекту-предку объекта, и может включать в себя корневую директорию. Например, компоненты для \DIR1\QUOTA1\QUOTA2 - это корень (т.е. \ ), DIR1, QUOTA и QUOTA2. Первый компонент - это корень.

На этапе 915 делается определение в отношении того, существует ли узел, который соответствует выбранному компоненту. Если это так, процесс ответвляется на этап 925, так как новый узел не должен создаваться; иначе, процесс ответвляется на этап 920. На этапе 920 создается новый узел, который соответствует этому компоненту.

На этапе 925 делается определение в отношении того, является ли в текущее время выбранный компонент последним компонентом нового пути. Если это так, процесс ответвляется на этап 905, где процесс возвращается. Иначе, процесс ответвляется на этап 930, где выбирается следующий компонент, и могут повторяться действия, ассоциированные с этапами 915, 925, до тех пор, пока не будет обработан последний компонент. Следует понимать, что процесс, описанный в связи с фиг.9, в сущности, модифицирует структуру данных, чтобы включить узлы для всех компонентов в новом пути, которые еще не существуют в структуре данных.

Фиг.10 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 825 по фиг.8, которые могут происходить при обновлении структуры данных, чтобы удалить префиксные узлы, ассоциированные со старым именем объекта, в соответствии с различными аспектами этого изобретения. В общем, когда узел перемещается в новое местоположение в структуре данных, все узлы, которые больше не нужны, удаляются из структуры данных. Узел является более не нужным, если он не имеет потомка, который имеет некоторую ассоциированную с ним политику, и сам узел не имеет ассоциированной с ним политики.

Процесс начинается на этапе 1005. На этапе 1010 выбирается узел, ассоциированный со старым объектом. На этапе 1015 удаляются политики старого узла, так как они уже были ранее скопированы в новый узел. На этапе 1020 делается определение в отношении того, имеет ли в текущее время выбранный узел политику, ассоциированную с ним. Если это так, процесс ответвляется на этап 1045; иначе, процесс ответвляется на этап 1025.

На этапе 1025 делается определение в отношении того, имеет ли узел каких-либо потомков. Если это так, процесс ответвляется на этап 1045; иначе, процесс ответвляется на этап 1030. Так как узлы поддерживаются для объектов, которые имеют политики и их предков, и не поддерживаются в противном случае, если узел имеет каких-либо потомков, это показывает, что, по меньшей мере, один из потомков

узла ассоциирован с некоторой политикой. Например, если узел, который не имел политики, имел двух потомков и один из этих потомков был переименован, так что он более не является потомком этого узла, узел не будет удаляться, пока второй потомок не будет также удален.

На этапе 1030 узел удаляется, так как он больше не нужен. На этапе 1035 делается определение в отношении того, имеет ли узел родителя. Если это так, процесс ответвляется на этап 1040, где выбирается родитель. От этапа 1040 обработка переходит на этап 1020, и могут повторяться действия, ассоциированные с этапами 1020-1035. Следует понимать, что в действительности эти действия рекурсивно поднимаются по структуре данных, удаляя узлы до тех пор, пока они не найдут узел, который либо сам ассоциирован с некоторой политикой, либо который имеет потомка, ассоциированного с некоторой политикой.

На этапе 1045 процесс завершается.

Следует понимать, что диаграммы последовательности операций по фиг.6-10 представляют один способ, в котором может обновляться структура данных пространства имен, чтобы быть совместимой, когда происходят переименования в файловой системе, и что содержание структуры данных пространства имен, совместимой с пространством имен файловой системы, в течение переименований может быть реализовано многообразием способов без отхода от сущности или объема этого изобретения.

Фиг.11 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 640 по фиг.6, которые могут происходить, чтобы поддерживать согласованность между пространством имен фильтра и пространством имен файловой системы в течение относящихся к удалению операций, в соответствии с различными аспектами этого изобретения. На этапе 1105 процесс начинается.

На этапе 1110 делается определение в отношении того, является ли операция операцией создания для удаления при закрытии. Если это так, процесс ответвляется на этап 1115; иначе, процесс ответвляется на этап 1130. Операция создания может создавать объект или открывать уже существующий объект. Объект может создаваться для удаления при закрытии. Когда объект создается для удаления при закрытии, это означает, что объект должен удаляться после того, как этот объект будет закрыт. То есть объект может создаваться, существовать и продолжать существовать до тех пор, пока все процессы, которые имеют описатели для этого объекта, не закроют свои описатели. После того как все описатели будут закрыты, объект удаляется. На этапе 1115 выполняются действия для создания, как описывается более детально в связи с фиг.12.

На этапе 1130 делается определение в отношении того, является ли операция операцией задания состояния. Если это так, процесс ответвляется на этап 1135; иначе, процесс ответвляется на этап 1140. В одном варианте осуществления объект может иметь флаг, ассоциированный с ним, который показывает, должен ли быть объект удален после закрытия всех описателей для этого объекта. Этот флаг может устанавливаться (т.е. показывать, что объект должен удаляться) или сбрасываться (т.е. показывать, что объект не должен удаляться). Более того, этот флаг может устанавливаться или сбрасываться каждым процессом, который имеет какой-либо открытый описатель для объекта. Таким образом, один процесс может установить этот флаг, только чтобы этот флаг сбрасывался другим процессом. Объект остается или удаляется в зависимости от состояния (например, установлен или сброшен) флага

во время, когда последний процесс с описателем для этого объекта закрывает упомянутый описатель. На этапе 1135 действия, относящиеся к заданию состояния объекта, выполняются, как описано более детально в связи с фиг.14.

На этапе 1140 делается определение в отношении того, является ли операция операцией очистки. Если это так, процесс ответвляется на этап 1145; иначе, процесс ответвляется на этап 1155. Операция очистки может выдаваться каждый раз, когда процесс закрывает описатель для объекта. Как упоминалось ранее, однако, объект может в действительности не удаляться до тех пор, пока последний процесс с некоторым описателем для этого объекта не закроет упомянутый описатель. Даже после того как последний процесс закроет объект, объект может не быть удален, если, например, во время последнего закрытия флаг для удаления объекта показывает, что объект не должен удаляться при закрытии. На этапе 1145 выполняются действия очистки, как описывается более детально в связи с фиг.15.

На этапе 1155 процесс завершается. Процесс, описанный в связи с фиг.11, может повторяться каждый раз, когда фильтр принимает относящуюся к удалению I/O операцию для интересующего объекта.

Фиг.12 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1115 по фиг.11, которые могут происходить, когда принимается операция создания для удаления при закрытии, в соответствии с различными аспектами этого изобретения. На этапе 1205 процесс начинается.

На этапе 1210 делается определение в отношении того, находится ли операция в идущей перед операцией стадии. Если это так, процесс ответвляется на этап 1215; иначе, процесс ответвляется на этап 1220.

На этапе 1215 реализуется идущий после операции обратный вызов, так что фильтр примет обратный вызов в течение идущей после операции стадии операции. На этапе 1220 делается определение в отношении того, потерпела ли операция неудачу. Если это так, процесс ответвляется на этап 1240; иначе, процесс ответвляется на этап 1225.

На этапе 1225 контекст потока выделяется (если он еще не существует) и ассоциируется с объектом. На этапе 1230 в контексте потока сохраняется имя объекта. На этапе 1235 в контексте потока обновляется флаг удаления. Для объектов, созданных для удаления при закрытии, этот флаг может показывать, что объект должен быть удален, когда все описатели для этого объекта закрыты. Этот тип флага может не модифицироваться последующими процессами. Когда все описатели для объекта закрываются, объект, который был открыт для удаления при закрытии, удаляется.

Для объектов, которые не были открыты для удаления при закрытии, флаг может устанавливаться или сбрасываться (посредством операции задания состояния), чтобы показывать, что объект должен или не должен удаляться при закрытии. Этот флаг затем может быть изменен последующими операциями задания состояния. Удаляется ли объект после того, как все описатели будут закрыты, зависит от состояния флага, когда закрывается последний описатель для объекта.

На этапе 1240 процесс завершается.

Фиг.13 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 632 по фиг.6, которые могут происходить, когда принимается операция переименования, в соответствии с различными аспектами этого изобретения. На этапе 1305 процесс начинается.

На этапе 1310 делается определение в отношении того, находится ли операция в идущей перед операцией стадии. Если это так, процесс ответвляется на этап 1330; иначе, процесс ответвляется на этап 1315.

На этапе 1330 реализуется идущий после операции обратный вызов, так что фильтр примет обратный вызов в течение идущей после операции стадии операции. На этапе 1315 делается определение в отношении того, потерпела ли операция неудачу. Если это так, процесс ответвляется на этап 1335; иначе, процесс ответвляется на этап 1320.

На этапе 1320 делается определение в отношении того, существует ли контекст потока, который ассоциирован с объектом. Если это так, процесс ответвляется на этап 1325; иначе, процесс ответвляется на этап 1335. Контекст потока может не существовать, если, например, объект не был открыт для удаления при закрытии и никаких состояний не было задано или сброшено для флага, который показывает, должен ли объект удаляться при закрытии.

На этапе 1325 в контексте потока обновляется имя, чтобы отражать то имя, к которому объект был изменен.

Это полезно, например, позже при определении того, будет ли это изменение имени затрагивать какой-либо узел.

На этапе 1335 процесс завершается.

Фиг.14 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1135 по фиг.11, которые могут происходить, когда принимается операция задания состояния, в соответствии с различными аспектами этого изобретения. На этапе 1405 процесс начинается.

На этапе 1410 делается определение в отношении того, находится ли операция в идущей перед операцией стадии. Если это так, процесс ответвляется на этап 1415; иначе, процесс ответвляется на этап 1420.

На этапе 1415 реализуется идущий после операции обратный вызов, так что фильтр примет обратный вызов в течение идущей после операции стадии операции. На этапе 1420 делается определение в отношении того, потерпела ли операция неудачу. Если это так, процесс ответвляется на этап 1440; иначе, процесс ответвляется на этап 1425.

На этапе 1425 делается определение в отношении, существует ли контекст потока, который ассоциирован с объектом. Если это так, процесс ответвляется на этап 1435; иначе, процесс ответвляется на этап 1430, где создается контекст потока, и там сохраняется имя объекта. Контекст потока ассоциируется с объектом.

На этапе 1435 обновляется флаг удаления, чтобы отражать состояние удаления объекта. Как упоминалось прежде, какой-либо процесс может устанавливать этот флаг, чтобы показывать, что объект должен быть удален при закрытии, или сбрасывать этот флаг, чтобы показывать, что объект не должен быть удален при закрытии. На этапе 1440 процесс завершается.

Фиг.15 - это диаграмма последовательности операций, которая в общем представляет действия, которые соответствуют этапу 1145 по фиг.11, которые могут происходить, когда принимается операция очистки, в соответствии с различными аспектами этого изобретения. На этапе 1505 процесс начинается.

На этапе 1510 делается определение в отношении того, находится ли операция в идущей перед операцией стадии. Если это так, процесс ответвляется на этап 1515; иначе, процесс ответвляется на этап 1520.

На этапе 1515 реализуется идущий после операции обратный вызов, так что фильтр

примет обратный вызов в течение идущей после операции стадии операции. На этапе 1520 делается определение в отношении того, потерпела ли операция неудачу. Если это так, процесс ответвляется на этап 1550; иначе, процесс ответвляется на этап 1525.

На этапе 1525 делается определение в отношении того, существует ли контекст потока, который ассоциирован с объектом. Если это так, процесс ответвляется на этап 1530; иначе, процесс ответвляется на этап 1550. На этапе 1530 делается определение в отношении того, является ли объект интересующим объектом для фильтра. Если это так, процесс ответвляется на этап 1535; иначе, процесс ответвляется на этап 1550.

На этапе 1535 делается определение в отношении того, установлен ли флаг удаления на удаление или был ли объект открыт для удаления при закрытии. Если удовлетворяется какое-либо из этих условий, процесс ответвляется на этап 1540; иначе, процесс ответвляется на этап 1550. В одном варианте осуществления, когда объект открыт для удаления при закрытии, позже можно не указывать, что этот объект не будет удаляться при закрытии.

На этапе 1540 делается определение в отношении того, удален ли объект. Так как много процессов могут открывать объект и объект удаляется только после того, как все процессы закроют описатели для этого объекта, некоторый объект может не удаляться, даже хотя все другие критерии были удовлетворены. В одном варианте осуществления этого изобретения операционная система может информировать фильтр, когда объект был удален (вместо того, чтобы фильтр сам проверял это). В таком варианте осуществления способ, описанный в связи с фиг.15, может содержать определение того, является ли объект интересующим объектом, и удаление узлов по обстоятельствам.

На этапе 1545 узел и любые соответствующие узлы удаляются по обстоятельствам. Если узел является предком других узлов в структуре данных, описанной в связи с фиг.5, эти другие узлы также удаляются. Более того, могут выполняться действия, аналогичные действиям, описанным в связи с фиг. 10, чтобы удалять узлы, которые являются узлами-предками этого узла и которые больше не нужны. На этапе 1550 процесс завершается.

Следует понимать, что действия, описанные в связи с фиг. 11-15, могут заменяться другими действиями и могут выполняться в другом порядке в зависимости от способа работы операционной системы без отхода от сущности или объема этого изобретения. Например, некоторые операционные системы могут не иметь флага, чтобы показывать, должен ли файл быть удален после того, как все описатели для этого файла будут закрыты. Скорее, такие операционные системы могут предоставлять отдельную операцию удаления для удаления объекта. В таких операционных системах способ для обнаружения изменений в отношении удалений может содержать отслеживание того, когда какой-либо объект будет удален, и удаление узлов (если есть какие-либо), которые ассоциированы с этим объектом.

Реализация идущих после операции обратных вызовов по фиг.12-14 может быть пропущена в вариантах осуществления, в которых идущий после операции обратный вызов происходит по умолчанию.

Как можно видеть из предшествующего подробного описания, предоставляется способ и система для поддержания согласованности пространства имен с файловой системой. Хотя это изобретение является восприимчивым к различным модификациям и альтернативным конструкциям, определенные проиллюстрированные его варианты

осуществления показаны на чертежах и были подробно описаны выше. Следует понимать, однако, что нет намерения ограничивать это изобретение конкретными раскрытыми формами, но наоборот, намерение состоит в том, чтобы охватить все модификации, альтернативные конструкции и эквиваленты, попадающие в рамки сущности и объема этого изобретения.

## Формула изобретения

1. Считываемый компьютером носитель, имеющий исполняемые компьютером инструкции для выполнения фильтром отслеживания запросов ввода/вывода, ассоциированных с этим фильтром, исполнением которых реализуется способ, содержащий этапы, на которых:

создают экземпляр фильтра файловой системы, ассоциированного с файловой системой, причем данный фильтр выполнен с возможностью обеспечивать функциональные возможности, не предусмотренные файловой системой;

регистрируют фильтр с помощью механизма регистрации, реализуемого в средстве управления фильтрами;

уведомляют посредством фильтра средство управления фильтрами о по меньшей мере одном типе запроса ввода/вывода, в котором он заинтересован;

задают посредством фильтра, должен ли фильтр быть уведомлен о предварительных обратных вызовах для упомянутого по меньшей мере одного типа запроса ввода/вывода;

задают посредством фильтра, должен ли фильтр быть уведомлен о последующих обратных вызовах для упомянутого по меньшей мере одного типа запроса ввода/вывода;

поддерживают данные, идентифицирующие один или более объектов, представляющих интерес для фильтра;

принимают запрос ввода/вывода в отношении объекта файловой системы;

создают посредством средства управления фильтрами структуру данных, содержащую данные обратного вызова из принятого запроса ввода/вывода;

определяют, что объект является объектом, представляющим интерес для фильтра;

вызывают фильтр посредством средства управления фильтрами;

пересылают посредством средства управления фильтрами эти данные обратного вызова фильтру;

посредством фильтра определяют действие фильтра, ассоциированное с принятым запросом ввода/вывода, и выполняют это определенное действие фильтра; и

обновляют пространство имен, ассоциированное с фильтром, на основе принятого запроса ввода/вывода.

2. Считываемый компьютером носитель по п.1, в котором запрос ввода/вывода содержит операцию переименования.

3. Считываемый компьютером носитель по п.2, в котором операция переименования обуславливает перемещение объекта в другое местоположение в файловой системе.

4. Считываемый компьютером носитель по п.2, в котором операция переименования обуславливает изменение имени, ассоциированного с объектом.

5. Считываемый компьютером носитель по п.1, в котором запрос ввода/вывода содержит операцию удаления.

6. Считываемый компьютером носитель по п.5, в котором операция удаления содержит установку флага для указания того, что объект должен быть удален после

того, как все описатели, ассоциированные с этим объектом, будут закрыты.

7. Считываемый компьютером носитель по п.6, дополнительно содержащий сбрасывание флага для указания того, что объект не должен быть удален.

8. Считываемый компьютером носитель по п.1, в котором запрос ввода/вывода содержит операцию очистки.

9. Считываемый компьютером носитель по п.8, в котором операция очистки выдается каждый раз, когда процесс закрывает описатель для объекта.

10. Считываемый компьютером носитель по п.1, в котором при определении того, что объект является объектом, представляющим интерес для фильтра, выполняют поиск имени этого объекта в структуре данных, поддерживаемой фильтром.

11. Считываемый компьютером носитель по п.10, в котором структура данных скомпонована иерархически.

12. Считываемый компьютером носитель по п.10, в котором структура данных включает в себя узел для каждого объекта, который имеет ассоциированную с ним политику.

13. Считываемый компьютером носитель по п.12, в котором структура данных дополнительно включает в себя узел для каждого предка каждого объекта, который имеет ассоциированную с ним политику.

14. Считываемый компьютером носитель по п.13, в котором предок содержит директорию, в которой объект находится.

15. Считываемый компьютером носитель по п.1, в котором объект содержит директорию файловой системы.

16. Считываемый компьютером носитель по п.1, в котором объект содержит файл файловой системы.

17. Считываемый компьютером носитель по п.1, в котором запрос ввода/вывода содержит изменение имени объекта.

18. Считываемый компьютером носитель по п.1, в котором запрос ввода/вывода содержит изменение местоположения объекта в файловой системе.

19. Способ выполнения фильтром отслеживания запросов ввода/вывода, ассоциированных с этим фильтром, содержащий этапы, на которых:

создают экземпляр фильтра файловой системы, ассоциированного с файловой системой, причем данный фильтр выполнен с возможностью обеспечивать функциональные возможности, не предусмотренные файловой системой;

регистрируют фильтр с помощью механизма регистрации, реализуемого в средстве управления фильтрами;

уведомляют посредством фильтра средство управления фильтрами о по меньшей мере одном типе запроса ввода/вывода, в котором он заинтересован;

задают посредством фильтра, должен ли фильтр быть уведомлен о предварительных обратных вызовах для упомянутого по меньшей мере одного типа запроса ввода/вывода;

задают посредством фильтра, должен ли фильтр быть уведомлен о последующих обратных вызовах для упомянутого по меньшей мере одного типа запроса ввода/вывода;

поддерживают данные, идентифицирующие один или более объектов, представляющих интерес для фильтра;

поддерживают пространство имен, ассоциированное с фильтром, которое отлично от пространства имен, поддерживаемого файловой системой;

принимают запрос ввода/вывода в отношении объекта файловой системы;

создают посредством средства управления фильтрами структуру данных, содержащую данные обратного вызова из принятого запроса ввода/вывода;

определяют, что объект является объектом, представляющим интерес для фильтра;

вызывают фильтр посредством средства управления фильтрами;

пересылают посредством средства управления фильтрами эти данные обратного вызова фильтру;

посредством фильтра определяют действие фильтра, ассоциированное с принятым запросом ввода/вывода, и выполняют это определенное действие фильтра; и

обновляют пространство имен, ассоциированное с фильтром, на основе принятого запроса ввода/вывода.

20. Способ по п.19, дополнительно содержащий этапы, на которых:

выделяют посредством фильтра контекст обратного вызова и реализуют посредством фильтра идущий после операции обратный вызов.

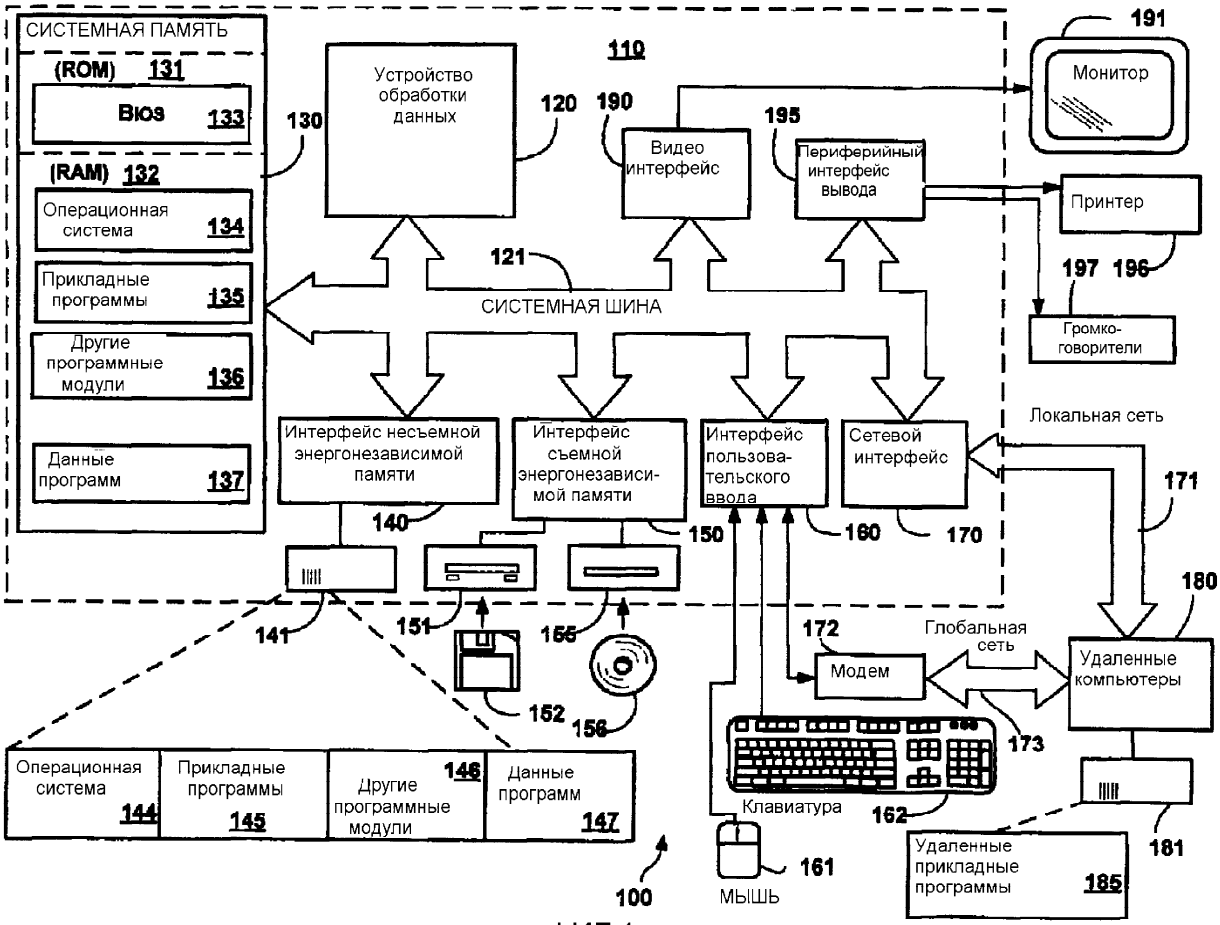
21. Способ по п.19, в котором при обновлении пространства имен:

добавляют новый узел нового объекта в структуру данных; перемещают политику, ассоциированную с прежним узлом, к этому новому узлу и удаляют упомянутый прежний узел из структуры данных.

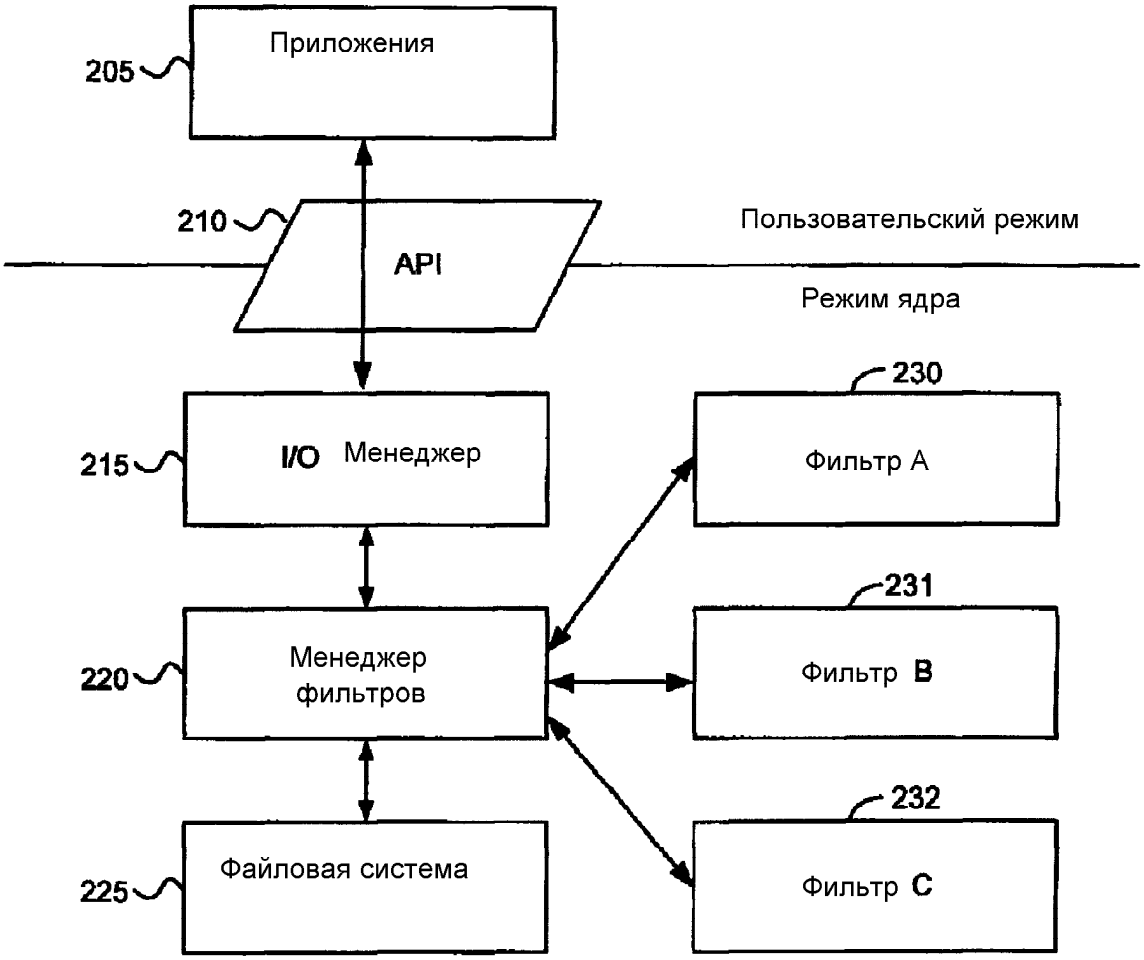
22. Способ по п.19, дополнительно содержащий этапы, на которых:

регистрируют множество фильтров с помощью упомянутого механизма регистрации и

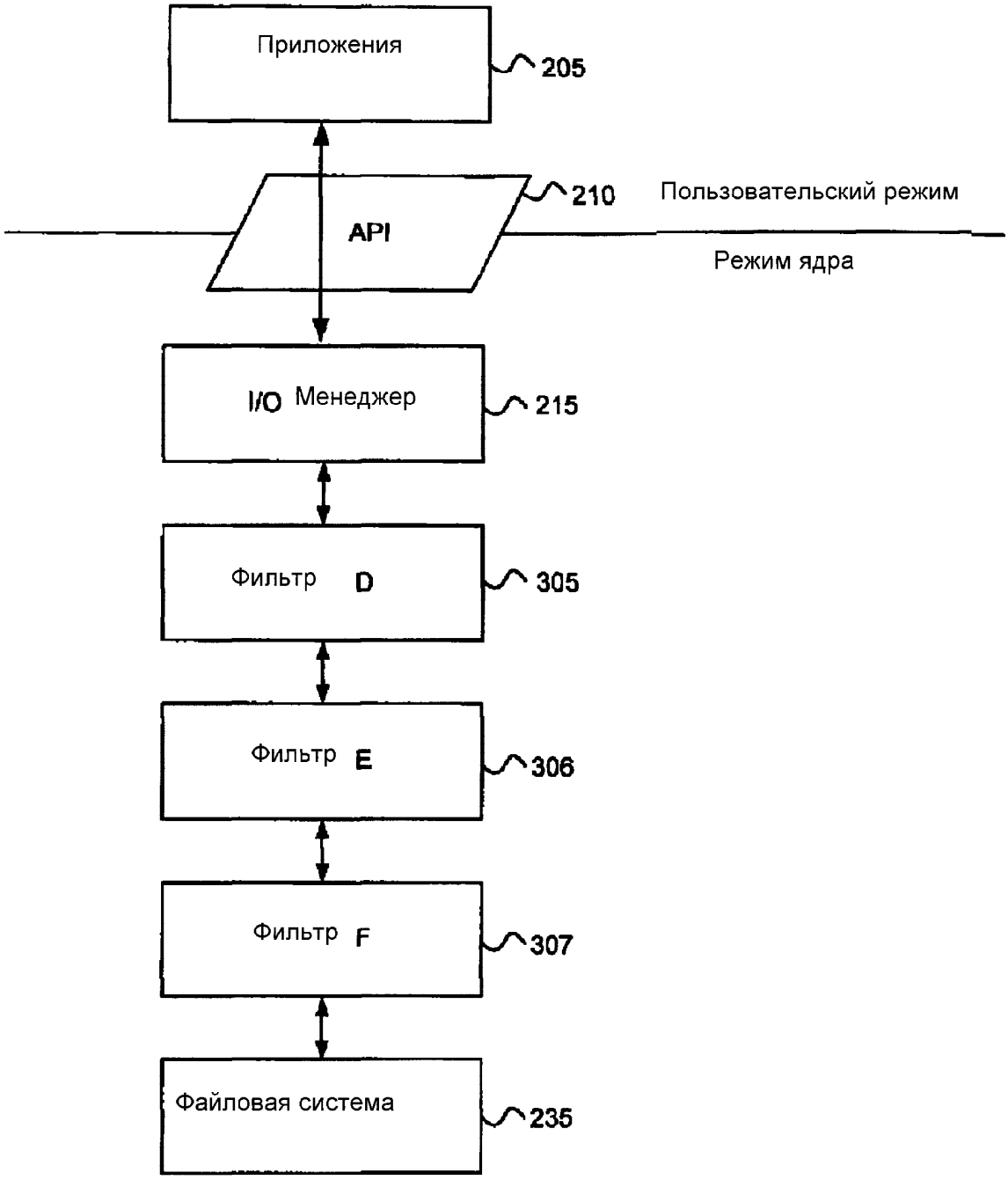
пересылают посредством по меньшей мере одного из этого множества фильтров данные обратного вызова по меньшей мере одному другому из упомянутого множества фильтров.


СИСТЕМНАЯ ПАМЯТЬ
(ROM) 131
BIOS 133
(RAM) 132
Операционная система 134
Прикладные программы 135
Другие программные модули 136
Данные программ 137
130
Устройство обработки данных
120
110
190
Видео интерфейс
195
Периферийный интерфейс вывода
191
Монитор
Принтер
197
196
Громко-говорители
121
СИСТЕМНАЯ ШИНА
Интерфейс несъемной энергонезависимой памяти
Интерфейс съемной энергонезависи-мой памяти
Интерфейс пользова-тельского ввода
Сетевой интерфейс
Локальная сеть
171
140
150
160
170
180
141
151
155
172
Модем
Глобальная сеть
Удаленные компьютеры
152
156
173
Операционная система 144
Прикладные программы 145
Другие программные модули 146
Данные программ 147
Клавиатура
162
Удаленные прикладные программы 185
181
100
161
МЫШЬ

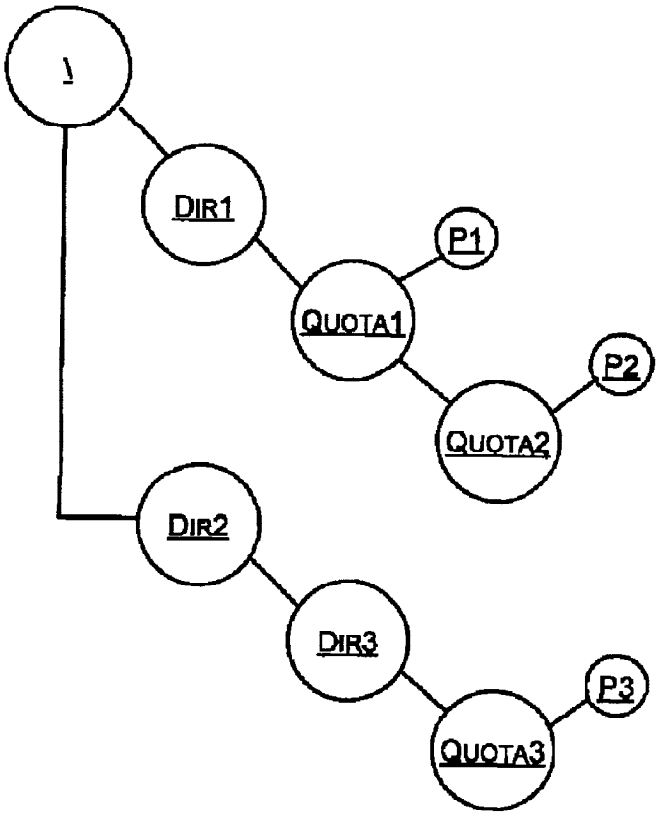
ФИГ.1


Приложения
205
210
API
Пользовательский режим
Режим ядра
230
I/O Менеджер
215
Фильтр A
231
Менеджер фильтров
220
Фильтр B
232
Файловая система
225
Фильтр C

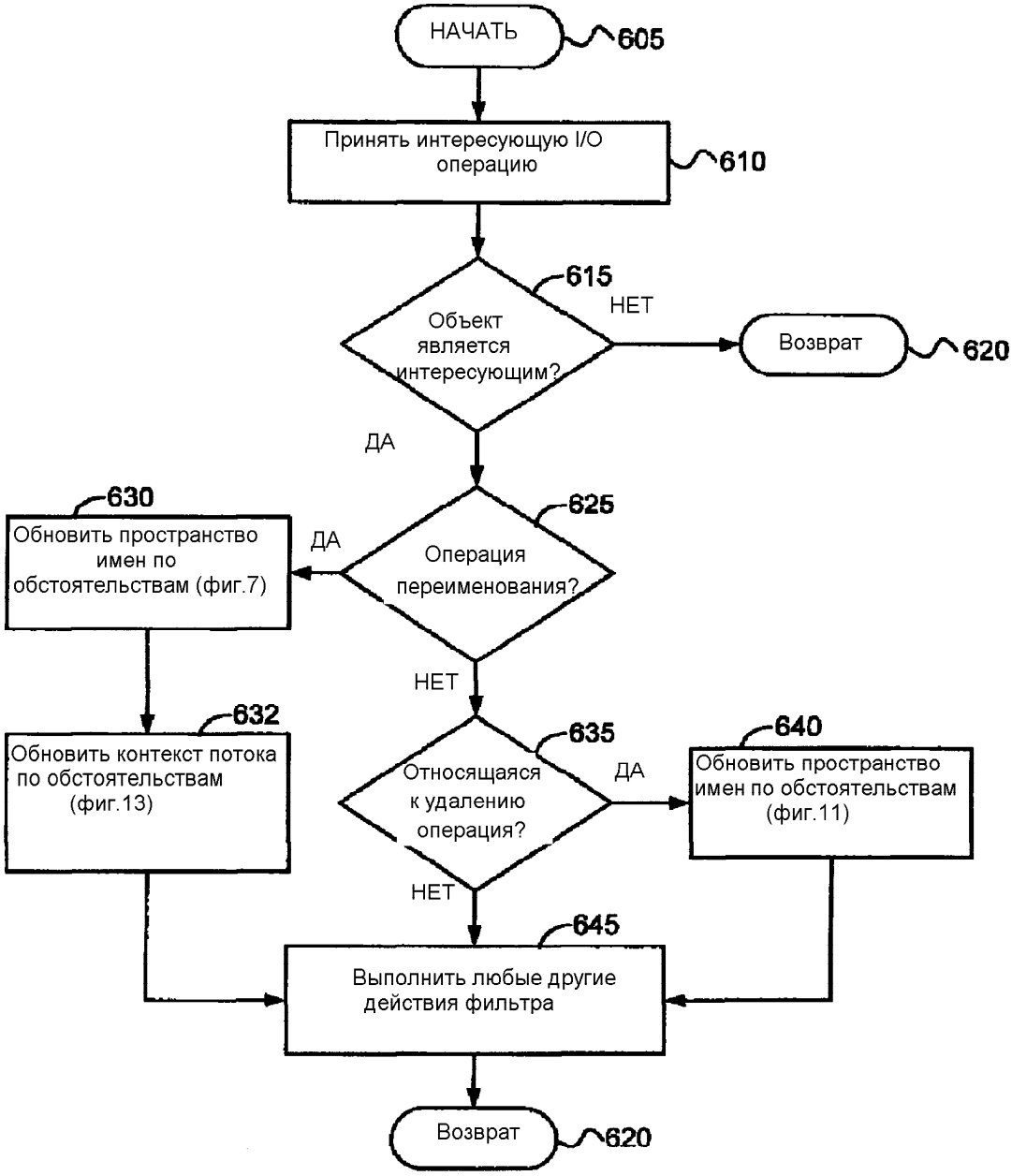
ФИГ.2


Приложения
205
210
Пользовательский режим
API
Режим ядра
I/O Менеджер
215
Фильтр D
305
Фильтр E
306
Фильтр F
307
Файловая система
235

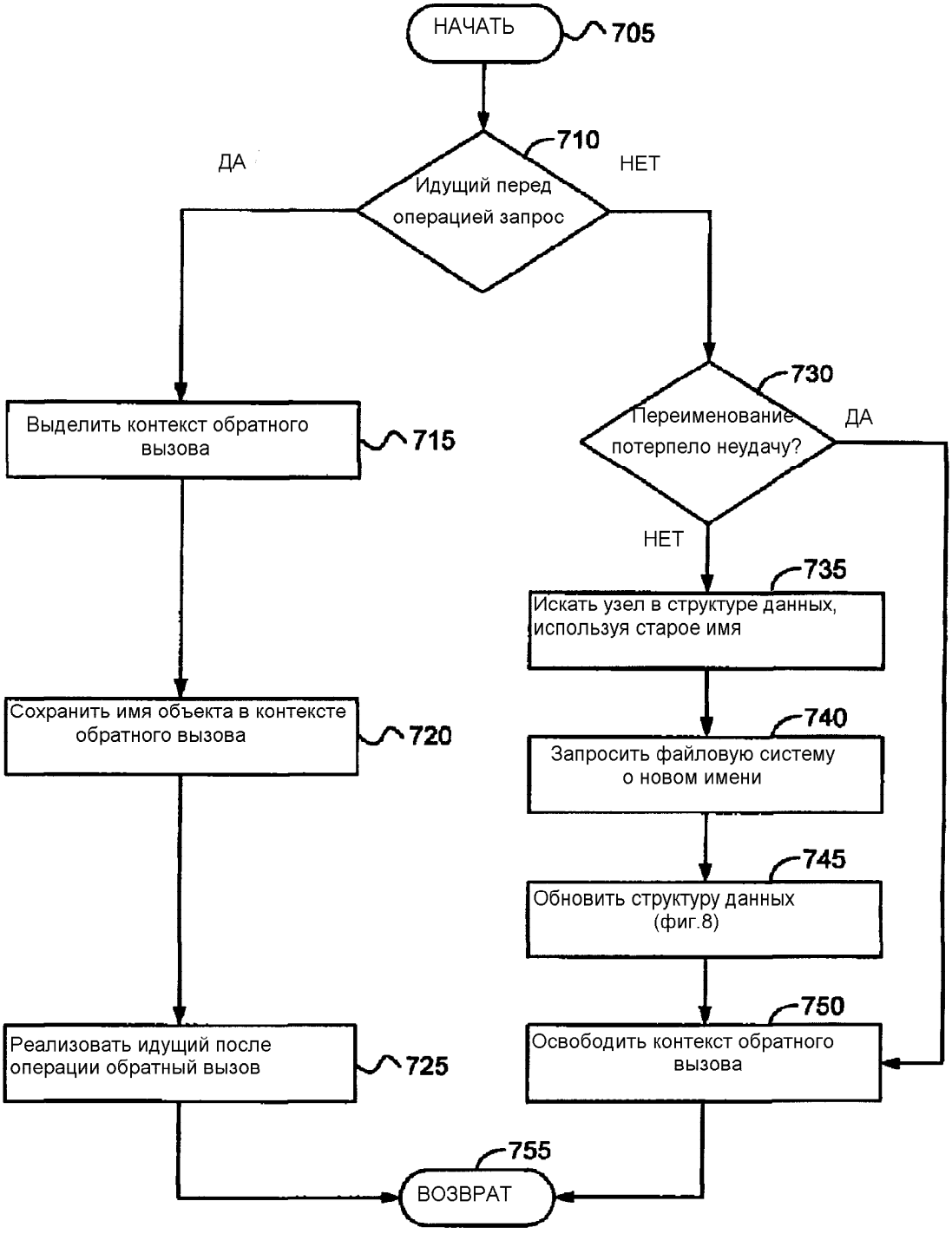
ФИГ.3



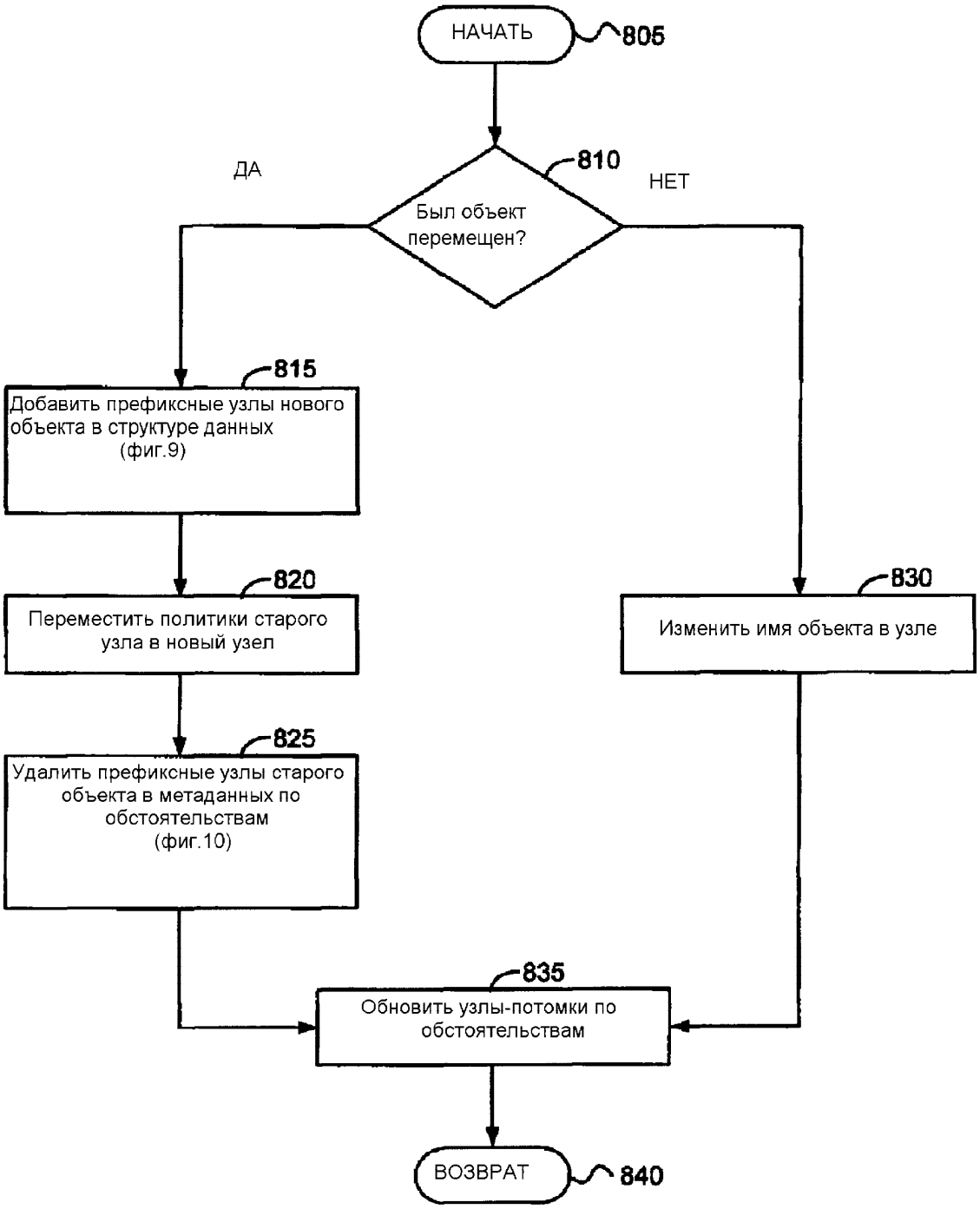

ФИГ.5


НАЧАТЬ
605
Принять интересующую I/O операцию
610
615
Объект является интересующим?
НЕТ
Возврат
620
ДА
630
Обновить пространство имен по обстоятельствам (фиг.7)
ДА
625
Операция переименования?
НЕТ
632
Обновить контекст потока по обстоятельствам (фиг.13)
635
Относящаяся к удалению операция?
ДА
640
Обновить пространство имен по обстоятельствам (фиг.11)
НЕТ
645
Выполнить любые другие действия фильтра
Возврат
620

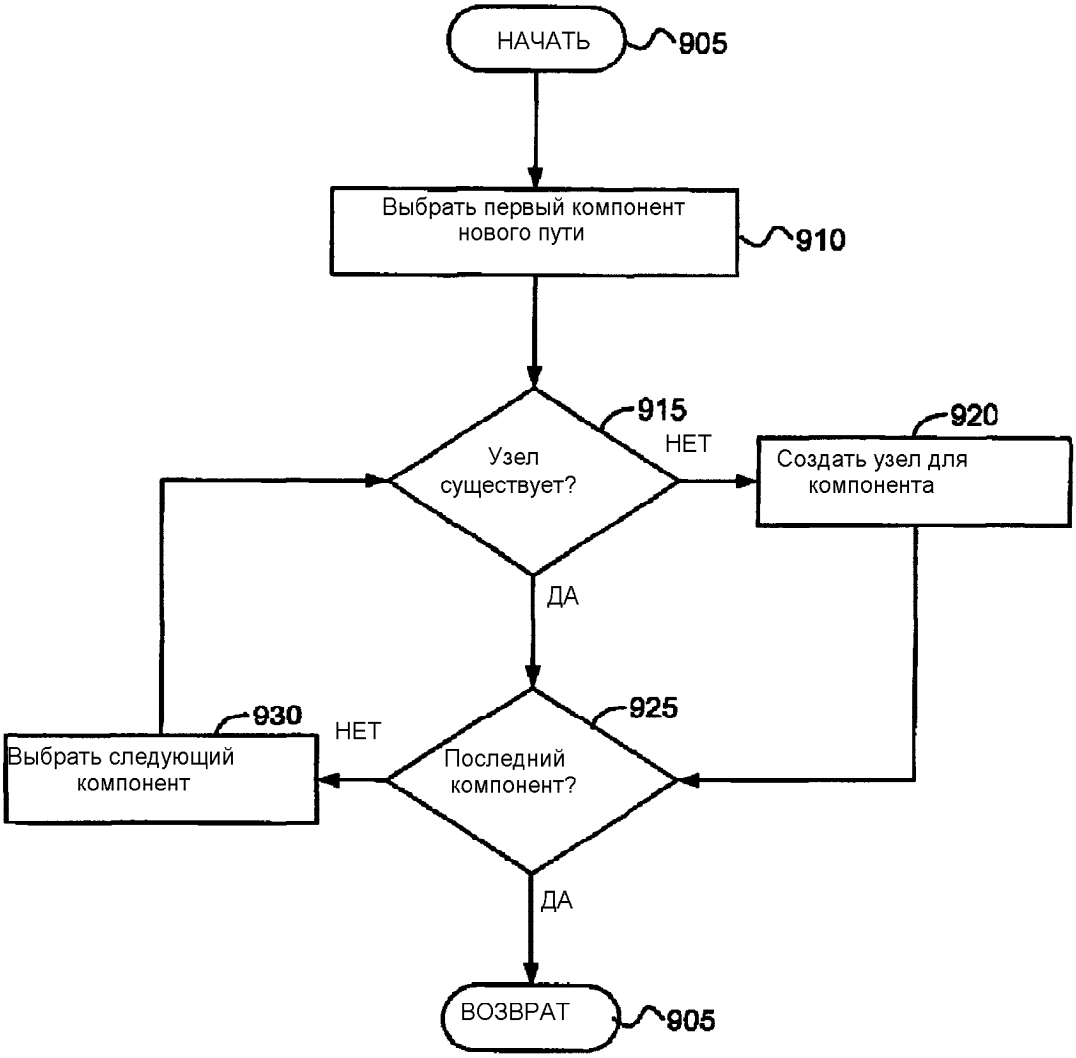
ФИГ.6


НАЧАТЬ
705
710
Идущий перед операцией запрос
ДА
НЕТ
Выделить контекст обратного вызова
715
Сохранить имя объекта в контексте обратного вызова
720
Реализовать идущий после операции обратный вызов
725
730
Переименование потерпело неудачу?
ДА
НЕТ
735
Искать узел в структуре данных, используя старое имя
740
Запросить файловую систему о новом имени
745
Обновить структуру данных (фиг.8)
750
Освободить контекст обратного вызова
755
ВОЗВРАТ

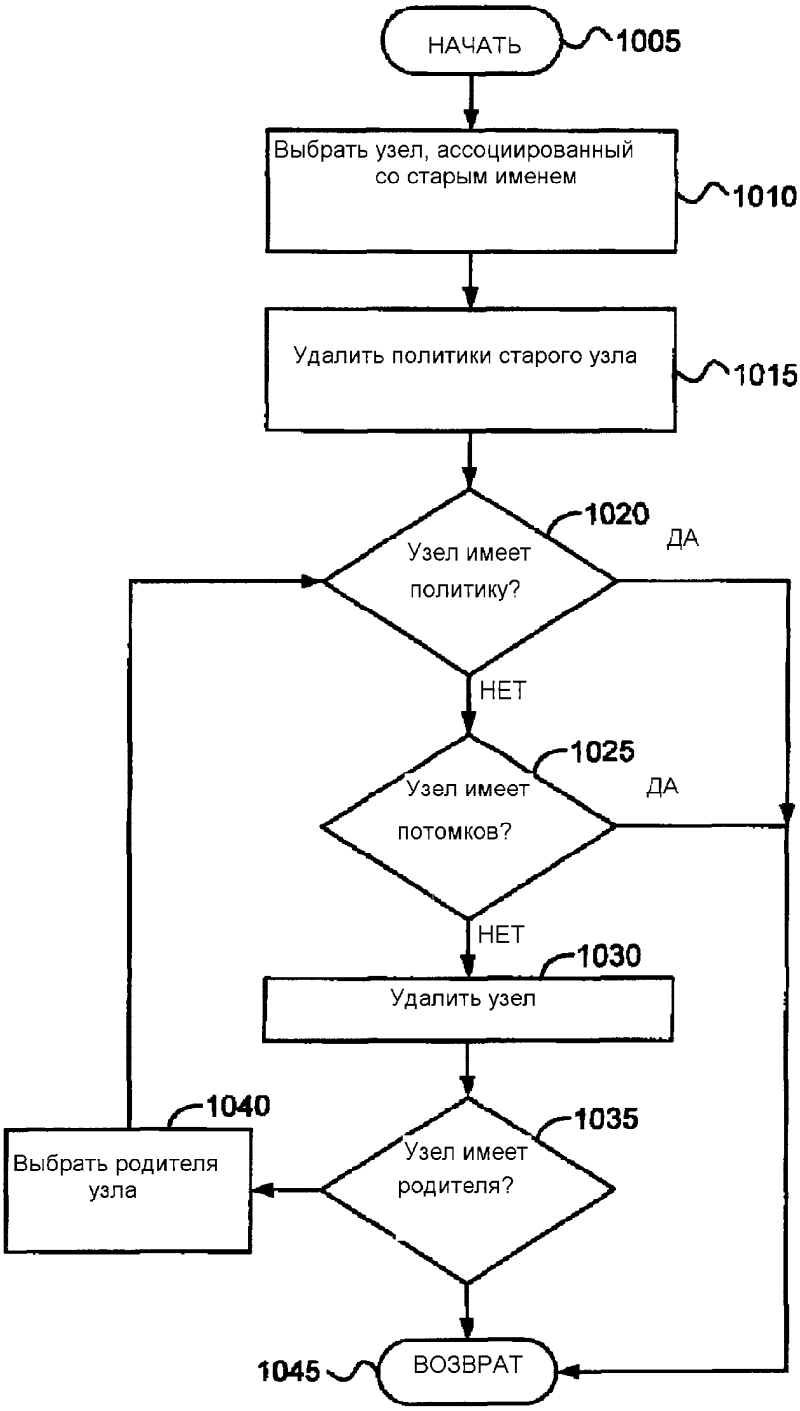
ФИГ.7


НАЧАТЬ
805
Был объект перемещен?
810
ДА
НЕТ
Добавить префиксные узлы нового объекта в структуре данных (фиг.9)
815
Переместить политики старого узла в новый узел
820
Удалить префиксные узлы старого объекта в метаданных по обстоятельствам (фиг.10)
825
Изменить имя объекта в узле
830
Обновить узлы-потомки по обстоятельствам
835
ВОЗВРАТ
840

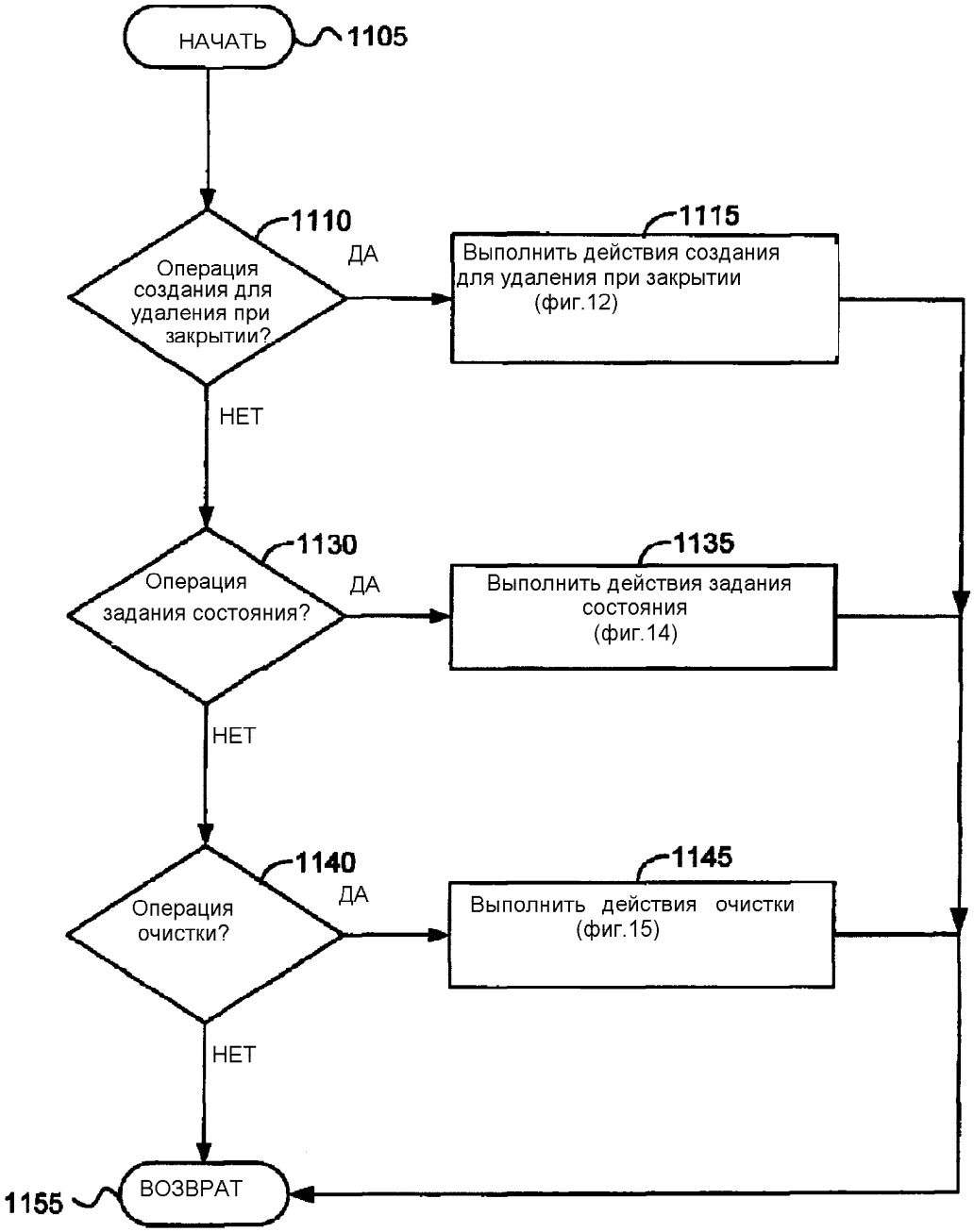
ФИГ.8


НАЧАТЬ
905
Выбрать первый компонент нового пути
910
915
Узел существует?
НЕТ
920
Создать узел для компонента
ДА
930
НЕТ
925
Выбрать следующий компонент
Последний компонент?
ДА
ВОЗВРАТ
905

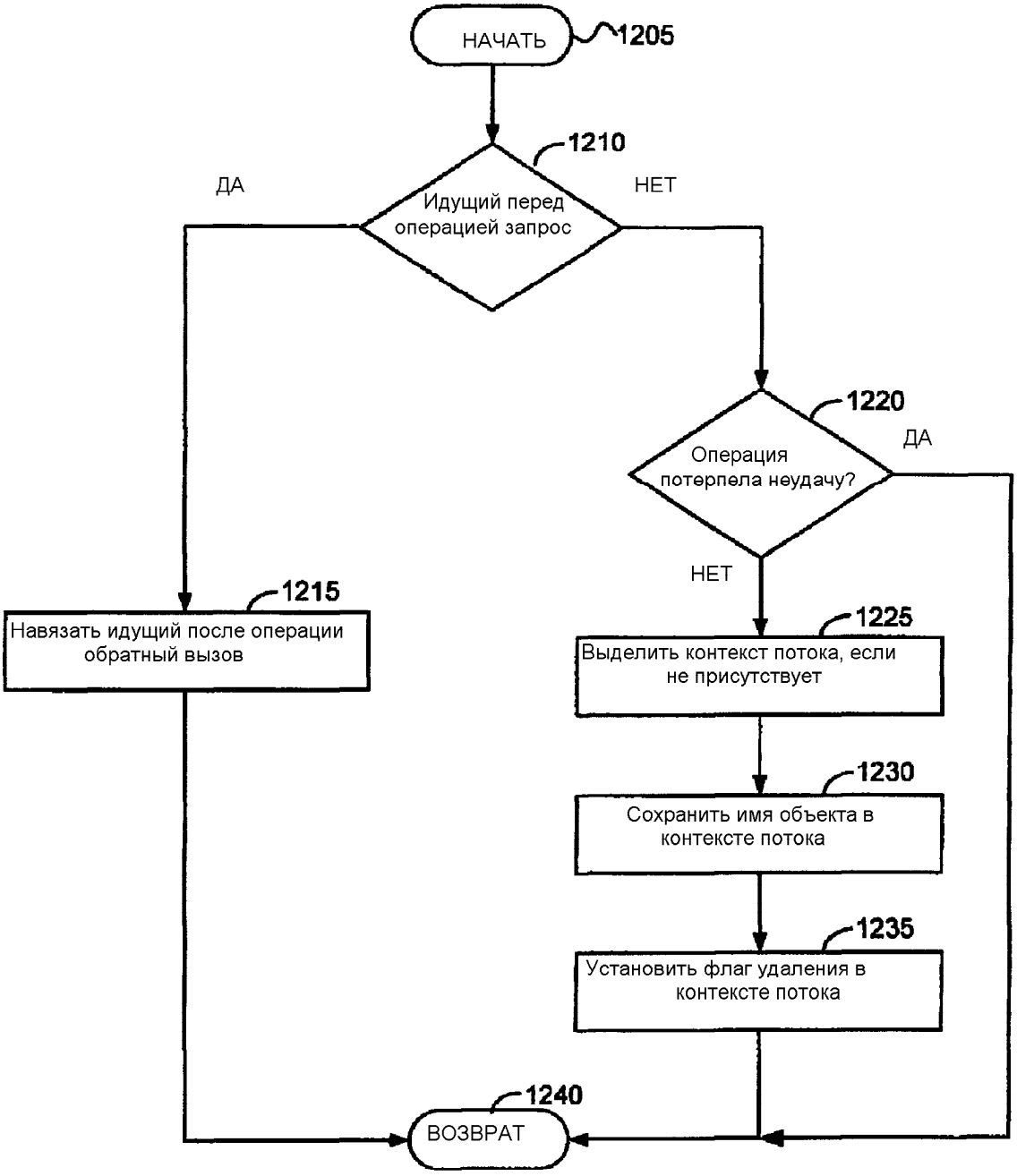
ФИГ.9


НАЧАТЬ
1005
Выбрать узел, ассоциированный со старым именем
1010
Удалить политики старого узла
1015
1020
Узел имеет политику?
ДА
НЕТ
1025
Узел имеет потомков?
ДА
НЕТ
1030
Удалить узел
1035
Узел имеет родителя?
1040
Выбрать родителя узла
1045
ВОЗВРАТ

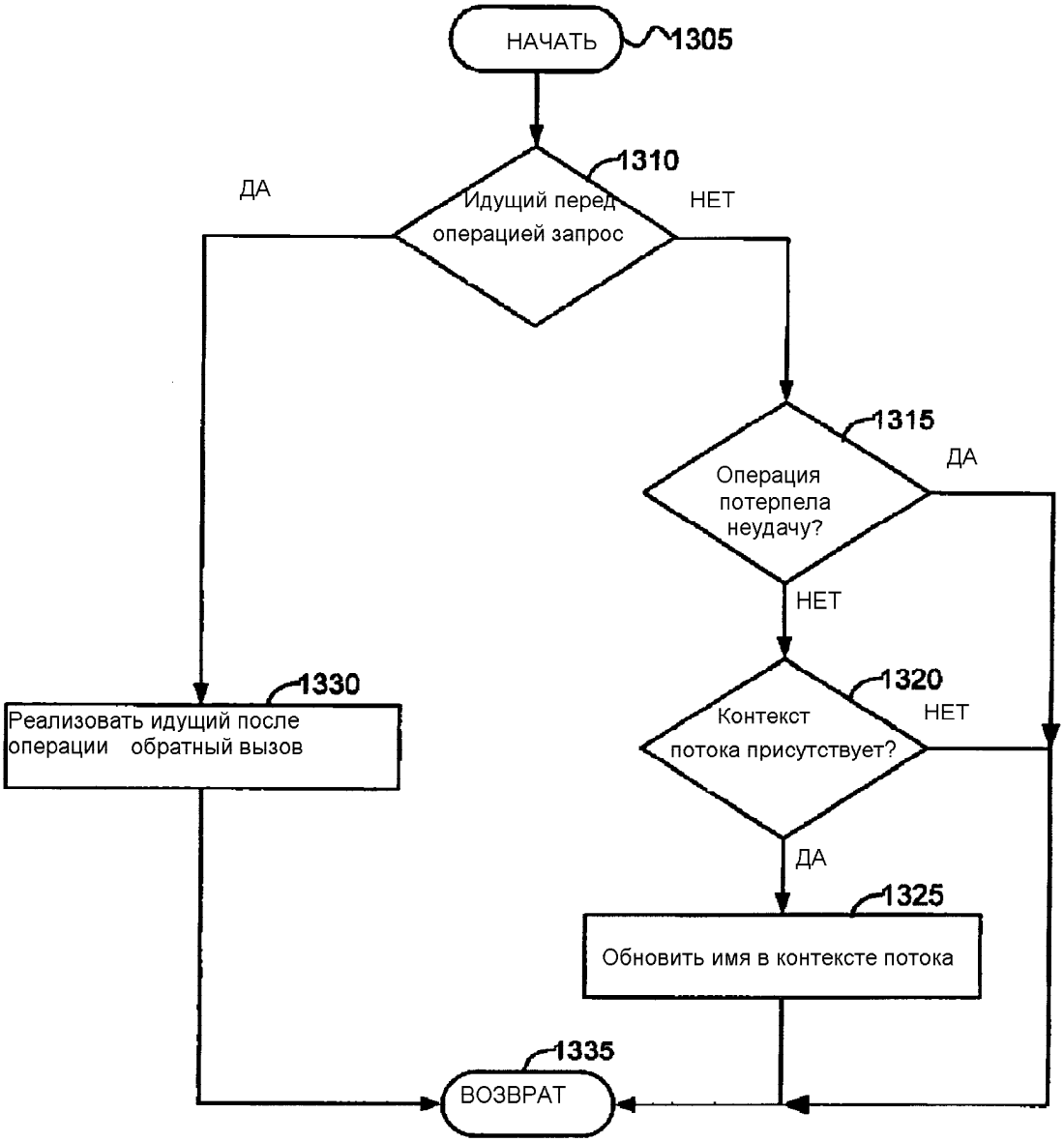
ФИГ.10


НАЧАТЬ
1105
1110
Операция создания для удаления при закрытии?
ДА
1115
Выполнить действия создания для удаления при закрытии (фиг.12)
НЕТ
1130
Операция задания состояния?
ДА
1135
Выполнить действия задания состояния (фиг.14)
НЕТ
1140
Операция очистки?
ДА
1145
Выполнить действия очистки (фиг.15)
НЕТ
1155
ВОЗВРАТ

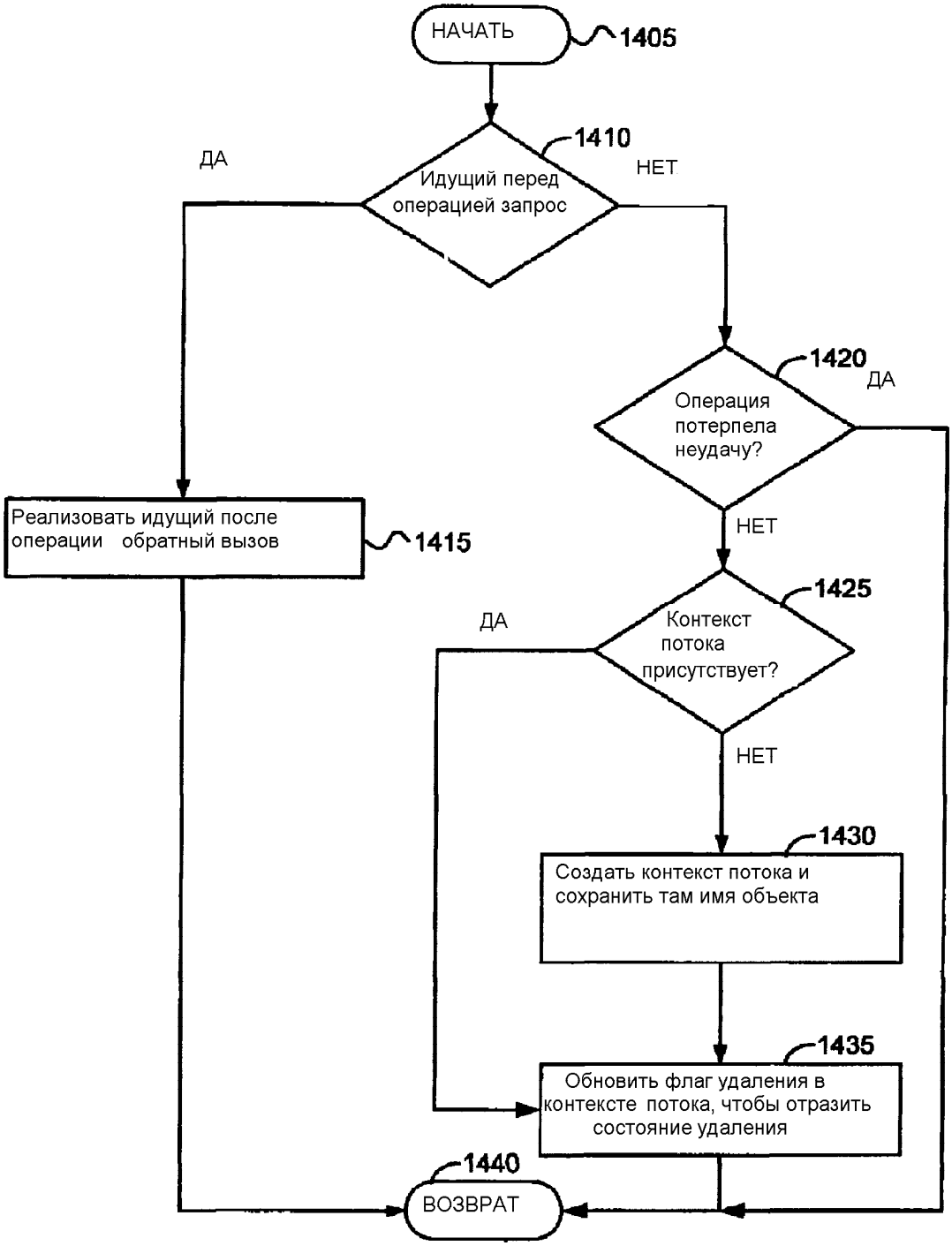
ФИГ.11


НАЧАТЬ
1205
1210
Идущий перед операцией запрос
ДА
НЕТ
1220
Операция потерпела неудачу?
ДА
НЕТ
1215
Навязать идущий после операции обратный вызов
1225
Выделить контекст потока, если не присутствует
1230
Сохранить имя объекта в контексте потока
1235
Установить флаг удаления в контексте потока
1240
ВОЗВРАТ

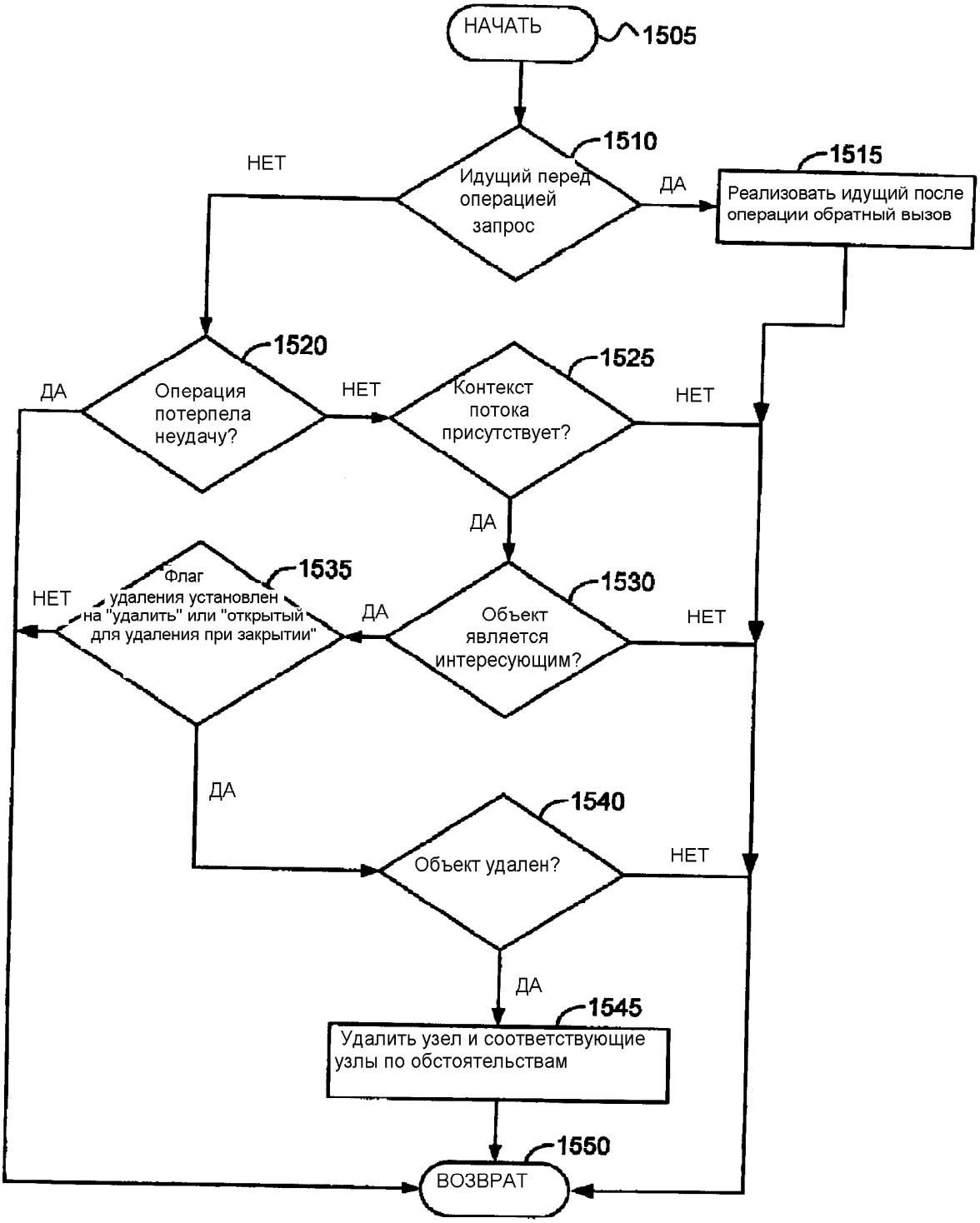
ФИГ.12


НАЧАТЬ
1305
1310
Идущий перед операцией запрос
ДА
НЕТ
1315
Операция потерпела неудачу?
ДА
НЕТ
1320
Контекст потока присутствует?
НЕТ
ДА
1325
Обновить имя в контексте потока
1330
Реализовать идущий после операции обратный вызов
1335
ВОЗВРАТ

ФИГ.13


НАЧАТЬ
1405
1410
Идущий перед операцией запрос
ДА
НЕТ
1420
Операция потерпела неудачу?
ДА
НЕТ
Реализовать идущий после операции обратный вызов
1415
1425
Контекст потока присутствует?
ДА
НЕТ
1430
Создать контекст потока и сохранить там имя объекта
1435
Обновить флаг удаления в контексте потока, чтобы отразить состояние удаления
1440
ВОЗВРАТ

ФИГ.14


НАЧАТЬ
1505
1510
Идущий перед операцией запрос
НЕТ
ДА
1515
Реализовать идущий после операции обратный вызов
1520
Операция потерпела неудачу?
ДА
НЕТ
1525
Контекст потока присутствует?
НЕТ
ДА
1530
Объект является интересующим?
НЕТ
ДА
1535
Флаг удаления установлен на "удалить" или "открытый для удаления при закрытии"
НЕТ
ДА
1540
Объект удален?
НЕТ
ДА
1545
Удалить узел и соответствующие узлы по обстоятельствам
1550
ВОЗВРАТ

ФИГ.15