



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21), (22) Заявка: **2005135846/09**, 17.11.2005(24) Дата начала отсчета срока действия патента:
17.11.2005(30) Конвенционный приоритет:
20.12.2004 US 11/018,916(43) Дата публикации заявки: **27.05.2007**(45) Опубликовано: **10.06.2010** Бюл. № 16(56) Список документов, цитированных в отчете о
поиске: **WO 02/19075 A2, 07.03.2002. RU 2145727**
C1, 20.02.2000. US 6243831 B1, 05.06.2001. US
5813009 A, 22.09.1998.

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):

ДЖОУНЗ Брайан М. (US),
ЛЮ Кэрл Л. (US),
РОТШИЛЛЕР Чэд Б. (US),
МАККАХИ Роберт Р. (US),
ВИЛЛАРОН Шон А. (US),
У Су-Пяо Билл (US)

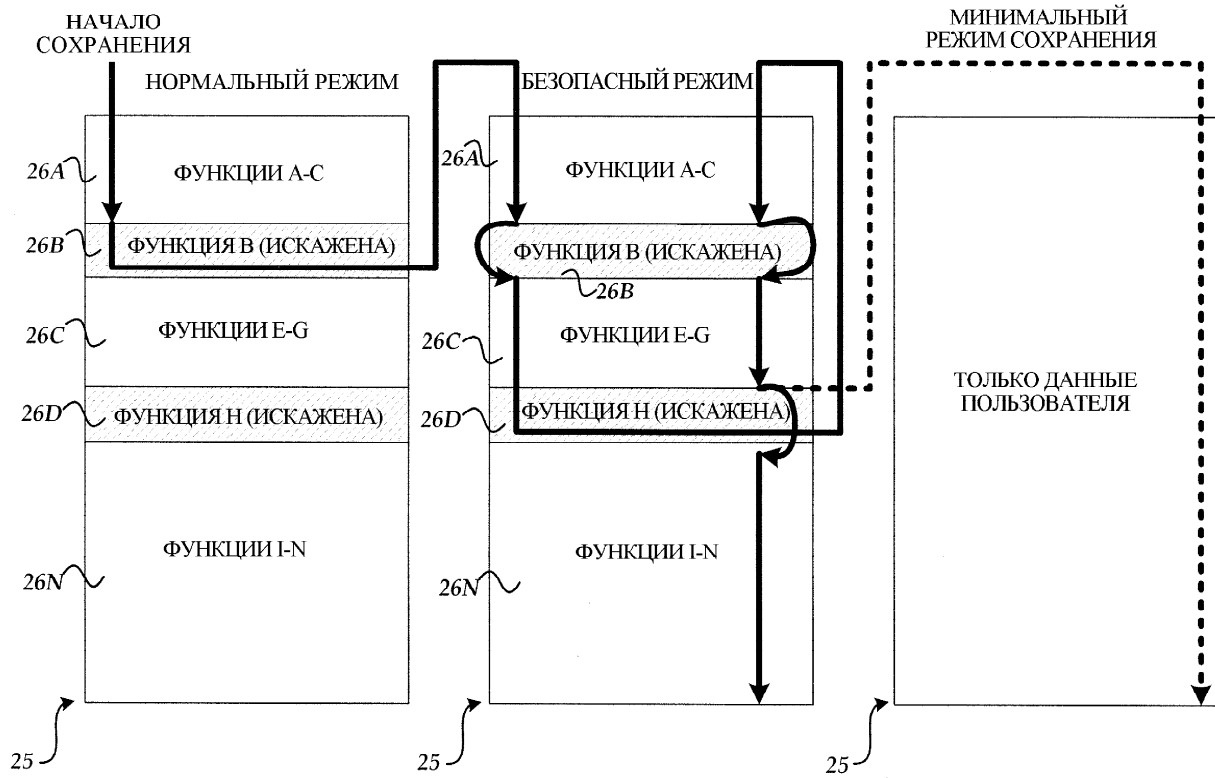
(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)**(54) СПОСОБ И МАШИНОЧИТАЕМЫЙ НОСИТЕЛЬ ДЛЯ ВЕРИФИКАЦИИ И СОХРАНЕНИЯ
ЭЛЕКТРОННОГО ДОКУМЕНТА**

(57) Реферат:

Изобретение относится к способу и машиночитаемому носителю для сохранения содержимого структуры памяти, запомненной в энергозависимой памяти. Техническим результатом является повышение защиты данных от уничтожения и искажения. Способ содержит следующие этапы: осуществляют попытку сохранения каждой части структуры памяти в первом режиме, причем в первом режиме над каждой из частей выполняют минимальную проверку целостности; определяют, является ли часть файла данных не поддающейся сохранению; и в ответ на определение того, что часть является не поддающейся сохранению, осуществляют

попытку сохранения содержимого структуры памяти во втором режиме, в котором над каждой из частей выполняют более подробную проверку целостности и в котором сохранение каждой не поддающейся сохранению части пропускают, причем во время попытки сохранения содержимого структуры памяти во втором режиме осуществляют определение того, может ли быть восстановлена не поддающаяся сохранению часть, и в ответ на определение того, что не поддающаяся сохранению часть может быть восстановлена, восстанавливают не поддающуюся сохранению часть и сохраняют восстановленную часть в файл данных. 2 н. и 8 з.п. ф-лы, 4 ил.



ФИГ.2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.
G11C 7/00 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2005135846/09**, 17.11.2005

(24) Effective date for property rights:
17.11.2005

(30) Priority:
20.12.2004 US 11/018,916

(43) Application published: **27.05.2007**

(45) Date of publication: **10.06.2010 Bull. 16**

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

(72) Inventor(s):

**DZhOUNZ Brajan M. (US),
LJu Kehrol L. (US),
ROTShILLER Chehd B. (US),
MAKKAKhI Robert R. (US),
VILLARON Shon A. (US),
U Su-Pjao Bill (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) METHOD AND MACHINE-READABLE CARRIER FOR VERIFICATION AND STORAGE OF ELECTRONIC DOCUMENT

(57) Abstract:

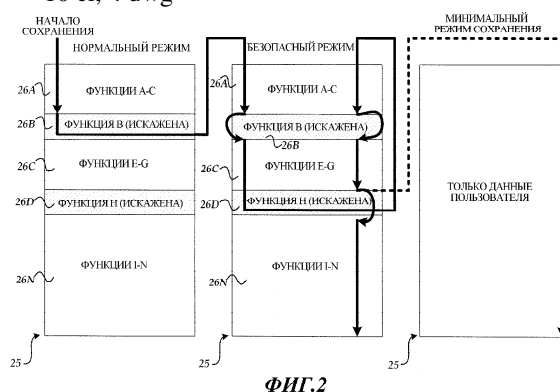
FIELD: information technologies.

SUBSTANCE: attempt is made to save each part of memory structure in the first mode, besides in the first mode a minimum verification of integrity is carried out for each part; it is identified whether part of data file is storable or not; and in response to identification of the fact of part being storable or not, an attempt is made to save content of memory structure in the second mode, in which a more detailed verification of integrity is carried out for each part, and in which storage of each non-storable part is omitted. At the same time in process of attempt to save content of memory structure in the second mode, it is identified whether the non-storable part may be recovered or not, and in response to identification of the fact of the non-

storable part being recoverable or not, its non-storable part is recovered, and recovered part is saved in data file.

EFFECT: improved protection of data against damage and distortion.

10 cl, 4 dwg



Данная заявка является родственной по отношению к одновременно поданной заявке на получение патента США 11/018,914 (Дело поверенного № 60001.0448US01), которая называется "Способ и Машиночитаемый Носитель для Загрузки Содержимого Файла данных", которая включена в настоящее описание в качестве ссылки.

УРОВЕНЬ ТЕХНИКИ

В сегодняшнем обществе компьютеры повсеместно используются для выполнения широкого множества задач и для целей развлечения. Например, в настоящее время компьютеры используются для игр, обмена информацией, исследований и для фактически бесконечного множества других применений. Одним из самых распространенных вариантов использования компьютеров как юридическими, так и физическими лицами является создание электронных и напечатанных документов. Существуют компьютерные прикладные программы для того, чтобы создавать все виды электронных документов, в том числе электронные таблицы, презентации, текстовые документы, графические документы, такие как диаграммы и цифровые изображения, документы автоматизированного проектирования и множество других типов электронных документов.

Электронные документы часто включают в себя содержимое, которое является очень важным. Кроме того, в случае утраты содержимого электронного документа во многих случаях было бы затруднительно, а то и вовсе невозможно восстановить данные. Например, чрезвычайно сложные юридические, деловые, маркетинговые и технические документы часто создаются таким образом, что они не могут быть легко восстановлены в случае искажения или уничтожения файла данных, хранящего документ. Даже в случаях, когда содержимое документа может быть легко восстановлено, для пользователя может быть совершенно неприемлемой потеря даже малой части его данных. Соответственно очень важно, чтобы данные, содержащиеся в электронных документах, были защищенными от уничтожения и искажения.

Современные компьютерные системы включают в себя проверку ошибок и другие механизмы для защиты от непреднамеренного искажения или потери системной памяти. К сожалению, даже с этими механизмами для документа, запомненного в энергозависимой системной памяти, местами существует возможность искажения до сохранения документа в файл данных в устройстве массовой памяти. Искажение может являться результатом поврежденной памяти, поврежденного контроллера памяти, ошибок управления памятью, загрузки поврежденных или искаженных данных, аварийного отказа прикладной программы, а также по другим причинам. Поскольку потеря какого бы то ни было объема данных может быть неприемлемой для пользователя и поскольку время и трудозатраты, необходимые для восстановления искаженного документа, часто являются очень большими, важно, чтобы так много данных, насколько это возможно, восстанавливались из искаженного документа, запомненного в энергозависимой памяти до сохранения содержимого памяти в устройстве массовой памяти.

По этим и другим соображениям были сделаны различные варианты осуществления настоящего изобретения.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

В соответствии с настоящим изобретением вышеупомянутые и другие проблемы решаются при помощи способа и машиночитаемого носителя для сохранения содержимого документа, запомненного в структуре памяти в энергозависимой памяти, в файл данных, запомненный в устройстве массовой памяти. При помощи

использования различных вариантов осуществления настоящего изобретения искаженные части структуры памяти (также называемые "записи") идентифицируют во время сохранения структуры памяти, и осуществляется попытка восстановления этих частей. Если искаженные части не могут быть восстановлены, сохранение искаженных частей пропускают. Неискаженные и восстановленные части структуры памяти затем сохраняют в файл данных в устройстве массовой памяти. Если части структуры памяти не могут быть восстановлены или пропущены, осуществляют попытку сохранения только данных пользователя, содержащихся в структуре памяти. Этим способом данные пользователя, содержащиеся в структуре памяти, могут быть сохранены в массовой памяти даже в случаях серьезного искажения в остатке структуры памяти.

В соответствии с одним аспектом изобретения способ обеспечивается для сохранения в файл данных в устройстве массовой памяти структуры памяти, запомненной в энергозависимой памяти, которая включает в себя одну или более частей. В соответствии со способом обеспечивается множество режимов сохранения. В "нормальном" режиме сохранения осуществляют попытку сохранения каждой части структуры памяти нормальным способом. Нормальный режим сохранения включает в себя минимальную проверку целостности каждой из частей структуры памяти таким образом, чтобы данные могли быть сохранены быстро. Если встречается такая часть структуры памяти, которая потеряна или искажена во время нормального режима, второй режим, называемый "безопасным" режимом сохранения, используется для того, чтобы попытаться сохранить части структуры памяти. Часть структуры памяти может быть сочтена искаженной и поэтому не поддающейся сохранению в следующих случаях: если она вызывает ошибку в прикладной программе или аварийный отказ прикладной программы, пытающейся сохранить ее, если часть включает в себя непредсказуемые значения данных, если часть представляет собой потерянные данные, если часть включает в себя недопустимые записи или недопустимый расширяемый язык разметки ("XML"), а также в других случаях.

В безопасном режиме сохранения над каждой частью структуры памяти выполняется подробная проверка целостности. В безопасном режиме сохранения также может быть осуществлена попытка восстановления искаженных частей структуры памяти. Любые части, которые могут быть восстановлены, затем сохраняются. При встрече в безопасном режиме сохранения с частью структуры памяти, которая потеряна или искажена и которая также не поддается восстановлению, сохранение неподдающейся восстановлению части пропускается. При встрече с частями структуры памяти, которые не поддаются восстановлению и для которых не может быть пропущено сохранение, используют третий режим сохранения, называемый "минимальным" режимом сохранения, при котором пытаются сохранить некоторые части структуры памяти.

В минимальном режиме сохранения сохраняют только части структуры памяти, которые включают в себя данные пользователя. Например, данные пользователя могут содержать текстовые данные или числовые данные, которые были введены пользователем. В качестве примера, если структура памяти содержит данные для электронной таблицы, попытку осуществляют в минимальном режиме сохранения, сохраняя только данные, содержащиеся в ячейках электронной таблицы. В минимальном режиме не осуществляют попыток сохранения других типов данных, которые могут содержаться в структуре памяти, таких как внедренные объекты, сводные таблицы, автофильтры, графики, стили, форматирования, а также

пользовательские настройки или настройки приложения.

В соответствии с другими вариантами осуществления изобретения также обеспечивается машиночитаемый носитель, на котором запомнены выполняемые компьютером команды. При выполнении компьютером выполняемых компьютером команд они вызывают обеспечение компьютером первого режима сохранения для сохранения содержимого структуры памяти, которая имеет одну или более частей. В первом режиме сохранения выполняют минимальную проверку целостности частей структуры памяти, поскольку они сохраняются в файл данных в устройстве массовой памяти. Выполняемые компьютером команды также вызывают обеспечение компьютером второго режима сохранения для сохранения структуры памяти, в которой по отношению к частям структуры памяти выполняется более подробная проверка целостности, чем в первом режиме сохранения. Во втором режиме сохранения также может быть осуществлена попытка восстановления частей, которые являются не поддающимися сохранению. Кроме того, во втором режиме сохранения сохранение любых не поддающихся сохранению частей пропускают.

Выполняемые компьютером команды также заставляют компьютер начинать сохранение содержимого структуры памяти в первом режиме сохранения. Если часть структуры памяти определена как не поддающаяся сохранению в первом режиме сохранения, компьютер переключается во второй режим сохранения и пытается сохранить структуру памяти в этом режиме сохранения. Если во втором режиме сохранения сталкиваются с не поддающейся сохранению частью, которая может быть восстановлена, не поддающаяся сохранению часть восстанавливается и сохраняется. Если не поддающаяся сохранению часть не может быть восстановлена, сохранение не поддающейся сохранению части пропускается.

В соответствии с вариантом осуществления изобретения выполняемые компьютером команды также заставляют компьютер обеспечивать третий режим сохранения, в котором сохраняются только те части структуры памяти, которые включают в себя данные пользователя. Если во втором режиме сохранения определено, что часть структуры памяти является не поддающейся сохранению и что не поддающаяся сохранению часть не может быть восстановлена или пропущена, осуществляют попытку сохранения содержимого структуры памяти в третьем режиме сохранения.

Изобретение может быть осуществлено в виде компьютерного процесса, компьютерной системы или в виде готового изделия, такого как компьютерный программный продукт или машиночитаемый носитель. Компьютерный программный продукт может представлять собой компьютерный носитель данных, считываемый компьютерной системой, и закодированную форму компьютерной программы команд для выполнения компьютерного процесса. Компьютерный программный продукт также может представлять собой распространяемый сигнал на носителе, считываемом компьютерной системой, и закодированную форму компьютерной программы команд для выполнения компьютерного процесса.

Эти и различные другие особенности, так же как и преимущества, которые характеризуют настоящее изобретение, будут очевидны из прочтения нижеследующего подробного описания и просмотра соответствующих чертежей.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Фиг.1 представляет собой диаграмму архитектуры компьютерной системы, иллюстрирующей используемую компьютерную систему, используемую с различными вариантами осуществления изобретения;

Фиг.2 представляет собой блок-схему, иллюстрирующую аспекты структуры памяти и различных режимов сохранения, обеспечиваемых вариантами осуществления изобретения; и

Фиг.3А-3В представляют собой схемы последовательности операций, показывающие иллюстративный процесс для сохранения структуры памяти в соответствии с различными вариантами осуществления изобретения.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

Теперь обратимся к чертежам, на которых одни и те же номера позиций представляют одни и те же элементы, и будут описаны различные аспекты настоящего изобретения. В частности, Фиг.1 и соответствующая ей часть описания предназначены для того, чтобы обеспечить краткое изложение, общее описание подходящей вычислительной среды, в которой могут быть осуществлены варианты осуществления изобретения. Несмотря на то что изобретение будет описано в общем контексте программных модулей, которые выполняются в соединении с программными модулями, которые выполняются в операционной системе на персональном компьютере, специалистам в данной области техники будет понятно, что изобретение также может быть осуществлено в комбинации с другими типами компьютерных систем и программных модулей.

Вообще, программные модули включают в себя подпрограммы, программы, компоненты, структуры данных и другие типы структур, которые выполняют конкретные задачи или осуществляют определенные абстрактные типы данных. Кроме того, специалистам в данной области техники будет понятно, что изобретение может быть осуществлено другими конфигурациями компьютерной системы, включающими в себя наладонные устройства, многопроцессорные системы, программируемую бытовую электронику на основе микропроцессоров, миникомпьютеры, главные компьютеры и т.п. Изобретение также может быть осуществлено в распределенных вычислительных средах, в которых задачи выполняются удаленными устройствами обработки данных, которые соединены через сеть передачи данных. В распределенной вычислительной среде программные модули могут быть расположены как в локальных, так и в удаленных запоминающих устройствах.

Обращаясь теперь к Фиг.1, увидим, что будет описана иллюстративная компьютерная архитектура применительно к компьютеру 2, используемому в различных вариантах осуществления изобретения. Компьютерная архитектура, показанная на Фиг.1, иллюстрирует обычный настольный или портативный компьютер, включающий в себя центральное процессорное устройство 5 (ЦПУ, CPU), системную память 7, включающую в себя оперативное запоминающее устройство 9 (ОЗУ, RAM) и постоянное запоминающее устройство 11 (ПЗУ, ROM), и системную шину 12, которая соединяет память с центральным процессором 5. Базовая система ввода-вывода, содержащая основные подпрограммы, которые помогают передавать информацию между элементами в пределах компьютера, например, при начальном запуске, хранится в ПЗУ 11. Компьютер 2 дополнительно включает в себя устройство 14 массовой памяти для хранения операционной системы 16, прикладных программ и других программных модулей, которые более подробно будут описаны ниже.

Устройство 14 массовой памяти связано с центральным процессором 5 через контроллер массовой памяти (не показан), связанный с шиной 12. Устройство 14 массовой памяти и связанный с ним машиночитаемый носитель обеспечивают

энергонезависимое хранилище данных для компьютера 2. Несмотря на то что содержащееся здесь описание машиночитаемых носителей относится к устройству массовой памяти, такому как жесткий диск или привод для компакт-дисков (CD-ROM), специалистам в данной области техники будет понятно, что машиночитаемые носители могут являться любыми доступными носителями, к которым может обращаться компьютер 2.

Только в качестве примера, а не ограничения машиночитаемые носители могут включать в себя компьютерные носители данных и среду передачи информации.

Компьютерные носители данных включают в себя энергозависимые и энергонезависимые, сменные и несменные носители, осуществленные любым способом или технологией для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули или другие данные.

Компьютерные носители данных включают в себя, но не ограничиваясь перечисленным, ОЗУ, ПЗУ, стираемое программируемое постоянное запоминающее устройство (СППЗУ, EPROM), электрически стираемое программируемое постоянное запоминающее устройство (ЭСППЗУ, EEPROM), флэш-память или память, основанную на другой твердотельной технологии, CD-ROM, универсальные цифровые диски (DVD) или другие оптические носители данных, магнитные кассеты, магнитную ленту, накопители на магнитных дисках или другие магнитные носители данных либо любые другие носители, которые могут использоваться для хранения требуемой информации, к которым может обращаться компьютер 2.

В соответствии с различными вариантами осуществления изобретения компьютер 2 может функционировать в сетевом окружении, используя логические подключения к удаленным компьютерам через сеть 18, такую как Интернет. Компьютер 2 может подключаться к сети 18 через модуль 20 сетевого интерфейса, связанный с шиной 12. Следует понимать, что модуль 20 сетевого интерфейса также может использоваться для подключения к сетям других типов и к удаленным вычислительным системам. Компьютер 2 также может включать в себя контроллер 22 ввода-вывода для приема и обработки вводимых данных от множества других устройств, таких как клавиатура, мышь или электронное перо (не показанное на Фиг.1). Точно так же контроллер 22 ввода-вывода может обеспечивать вывод на экран дисплея, принтер или устройство вывода другого типа.

Как вкратце упомянуто выше, множество программных модулей и файлов данных могут храниться в устройстве 14 массовой памяти и ОЗУ 9 компьютера 2, включая в себя операционную систему 16, подходящую для управления функционированием сетевого персонального компьютера, такую как операционная система WINDOWS XP от MICROSOFT CORPORATION из города Редмонд, штат Вашингтон. Устройство 14 массовой памяти и ОЗУ 9 также могут хранить один или более программных модулей. В частности, устройство 14 массовой памяти и ОЗУ 9 могут хранить прикладную программу 10 электронной таблицы. Как известно специалистам в данной области техники, прикладная программа 10 электронной таблицы функционирует, обеспечивая функциональные возможности для создания и редактирования электронных таблиц.

В соответствии с одним вариантом осуществления изобретения прикладная программа 10 электронной таблицы включает в себя прикладную программу электронной таблицы EXCEL от MICROSOFT CORPORATION. Однако следует понимать, что для воплощения различных аспектов настоящего изобретения могут использоваться другие прикладные программы электронной таблицы от других изготовителей. Также следует понимать, что несмотря на то что варианты

осуществления описанного здесь изобретения представлены в контексте прикладной программы электронной таблицы, изобретение может использоваться с любым другим типом прикладной программы, которая сохраняет данные в файл данных.

Например, варианты осуществления описанного здесь изобретения могут использоваться в прикладной программе обработки текстов, прикладной программе презентаций, чертежной прикладной программе, или прикладной программе автоматизированного проектирования, или в прикладной программе базы данных.

Как показано на Фиг.1, части прикладной программы 10 электронной таблицы могут быть загружены в энергозависимое ОЗУ 9 во время выполнения. Кроме того, вместе с созданием и редактированием документа электронной таблицы прикладная программа 10 электронной таблицы может использовать часть ОЗУ 9 для хранения документа. В частности, прикладная программа 10 электронной таблицы может использовать одну или более структур 25 памяти для хранения данных, представляющих документ электронной таблицы. Время от времени либо в ответ на запрос пользователя, либо автоматизированным способом прикладная программа 10 электронной таблицы функционирует, сохраняя содержимое структур 25 памяти в файл 24 данных, запоминаемый в устройстве 14 массовой памяти. Файл 24 данных содержит данные, представляющие различные аспекты документа электронной таблицы, такие как данные пользователя, включающие в себя содержимое ячеек электронной таблицы, настройки приложения, информацию о форматировании и другие данные, соответствующие различным функциям, обеспеченным прикладной программой 10 электронной таблицы. Как более подробно будет описано ниже относительно Фиг.2-3В, способ для сохранения содержимого структур 25 памяти в файл 24 данных используется прикладной программой 10 электронной таблицы, который подсчитывает возможность искажения в структурах 25 памяти и который пытается довести до максимума объем данных пользователя, которые сохраняются в файл 24 данных, даже если структуры 25 памяти становятся искаженными.

Обратившись теперь к Фиг.2, увидим, что в соответствии с содержимым структур 25 памяти и функционированием механизма сохранения, используемого прикладной программой 10 электронной таблицы, будут обеспечиваться дополнительные детали. Как показано на Фиг.2, структуры 25 памяти подразделяются на множество частей 26А-26N. Каждая из частей 26А-26N используется для запоминания информации, касающейся одной или более функций, поддерживаемых прикладной программой 10 электронной таблицы. Кроме того, информация для других, но связанных функций может быть сохранена в одной отдельно взятой из частей 26А-26N.

Например, как показано на Фиг.2, данные для функций А-С сохраняются в части 26А. Данные для функции D сохраняются в части 26В. Данные для функций E-G сохраняются в части 26С и так далее. Данные пользователя могут быть запомнены в любой из частей 26А-26N. Следует понимать что данные, запомненные в структурах 25 памяти, могут быть запомнены способом, состоящим из нескольких не следующих друг за другом операций, и что данные для связанных функций могут быть запомнены в отдельных ячейках памяти.

Как кратко описано выше и показано на Фиг.2, это возможно для содержащихся в частях 26А-26N данных, являющихся искаженными. Искажение может являться результатом поврежденной памяти, поврежденного контроллера памяти, ошибок управления памятью, загрузки поврежденных или искаженных данных, аварийного отказа прикладной программы, а также по другим причинам. Данные для конкретной части также могут быть определены как потерянные. Часть структуры 25 памяти

может быть сочтена искаженной и поэтому не поддающейся запоминанию в следующих случаях: если она вызывает ошибку в прикладной программе или аварийный отказ прикладной программы, пытающейся сохранить ее, если часть включает в себя непредсказуемые значения данных, если часть представляет собой

5 потерянные данные, если часть включает в себя недопустимые записи или недопустимый расширяемый язык разметки ("XML"), а также в других случаях. В иллюстративных структурах 25 памяти, показанных на Фиг.2, части 26B и 26D стали искаженными.

10 Как описано здесь, части структур 25 памяти являются поддающимися сохранению прикладной программой 10 электронной таблицы несмотря на искажения частей 26B и 26D. Фиг.2 также иллюстрирует этот процесс сохранения, использующий иллюстративные структуры 25 памяти. В частности прикладная программа 10

15 электронной таблицы начинает сохранение структуры 25 памяти в нормальном режиме сохранения. В нормальном режиме сохранения по отношению к частям 26A-26N выполняется минимальная проверка целостности структур 25 памяти. При встрече с искаженной частью структур 25 памяти во время сохранения в нормальном режиме прикладная программа 10 электронной таблицы переключается в безопасный

20 режим сохранения и начинает сохранение структур 25 памяти с начала. Например, как показано на Фиг.2, при встрече с искаженной частью 26B в нормальном режиме сохранения режим сохранения изменяют на безопасный режим сохранения и сохранение начинается снова с начала структур 25 памяти. Следует понимать, что в соответствии с вариантами осуществления изобретения сохранение дополнительных

25 частей структур 25 памяти не требует возвращения к началу.

В безопасном режиме сохранения по отношению к частям 26A-26N структур 25 памяти выполняют дополнительную по сравнению с нормальным режимом сохранения проверку целостности. Дополнительно, при встрече с искаженной частью

30 во время сохранения в безопасном режиме осуществляют попытку восстановления искаженной части. Если искаженная часть может быть восстановлена, эту часть сохраняют. Если искаженная часть не может быть восстановлена, то сохранение искаженной части пропускают. Например, как показано на Фиг.2, часть 26B искажена и не может быть восстановлена. Поэтому сохранение части 26B пропускают и

35 сохраняют часть 26C.

После того как часть 26C была сохранена, осуществляют попытку сохранения части 26D. Однако, как показано на Фиг.2, часть 26D искажена. Соответственно осуществляют попытку восстановления части 26D. Если часть 26D не может быть

40 сохранена, сохранение части 26D пропускают, и этот процесс продолжается до тех пор, пока остающиеся части не будут сохранены или пропущены. В соответствии с вариантом осуществления изобретения сохранение структур 25 памяти может возвратиться к началу структур 25 памяти после встречи с не поддающейся

45 сохранению частью и определения ее как не поддающейся восстановлению. Это проиллюстрировано на Фиг.2. Возвращение к началу структур 25 памяти этим способом предоставляет возможность сохранения других частей файла 24 данных, которые связаны с не поддающейся сохранению частью, которая будет пропущена даже при том, что связанные части, возможно, не искажены.

50 Если во время сохранения структур 25 памяти встречаются с частью, которая является не поддающейся сохранению и не поддающейся восстановлению, прикладная программа 10 электронной таблицы может переключиться в третий режим сохранения, называемый минимальным режимом сохранения. В минимальном режиме

сохранения осуществляют попытку сохранения из структур 25 памяти только данных пользователя. В частности относительно текстового документа осуществляют попытку сохранения только текста документа. Относительно документа электронной таблицы, осуществляют попытку сохранения содержимого ячеек электронной

5 таблицы, включающих в себя данные, введенные пользователем, формулы и данные, сгенерированные формулами. Этим способом, даже если части структур 25 памяти искажены, некоторые или все данные пользователя могут быть восстановлены и сохранены. Этот процесс проиллюстрирован на Фиг.2 пунктирной линией и

10 выполняется в случае, если часть 26D была определена как не поддающаяся сохранению и не поддающаяся восстановлению. Дополнительные детали в соответствии с этим процессом предоставлены ниже относительно Фиг.3А-3В.

Обращаясь теперь к Фиг.3А-3В, увидим, что будет описана подпрограмма 300, иллюстрирующая процесс, выполняемый прикладной программой 10 электронной

15 таблицы для сохранения содержимого структур 25 памяти. При прочтении описания представленных здесь подпрограмм следует понимать, что логические операции различных вариантов осуществления настоящего изобретения осуществляются (1) в качестве последовательности осуществляемых компьютером действий или

20 программных модулей, выполняющихся на компьютерной системе и/или (2) в качестве взаимосвязанных логических схем вычислительной машины или схемных модулей в компьютерной системе. Конкретный вариант выполнения выбирается в зависимости от требований к рабочим характеристикам компьютерной системы, осуществляющей изобретение. Соответственно логические операции, проиллюстрированные на Фиг.3А-

25 3В, и составляющие описанные здесь варианты осуществления настоящего изобретения упоминаются как операции, структуры, действия или модули. Специалисту в данной области техники будет понятно, что эти операции, структуры, действия и модули могут быть осуществлены при помощи программного обеспечения,

30 программно-аппаратных средств, цифровых логических схем специального назначения и любой комбинации вышеперечисленного без отклонения от сущности или объема настоящего изобретения, изложенных в сформулированной далее формуле изобретения.

Следует понимать, что подпрограмма 300 при своем функционировании использует

35 несколько переменных. В частности переменная "режим" отслеживает текущий режим сохранения. Эта переменная может принимать значения "безопасный", "нормальный" или "минимальный". Переменная «счетчик пропусков» отслеживает структуры 25 памяти, которые должны быть пропущены в случае, когда сохранение структур 25

40 памяти возвращается к началу после встречи с искаженной частью. Переменная "число пропускаемых записей" описывает текущее количество секций, которые должны быть пропущены на текущей попытке сохранения. Переменная "текущая запись" идентифицирует текущую секцию в обрабатываемом файле данных. Следует понимать, что для выполнения одной и той же задачи может использоваться большее

45 или меньшее количество переменных. Кроме того, следует понимать, что проиллюстрированная на Фиг.3А и 3В подпрограмма 300 представляет только один возможный вариант осуществления изобретения и что специалисту в данной области техники будет очевидно множество других вариантов осуществления.

Подпрограмма 300 начинается с любой из операций 302, 304 или 306. В частности, в соответствии с вариантами осуществления изобретения может быть обеспечен пользовательский интерфейс, который предоставляет пользователю возможность выбора, сохранять ли документ в нормальном режиме (операция 304), в безопасном

режиме сохранения (операция 302) или в минимальном режиме сохранения (306). Этот пользовательский интерфейс может быть представлен пользователю тогда, когда пользователь запрашивает сохранение документа. Основываясь на выборе пользователя в пользовательском интерфейсе, подпрограмма 300 начинает свое функционирование с любой из операций 302, 304 или 306.

Если сохранение начинается в безопасном режиме сохранения, подпрограмма 300 начинается с операции 302, на которой переменная режима имеет значение "безопасный". Затем подпрограмма 300 переходит к операции 308. Если сохранение должно начаться в нормальном режиме сохранения, подпрограмма 300 начинается с операции 304, на которой переменная режима имеет значение "нормальный". Затем подпрограмма 300 переходит от операции 304 к операции 308. Если сохранение должно начаться в минимальном режиме сохранения, подпрограмма начинается с операции 306, на которой переменная режима имеет значение "минимальный". От операции 306 подпрограмма 300 переходит к описанной ниже операции 348.

На операции 308 инициализируют переменную «счетчик пропусков», указывая, что никакие записи не должны быть пропущены. Затем подпрограмма 300 переходит к операции 310, на которой текущей записи устанавливают значение, равное первой записи в структурах памяти. Переменную «число пропускаемых записей» устанавливают равной числу пропускаемых записей. На первом проходе ее устанавливают на число пропускаемых записей, равное нулю. От операции 310 подпрограмма 300 переходит к операции 312.

На операции 312 осуществляют попытку сохранения текущей записи в текущем режиме. Например, если переменная режима равна "нормальному", то по отношению к сохраняемой секции выполняют минимальную проверку целостности. Если переменная режима равна "безопасному", выполняют дополнительную проверку целостности. От операции 312 подпрограмма 300 переходит к операции 314, на которой осуществляют определение того, является ли текущая запись не поддающейся сохранению (то есть либо искаженной, либо потерянной). Если текущая запись поддается сохранению, подпрограмма 300 переходит к операции 316, на которой осуществляется определение того, остались ли еще сохраняемые записи. Если существуют другие записи, подпрограмма 300 переходит от операции 316 к операции 318, на которой переменная "текущая запись" указывает на следующую запись в структурах памяти 25. Затем подпрограмма 300 переходит к операции 312, на которой сохраняют следующую запись. Если на операции 316 определено, что не осталось дополнительных сохраняемых записей, подпрограмма 300 переходит к операции 320, на которой она завершается. Этим способом, если не существуют искаженные или потерянные записи, то все записи сохраняются в текущем режиме.

Следует понимать, что в вариантах осуществления изобретения некоторые проверки целостности могут быть выполнены на уровне функций в отличие от уровня записей. Для выполнения такой проверки целостности на уровне функций осуществляют попытку сохранения всех записей для конкретной функции. Затем осуществляют определение того, допустимы ли данные для функции. Если данные являются недопустимыми, структуру данных с пропусками обновляют записями для пропускаемой функции и осуществляют другую попытку сохранения файла. Подобным способом также могут осуществляться проверки целостности на файловом уровне.

Если на операции 314 определено, что текущая запись является не поддающейся сохранению, подпрограмма 314 переходит к операции 322, на которой осуществляют

определение того, является ли текущий режим нормальным режимом. Если текущий режим является нормальным режимом, то подпрограмма 300 переходит к операции 324, на которой переменная «счетчик пропусков» обновляется, указывая на то, что часть структур 25 памяти была идентифицирована как, возможно, требующая
 5 быть пропущенной. Затем подпрограмма 300 переходит к операции 324, на которой переменная режима принимает значение "безопасный". Этим способом после столкновения с не поддающейся сохранению частью структур 25 памяти, режим сохранения переключается с нормального на безопасный режим. Затем
 10 подпрограмма 300 возвращается назад к операции 310, на которой обработка файла данных возвращается к началу.

Если на операции 322 определено, что текущий режим сохранения не является нормальным режимом, подпрограмма 300 переходит к операции 328, на которой
 15 осуществляют определение того, является ли текущий режим сохранения безопасным режимом. Поскольку в этой части подпрограммы 300 возможными значениями должны быть только нормальные или безопасные режимы сохранения, подпрограмма переходит к операции 330, на которой происходит возврат по ошибке в случае, если текущий режим сохранения не является безопасным режимом. Затем
 20 подпрограмма 300 переходит от операции 330 к операции 320, на которой она завершается. Однако если на операции 328 определено, что текущий режим является безопасным режимом, подпрограмма 300 переходит к операции 332.

На операции 332 осуществляют попытку восстановления текущей записи. На операции 334 осуществляют определение того, является ли текущая запись
 25 поддающейся восстановлению. Если запись является поддающейся восстановлению, подпрограмма 300 переходит к операции 336, на которой сохраняется текущая запись. На операции 336 переменная «счетчик пропусков» также обновляется, указывая, что сохранение текущей записи не должно быть пропущено, поскольку запись является
 30 поддающейся восстановлению. От операции 336 подпрограмма 300 переходит назад, к операции 316, на которой оставшаяся часть записей структур 25 памяти обрабатывается описанным выше способом.

Если на операции 334 определено, что текущая запись не может быть восстановлена, то подпрограмма 300 переходит к операции 338. На операции 338
 35 осуществляют определение того, равно ли число пропускаемых записей нулю. Это могло бы иметь место в случае, когда сохранение было начато в нормальном режиме, на котором встретились с первой искаженной записью, и запись является не поддающейся восстановлению. В этом случае подпрограмма 300 переходит к
 40 операции 340, на которой переменная «счетчик пропусков» обновляется, указывая, что запись должна быть пропущена. Затем подпрограмма 300 возвращается к операции 310, на которой обработка структур 25 памяти возвращается к началу описанным выше способом.

Если на операции 338 определено, что переменная «число пропускаемых записей»
 45 не равна нулю, подпрограмма 300 переходит к операции 342, на которой осуществляют попытку пропуска сохранения текущей записи. На операции 344 осуществляют определение того, может ли быть пропущено сохранение текущей записи. Если сохранение текущей записи может быть пропущено, подпрограмма 300
 50 переходит к операции 346, на которой запись снабжают признаком в виде переменной «пропустить запись». Затем подпрограмма переходит к описанной выше операции 316.

Если на операции 344 определено, что текущая запись не может быть пропущена, подпрограмма 300 переходит к операции 306, на которой переменная «режим»

принимает значение "минимальный". Затем подпрограмма 300 переходит к операции 348, на которой осуществляют попытку сохранения структуры 25 памяти в минимальном режиме. Как описано выше, в минимальном режиме сохраняются только данные пользователя. Кроме того, если данные пользователя также искажены, осуществляют попытку сохранения настолько большого объема данных пользователя, насколько это возможно. Затем подпрограмма 300 переходит к операции 320, на которой она завершается.

Основываясь на вышеупомянутом, следует понимать, что различные варианты осуществления изобретения включают в себя способ, систему, устройство и машиночитаемый носитель для сохранения содержимого документа, запомненного в структуре энергозависимой памяти, в файл данных, запоминаемый в устройстве массовой памяти. Вышеупомянутое подробное описание, примеры и данные обеспечивают полное описание изготовления и использования сущности изобретения. Так как без отступления от духа или границ изобретения может быть сделано множество вариантов осуществления изобретения, изобретение заключено в изложенной ниже формуле изобретения.

Формула изобретения

1. Способ для сохранения содержимого структуры памяти, запомненной в энергозависимой памяти и имеющей одну или более частей, в файл данных в устройстве массовой памяти, причем способ содержит следующие действия:

осуществляют попытку сохранения каждой части структуры памяти в первом режиме, причем в первом режиме над каждой из частей выполняют минимальную проверку целостности;

в первом режиме определяют, является ли часть файла данных не поддающейся сохранению; и

в ответ на определение того, что часть является не поддающейся сохранению, осуществляют попытку сохранения содержимого структуры памяти во втором режиме, в котором над каждой из частей выполняют более подробную проверку целостности и в котором сохранение каждой не поддающейся сохранению части пропускают, причем во время попытки сохранения содержимого структуры памяти во втором режиме осуществляют определение того, может ли быть восстановлена не поддающаяся сохранению часть, и в ответ на определение того, что не поддающаяся сохранению часть может быть восстановлена, восстанавливают не поддающуюся сохранению часть и сохраняют восстановленную часть в файл данных.

2. Способ по п.1, в котором дополнительно пропускают сохранение не поддающейся сохранению части в ответ на определение того, что не поддающаяся сохранению часть не может быть восстановлена.

3. Способ по п.2, дополнительно содержащий следующие действия:

определяют, может ли быть пропущено сохранение не поддающейся сохранению части; и

в ответ на определение того, что сохранение не поддающейся сохранению части не может быть пропущено, осуществляют попытку сохранения содержимого структуры памяти в третьем режиме, причем в третьем режиме в файл данных сохраняют только части структуры памяти, соответствующие данным пользователя.

4. Способ по п.3, в котором данные пользователя включают в себя введенные пользователем текстовые данные.

5. Способ по п.3, в котором данные пользователя включают в себя введенные

пользователем числовые данные.

6. Машиночитаемый носитель, хранящий запомненные на нем выполняемые компьютером команды, которые при выполнении их компьютером заставляют компьютер выполнять следующие действия:

обеспечивают первый режим сохранения для сохранения содержимого структуры памяти, имеющей одну или более частей, из энергозависимой памяти в файл данных в устройстве массовой памяти, причем при сохранении в первом режиме сохранения над каждой из частей выполняют минимальную проверку целостности;

обеспечивают второй режим сохранения для сохранения содержимого структуры памяти в файл данных, в котором над каждой из частей выполняют более подробную проверку целостности и в котором сохранение каждой не поддающейся сохранению части пропускают;

начинают сохранение содержимого структуры памяти в первом режиме сохранения; и

при функционировании в первом режиме сохранения определяют, является ли часть структуры памяти не поддающейся сохранению, и в ответ на определение того, что часть является не поддающейся сохранению, переключаются во второй режим сохранения, причем во втором режиме сохранения осуществляют определение того, может ли не поддающаяся сохранению часть быть восстановлена, и в ответ на определение того, что не поддающаяся сохранению часть может быть восстановлена, восстанавливают не поддающуюся сохранению часть и сохраняют восстановленную часть в файл данных.

7. Машиночитаемый носитель по п.6, в котором во втором режиме сохранения сохранение не поддающейся сохранению части пропускают в ответ на определение того, что не поддающаяся сохранению часть не может быть восстановлена.

8. Машиночитаемый носитель по п.7, содержащий дополнительные машиночитаемые команды, которые при выполнении их компьютером заставляют компьютер выполнять следующие действия:

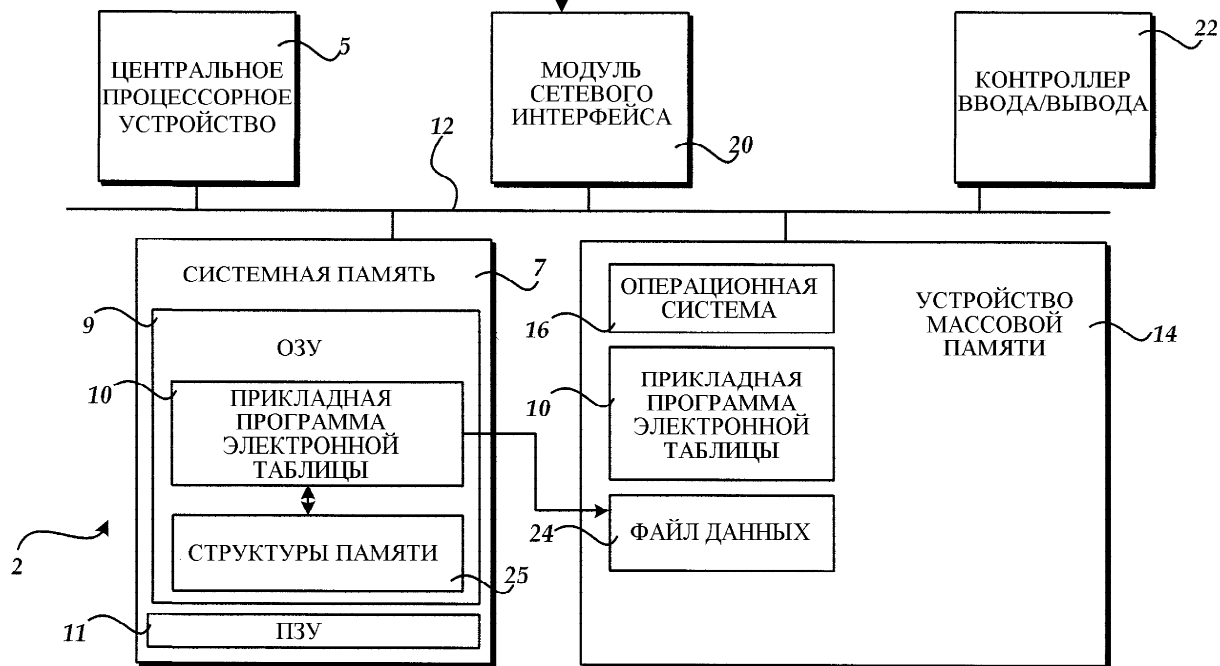
обеспечивают третий режим сохранения, причем в третьем режиме в файл данных сохраняют только те части структуры памяти, которые соответствуют данным пользователя;

при функционировании во втором режиме сохранения определяют, может ли сохранение не поддающейся сохранению части быть пропущено; и

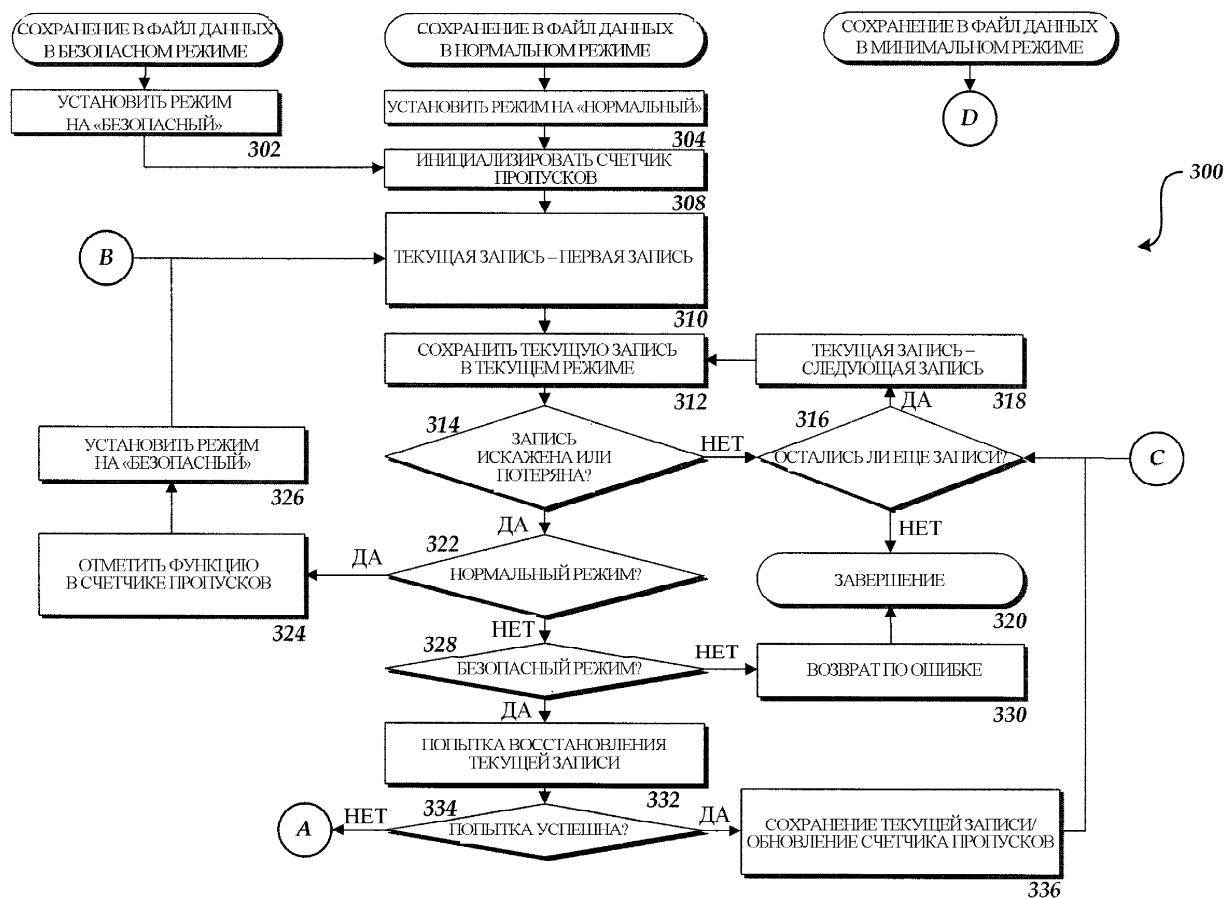
в ответ на определение того, что сохранение не поддающейся сохранению части не может быть пропущено, переключаются в третий режим сохранения.

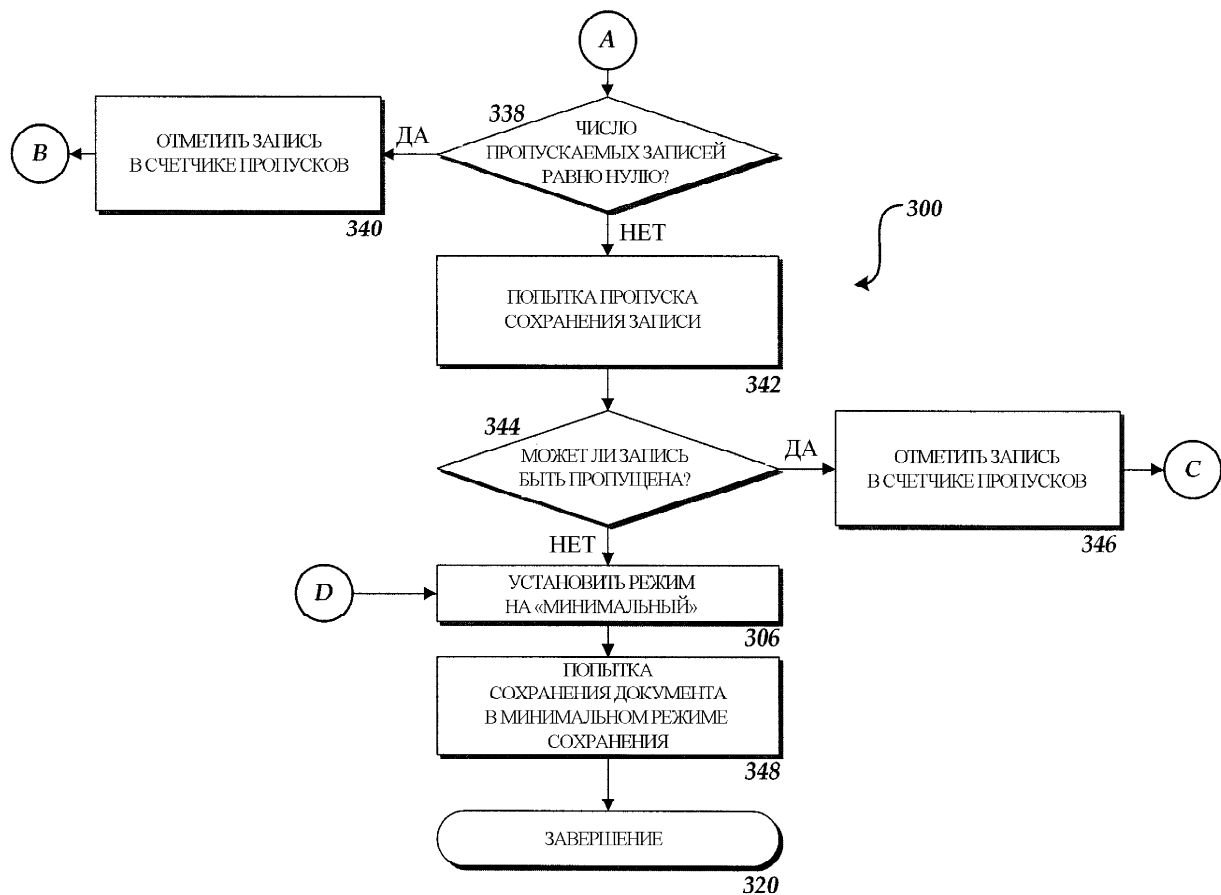
9. Машиночитаемый носитель по п.8, в котором данные пользователя содержат введенные пользователем текстовые данные.

10. Машиночитаемый носитель по п.9, в котором данные пользователя содержат введенные пользователем числовые данные.



ФИГ.1

**ФИГ.3А**



ФИГ.3В