



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 17/00 (2020.02); G06F 21/57 (2020.02); G06F 21/572 (2020.02); G06F 21/575 (2020.02); H04L 9/3247 (2020.02)

(21)(22) Заявка: 2018136581, 17.10.2018

(24) Дата начала отсчета срока действия патента:
17.10.2018

Дата регистрации:
23.04.2020

Приоритет(ы):

(30) Конвенционный приоритет:
18.10.2017 JP 2017-201956

(43) Дата публикации заявки: 17.04.2020 Бюл. № 11

(45) Опубликовано: 23.04.2020 Бюл. № 12

Адрес для переписки:

129090, Москва, ул. Большая Спасская, д. 25,
строение 3, ООО "Юридическая фирма
Городисский и Партнеры"

(72) Автор(ы):

СИМИДЗУ, Сота (JP)

(73) Патентообладатель(и):

КЭНОН КАБУСИКИ КАЙСЯ (JP)

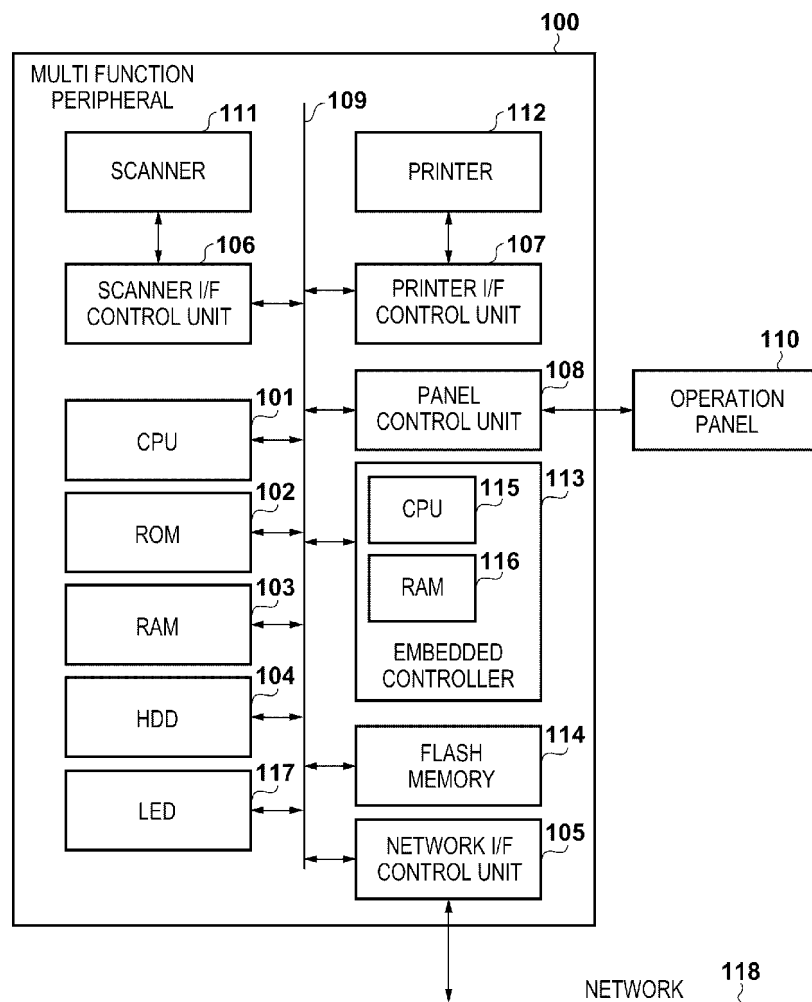
(56) Список документов, цитированных в отчете
о поиске: US 2017/0103209 A1, 13.04.2017. WO
2012/148422 A1, 01.11.2012. US 2006/0161784 A1,
20.07.2006. US 2007/0136807 A1, 14.07.2007. RU
2595924 C2, 27.08.2016.

(54) УСТРОЙСТВО ОБРАБОТКИ ИНФОРМАЦИИ, СПОСОБ ДЛЯ ЕГО УПРАВЛЕНИЯ И НОСИТЕЛЬ ХРАНЕНИЯ ДАННЫХ

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в обеспечении реализации механизма для обнаружения изменения системы во время активации. Устройство обработки информации последовательно активирует множество модулей после активации загрузочной программы. Каждый модуль использует верифицирующую информацию для верификации подписи модуля,

который должен активироваться следующим, чтобы обнаруживать изменение модуля, который должен активироваться следующим, и активирует модуль, который должен активироваться следующим, в случае если верификация подписи успешно выполняется, при этом каждый модуль хранит заранее верифицирующую информацию и собственную подпись. 3 н. и 6 з.п. ф-лы, 11 ил.



Фиг. 1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 17/00 (2020.02); *G06F 21/57* (2020.02); *G06F 21/572* (2020.02); *G06F 21/575* (2020.02); *H04L 9/3247* (2020.02)

(21)(22) Application: **2018136581, 17.10.2018**

(24) Effective date for property rights:
17.10.2018

Registration date:
23.04.2020

Priority:

(30) Convention priority:
18.10.2017 JP 2017-201956

(43) Application published: **17.04.2020 Bull. № 11**(45) Date of publication: **23.04.2020 Bull. № 12**

Mail address:

**129090, Moskva, ul. Bolshaya Spasskaya, d. 25,
stroenie 3, OOO "Yuridicheskaya firma
Gorodisskij i Partnery"**

(72) Inventor(s):

SHIMIZU, Shota (JP)

(73) Proprietor(s):

CANON KABUSHIKI KAISHA (JP)(54) **INFORMATION PROCESSING DEVICE, METHOD FOR CONTROL THEREOF AND DATA STORAGE MEDIUM**

(57) Abstract:

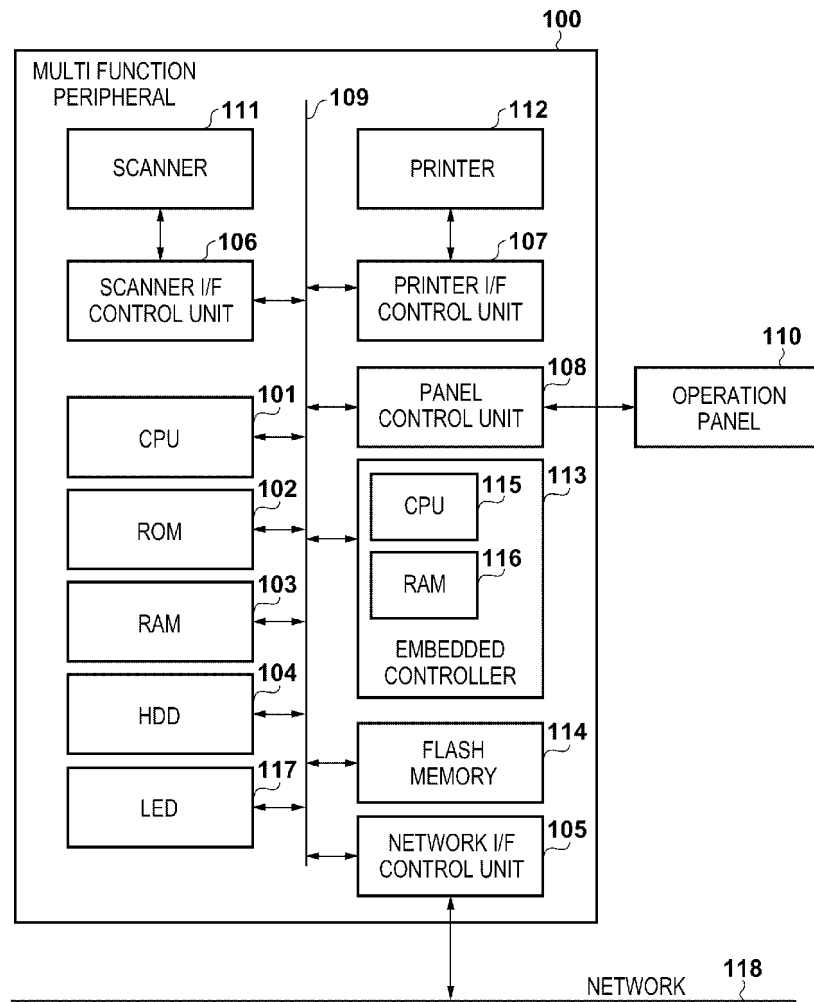
FIELD: computer equipment.

SUBSTANCE: invention relates to the computer equipment. Information processing device sequentially activates a plurality of modules after activation of the loading program. Each module uses verification information for verification of module signature, which must be activated next to detect change of module to be activated next, and activates a module to be activated

next, in case signature verification is successfully performed, wherein each module stores pre-verification information and own signature.

EFFECT: technical result consists in providing a mechanism for detecting change of system during activation.

9 cl, 11 dwg



Фиг. 1

Область техники, к которой относится изобретение

[0001] Настоящее изобретение относится к устройству обработки информации, к способу для его управления и к носителю хранения данных.

Уровень техники

5 [0002] Относительно уязвимостей программного обеспечения в компьютерной системе, атаки, которые изменяют программное обеспечение и используют компьютеры, являются проблемой. В качестве противодействия таким атакам, непатентный документ ("About TPM", [онлайн-версия], [поиск от 12 октября 2017 года], Insight International Inc., Интернет-адрес: <http://www.insight-intl.com/jigyousystem/about_TPM/about_TPM20.html>) предлагает TPM (доверенный платформенный модуль), который нацелен на укрепление безопасности в вычислительном окружении в компьютерной системе (платформе), для TCG (исследовательской группы в области доверенных вычислений). В TPM, хэш-значения для BIOS/UEFI и начального загрузчика последовательно получают из CRTM (центрального корня доверия для измерения) в периоды времени, когда они активируются, и они активируются при распространении PCR TPM один за другим. Кроме того, в TPM, обнаружение изменения выполняется посредством, после завершения активации, определения того, совпадают или нет значения PCR в TPM и корректные ответные значения. Такая защищенная активация упоминается как доверенная загрузка (защищенная загрузка). Следует отметить, что поскольку CRTM или TPM представляют собой конфигурации, которые не могут перезаписываться, атака, которая изменяет фрагмент для выполнения обработки обнаружения изменения, чтобы избежать ее, теоретически является невозможной.

[0003] Тем не менее, имеется проблема, как изложено ниже, в вышеприведенной традиционной технологии. Например, в конфигурации, которая не имеет TPM, к примеру, в недорогом устройстве, невозможно реализовывать доверенную загрузку. В случае такой конфигурации, обработка верификации для изменения выполняется после того, как активация приложения для выполнения этой обработки завершается. Вследствие этого, даже если изменение выполнено, невозможно сразу обнаруживать изменение, поскольку изменение обнаруживается после определенного времени работы BIOS или ОС, что обеспечивает запас, например, для выполнения приложения, которое изменяется со злым умыслом.

Сущность изобретения

[0004] Настоящее изобретение обеспечивает реализацию механизма для обнаружения изменения системы во время активации, даже в конфигурации, которая не имеет TPM.

35 [0005] Один аспект настоящего изобретения предоставляет устройство обработки информации, выполненное с возможностью последовательно активировать множество модулей после активации загрузочной программы посредством встроенного контроллера, причем каждый модуль содержит: блок обнаружения, который использует верифицирующую информацию для верификации подписи модуля, который должен активироваться следующим, чтобы обнаруживать изменение модуля, который должен активироваться следующим, и блок активации, который активирует модуль, который должен активироваться следующим, в случае если верификация подписи посредством блока обнаружения успешно выполняется, при этом каждый модуль хранит заранее верифицирующую информацию и собственную подпись.

45 [0006] Другой аспект настоящего изобретения предоставляет способ управления устройством обработки информации, выполненным с возможностью последовательно активировать множество модулей после активации загрузочной программы посредством встроенного контроллера, при этом:

- каждый модуль включает в себя: использование верифицирующей информации для верификации подписи модуля, который должен активироваться следующим, чтобы обнаруживать изменение модуля, который должен активироваться следующим, и активацию модуля, который должен активироваться следующим, когда верификация

5 подписи успешно выполняется.

[0007] Еще один другой аспект настоящего изобретения предоставляет машиночитаемый носитель хранения данных, сохраняющий компьютерную программу для инструктирования компьютеру выполнять каждый этап способа управления устройством обработки информации, выполненным с возможностью последовательно

10 активировать множество модулей после активации загрузочной программы посредством встроенного контроллера, при этом каждый модуль включает в себя: использование верифицирующей информации для верификации подписи модуля, который должен активироваться следующим, чтобы обнаруживать изменение модуля, который должен активироваться следующим, и активацию модуля, который должен активироваться

15 следующим, когда верификация подписи успешно выполняется.

[0008] Дополнительные признаки настоящего изобретения должны быть очевидными из нижеприведенного описания примерных вариантов осуществления со ссылкой на прилагаемые чертежи.

Краткое описание чертежей

20 [0009] Фиг. 1 является схемой аппаратной конфигурации многофункционального периферийного устройства согласно варианту осуществления.

[0010] Фиг. 2А и фиг. 2В являются схемами программной конфигурации многофункционального периферийного устройства согласно варианту осуществления.

[0011] Фиг. 3А-3Е являются характерными схемами, которые иллюстрируют операции

25 во время активации, согласно варианту осуществления.

[0012] Фиг. 4 является блок-схемой последовательности операций способа для описания процедуры обработки согласно варианту осуществления.

[0013] Фиг. 5 является блок-схемой последовательности операций способа для описания процедуры обработки согласно варианту осуществления.

30 [0014] Фиг. 6 является блок-схемой последовательности операций способа для описания процедуры обработки согласно варианту осуществления.

Подробное описание вариантов осуществления

[0015] Ниже подробно описываются предпочтительные варианты осуществления настоящего изобретения со ссылкой на чертежи. Следует отметить, что относительное

35 размещение компонентов, численные выражения и числовые значения, приводимые в этих вариантах осуществления, не ограничивают объем настоящего изобретения, если прямо не указано иное.

[0016] *Первый вариант осуществления*

Ниже приводится описание относительно первого варианта осуществления

40 настоящего изобретения, со ссылкой на прилагаемые чертежи. Следует отметить, что нижеприведенные варианты осуществления не имеют намерение ограничивать объем формулы изобретения настоящего изобретения, и что не все комбинации признаков, которые описываются согласно нижеприведенным вариантам осуществления, обязательно требуются относительно средств для того, чтобы разрешать проблемы

45 согласно настоящему изобретению. Помимо этого, описание приводится посредством рассмотрения многофункционального периферийного устройства (цифрового многофункционального периферийного устройства/MFP) в качестве примера устройства обработки информации согласно варианту осуществления. Тем не менее, объем

применения не ограничен многофункциональными периферийными устройствами, и возможно применение к любому устройству обработки информации.

[0017] Аппаратная конфигурация устройства обработки информации

Во-первых, со ссылкой на фиг. 1, приводится описание для аппаратной конфигурации многофункционального периферийного устройства 100, которое представляет собой устройство обработки информации согласно настоящему варианту осуществления, и встроенного контроллера 113. Многофункциональное периферийное устройство 100 содержит CPU 101, ROM 102 (постоянное запоминающее устройство), RAM 103 (оперативное запоминающее устройство), HDD 104 (жесткий диск), блок 105 управления сетевым интерфейсом, блок 106 управления интерфейсом сканера, блок 107 управления интерфейсом принтера, блок 108 управления панелью, сканер 111, принтер 112, встроенный контроллер 113, флэш-память 114 и светодиод 117. Помимо этого, встроенный контроллер 113 содержит CPU 115 и RAM 116.

[0018] CPU 101 выполняет программу многофункционального периферийного устройства 100, чтобы всесторонне управлять устройством в целом. ROM 102 представляет собой постоянное запоминающее устройство и сохраняет BIOS (базовую систему ввода-вывода) многофункционального периферийного устройства 100, фиксированные параметры и т.п. RAM 103 представляет собой оперативное запоминающее устройство и используется, например, для того, чтобы сохранять программу или временные данные, когда CPU 101 управляет многофункциональным периферийным устройством 100. HDD 104 представляет собой жесткий диск и сохраняет некоторые приложения и различные фрагменты данных. Флэш-память 114 сохраняет различные модули, такие как загрузчик, ядро и приложения.

[0019] CPU 115 встроенного контроллера 113 выполняет программу встроенного контроллера 113, чтобы выполнять определенное управление в многофункциональном периферийном устройстве 100. RAM 116 представляет собой оперативное запоминающее устройство и используется, например, для того, чтобы сохранять программу или временные данные, когда CPU 115 управляет многофункциональным периферийным устройством 100. Относительно встроенного контроллера 113, многофункциональное периферийное устройство 100 содержит главный контроллер для всестороннего управления системой. Главный контроллер сконфигурирован посредством включения, по меньшей мере, CPU 101, ROM 102 и RAM 103.

[0020] Блок 105 управления сетевым интерфейсом управляет передачей и приемом данных с сетью 118. Блок 106 управления интерфейсом сканера выполняет управление для считывания оригинала посредством сканера 111. Блок 107 управления интерфейсом принтера управляет обработкой печати и т.п. посредством принтера 112. Блок 108 управления панелью управляет функциональной панелью 110, которая представляет собой сенсорную панель, отображает различные фрагменты информации и управляет инструкцией, вводимой от пользователя. Шина 109 соединяет CPU 101, ROM 102, RAM 103, HDD 104, блок 105 управления сетевым интерфейсом, блок 106 управления интерфейсом сканера и блок 107 управления интерфейсом принтера между собой. Кроме того, шина 109 также соединяет блок 108 управления панелью, встроенный контроллер 113 и флэш-память 114 между собой. Управляющий сигнал из CPU 101 или сигналы данных между устройствами передаются и принимаются через шину 109. Светодиод 117 подсвечивается по мере необходимости и используется для того, чтобы передавать наружу аномальность в программном обеспечении или аппаратных средствах.

[0021] Программная конфигурация устройства обработки информации

Далее, со ссылкой на фиг. 2А, приводится описание для программных модулей,

хранимых посредством многофункционального периферийного устройства 100 согласно настоящему варианту осуществления. Многофункциональное периферийное устройство 100 включает в себя загрузочную программу 209 во встроенном контроллере 113, в качестве программного модуля. Кроме того, многофункциональное периферийное устройство 100 включает в себя BIOS 210, загрузчик 211, ядро 212, собственные программы 213, программу 214 по технологии Java (зарегистрированная торговая марка), блок 203 управления UI и блок 207 управления связью.

[0022] Блок 207 управления связью управляет блоком 105 управления сетевым интерфейсом, который соединен с сетью 118, который выполняет передачу и прием данных с внешним блоком через сеть 118. Блок 203 управления UI принимает ввод в функциональную панель 110 через блок 108 управления панелью и выполняет обработку в соответствии с вводом или экранным выводом в функциональную панель 110.

[0023] Загрузочная программа 209 представляет собой программу, которая выполняется посредством CPU 115 встроенного контроллера 113, когда источник мощности многофункционального периферийного устройства 100 включается, и помимо выполнения обработки, связанной с активацией, включает в себя блок 201 обнаружения изменения BIOS для обнаружения изменения BIOS. BIOS 210 представляет собой программу, которая выполняется посредством CPU 101 после выполнения загрузочной программы 209, и помимо выполнения обработки, связанной с активацией, включает в себя блок 202 обнаружения изменения загрузчика для обнаружения изменения загрузчика 211.

[0024] Загрузчик 211 представляет собой программу, выполняемую посредством CPU 101 после того, как обработка для BIOS 210 завершается, и помимо выполнения обработки, связанной с активацией, имеет блок 204 обнаружения изменения ядра для обнаружения изменения ядра. Ядро 212 представляет собой программу, выполняемую посредством CPU 101 после того, как обработка для загрузчика 211 завершается, и помимо выполнения обработки, связанной с активацией, имеет блок 205 обнаружения изменения собственных программ для обнаружения изменения собственных программ 213.

[0025] Собственные программы 213 представляют собой программы, которые выполняются посредством CPU 101 и содержат множество программ для предоставления различных функций, совместно с программой 214 по технологии Java (зарегистрированная торговая марка) многофункционального периферийного устройства 100. Например, собственные программы 213 включают в себя загрузочную программу либо программу для управления блоком 106 управления интерфейсом сканера или блоком 107 управления интерфейсом принтера. Загрузочная программа вызывается из собственных программ посредством ядра 212, и обработка активации выполняется. Помимо этого, собственные программы 213 имеют блок обнаружения изменения Java-программ для обнаружения изменения Java-программы в качестве одной из этих программ.

[0026] Java-программа 214 представляет собой программу, которая выполняется посредством CPU 101, и представляет собой программу, которая предоставляет различные функции, совместно с собственными программами 213 многофункционального периферийного устройства 100. Например, Java-программа 214 представляет собой программу для отображения экрана на функциональной панели 110.

[0027] *Процедура активации*

Ниже приводится описание относительно процедуры для активации

многофункционального периферийного устройства 100, со ссылкой на фиг. 3А и фиг. 3В. Фиг. 3А иллюстрирует порядок, в котором многофункциональное периферийное устройство 100 активируется без выполнения обнаружения изменения. Загрузочная программа 209 активирует BIOS 210, BIOS 210 активирует загрузчик 211, загрузчик 211 активирует ядро 212, и ядро 212 активирует загрузочную программу из собственных программ 213. Java-программа 214 активируется посредством загрузочной программы, и затем функции многофункционального периферийного устройства 100 предоставляются посредством взаимодействия собственных программ 213 и Java-программы 214. Управление активацией для каждого модуля выполняется в предварительно определенном порядке таким способом, и обработка активации последующего модуля выполняется, когда завершается активация предыдущего модуля.

[0028] Фиг. 3В иллюстрирует порядок, в котором многофункциональное периферийное устройство 100 активируется при выполнении обнаружения изменения. Как показано на чертеже, активация выполняется при выполнении обнаружения изменения в порядке загрузочной программы 209, затем BIOS 210, загрузчика 211, ядра 212, собственных программ 213 и Java-программы 214. Обнаружение изменения модуля, который должен активироваться, выполняется посредством модуля, активированного непосредственно предыдущим. Например, обнаружение изменения BIOS 210 выполняется посредством загрузочной программы 209. Помимо этого, фиг. 3В представляет местоположения хранения для каждой программы, цифровые подписи (в дальнейшем называемые "подписями") и местоположения хранения для открытых ключей (верифицирующей информации) для верификации подписей.

[0029] Ниже предполагается, что загрузочная программа 209 и BIOS 210 сохраняются в ROM 102, а загрузчик 211, ядро 212 и собственные программы 213 (первая программа) сохраняются во флэш-памяти 114. Кроме того, предполагается, что Java-программа 214 (вторая программа) сохраняется на HDD 104.

[0030] Открытый ключ 300 верификации BIOS-подписи сохраняется в загрузочной программе 209, а BIOS-подпись 302 и открытый ключ 303 верификации загрузчика сохраняются в BIOS 210. Подпись 304 загрузчика и открытый ключ 305 верификации ядра сохраняются в загрузчике 211. Помимо этого, подпись 306 ядра и открытый ключ 307 верификации собственной программы сохраняются в ядре 212, а подпись 309 собственной программы и открытый ключ 308 верификации Java-программы сохраняются в собственных программах 213. Кроме того, подпись 310 Java-программы сохраняется в Java-программе 214. Желательно, чтобы эти открытые ключи и подписи назначались заранее программам до заводской поставки многофункционального периферийного устройства 100. В многофункциональном периферийном устройстве 100 согласно настоящему варианту осуществления, соответствующие блоки обнаружения для ссылок с номерами 201, 202 и 204-206 выполняют обнаружение изменения посредством верификации соответствующей программы (соответствующего модуля), которая должна активироваться следующей, и активации следующей программы, если не возникает проблем.

[0031] Процедура обработки

Далее, со ссылкой на фиг. 4, приводится описание для процедуры для обработки во время активации многофункционального периферийного устройства 100 согласно настоящему варианту осуществления. Когда источник мощности для многофункционального периферийного устройства 100 вводится, загрузочная программа 209 считывается из ROM 102 в RAM 116, и она выполняется посредством CPU 115.

[0032] На этапе S401, блок 201 обнаружения изменения BIOS, включенный в

загрузочную программу 209, выполняет верификацию подписей для BIOS и определяют то, успешно выполнена или нет верификация подписи. В частности, блок 201 обнаружения изменения BIOS считывает BIOS 210, открытый ключ 303 верификации загрузчика и BIOS-подпись 302 из флэш-памяти 114 в RAM 116. Кроме того, блок 201 обнаружения изменения BIOS использует открытый ключ 300 верификации BIOS для того, чтобы выполнять верификацию BIOS-подписи 302 и определять то, успешно выполнена или нет верификация. Когда верификация подписи завершается неудачно, обработка переходит к этапу S403, и блок 201 обнаружения изменения BIOS инструктирует светодиоду 117 подсвечиваться, и завершает обработку. Между тем, если верификация подписи успешно выполняется, блок 201 обнаружения изменения BIOS прикладывает ток к CPU 101, и обработка загрузочной программы завершается. После этого, блок-схема последовательности операций способа переходит к обработке для этапа S402 и далее, которая выполняется посредством CPU 101.

[0033] Когда ток прикладывается, на этапе S402, CPU 101 считывает BIOS 210 и открытый ключ 303 верификации загрузчика из флэш-памяти 114 в RAM 103 и активирует BIOS 210. Последующая обработка полностью описывается в качестве обработки, которая обрабатывается посредством CPU 101.

[0034] Когда BIOS 210 активируется, обработка переходит к этапу S404. На этапе S404, BIOS 210 выполняет различную обработку инициализации, и блок 202 обнаружения изменения загрузчика, включенный в BIOS 210, считывает загрузчик 211, открытый ключ 305 верификации ядра и подпись 304 загрузчика из флэш-памяти 114 в RAM 103. Кроме того, блок 202 обнаружения изменения загрузчика использует открытый ключ 305 верификации загрузчика для того, чтобы выполнять верификацию подписи 304 загрузчика и определять то, успешно выполнена или нет верификация. Когда верификация подписи завершается неудачно, обработка переходит к этапу S412, и блок 202 обнаружения изменения загрузчика отображает сообщение об ошибке на функциональной панели 110, и обработка завершается. Между тем, когда верификация подписи успешно выполняется, блок 204 обнаружения изменения загрузчика завершает обработку, переходит к этапу S405, и BIOS 210 активирует загрузчик 211, который считан в RAM 103.

[0035] Когда загрузчик 211 активируется, обработка переходит к этапу S406. На этапе S406, загрузчик 211 выполняет различную обработку инициализации, и блок 204 обнаружения изменения ядра, включенный в загрузчик 211, считывает ядро 212, открытый ключ 307 верификации собственной программы и подпись 306 ядра из флэш-памяти 114 в RAM 103. Кроме того, на этапе S406, блок 204 обнаружения изменения ядра использования открытый ключ 305 верификации ядра, чтобы выполнять верификацию подписи 306 ядра и определять то, успешно выполнена или нет верификация. Когда верификация подписи завершается неудачно, обработка переходит к этапу S412, и блок 204 обнаружения изменения ядра отображает сообщение об ошибке на функциональной панели 110, и обработка завершается. Между тем, когда верификация подписи успешно выполняется, блок 204 обнаружения изменения ядра завершает обработку, переходит к этапу S407, и загрузчик 211 активирует ядро 212, которое считано в RAM 103.

[0036] Когда ядро 212 активируется, обработка переходит к этапу S408. На этапе S408, ядро 212 выполняет различную обработку инициализации. Кроме того, блок 205 обнаружения изменения программ, включенный в ядро 212, считывает собственные программы 213, открытый ключ 308 верификации Java-программы и подпись 309 собственной программы из флэш-памяти 114 в RAM 103. На этапе S408, блок 205

обнаружения изменения программ использует открытый ключ 307 верификации собственной программы для того, чтобы выполнять верификацию подписи 309 собственной программы и определять то, успешно выполнена или нет верификация. Когда верификация подписи завершается неудачно, на этапе S412, блок 205 обнаружения
 5 изменения программ отображает сообщение об ошибке на функциональной панели 110, и обработка завершается. Между тем, когда верификация подписи успешно выполняется, блок 205 обнаружения изменения программ завершает обработку и активирует собственные программы 213 на этапе S409.

[0037] Когда, из числа собственных программ 213, блок 206 обнаружения изменения
 10 Java-программ, который предназначен для выполнения обработки для обнаружения изменения, активируется, обработка переходит к этапу S410. На этапе S410, блок 206 обнаружения изменения Java-программ считывает Java-программу 214 и подпись 310 Java-программы из HDD 104 в RAM 103. На этапе S410, блок 206 обнаружения изменения Java-программ использует открытый ключ 308 верификации Java-программы для того,
 15 чтобы выполнять верификацию подписи 310 Java-программы и определять то, успешно выполнена или нет верификация. Когда верификация подписи завершается неудачно, обработка переходит к этапу S412, и блок 206 обнаружения изменения Java-программ отображает сообщение об ошибке на функциональной панели 110, и обработка завершается. Между тем, когда верификация подписи успешно выполняется, блок 206
 20 обнаружения изменения Java-программ завершает обработку, и собственные программы 213 активируют Java-программу 214 на этапе S411.

[0038] Как описано выше, устройство обработки информации согласно настоящему варианту осуществления последовательно активирует множество модулей после активации загрузочной программы. Каждый модуль использует верифицирующую
 25 информацию для верификации подписи модуля, который должен активироваться следующим, чтобы обнаруживать изменение модуля, который должен активироваться следующим, и активирует модуль, который должен активироваться следующим, когда верификация подписи успешно выполняется. Кроме того, каждый модуль хранит заранее верифицирующую информацию и собственную подпись. В силу этого, настоящее
 30 устройство обработки информации, даже без TPM-конфигурации, может обнаруживать изменение независимо от того, что из BIOS 210 загрузчика 211, ядра 212, собственных программ 213 или Java-программы 214 изменено. Помимо этого, поскольку обнаружение изменения выполняется непосредственно перед активацией каждой программы, и активация сразу прекращается, если изменение найдено, можно гарантировать, что
 35 многофункциональное периферийное устройство 100 работает только в защищенном состоянии в любой момент времени, и измененная программа не работает даже на один момент. Помимо этого, в соответствии с другим аспектом, настоящее устройство обработки информации может предоставлять более надежный механизм для обнаружения изменения системы.

40 [0039] Второй вариант осуществления

Ниже приводится описание для второго варианта осуществления настоящего изобретения. В первом варианте осуществления, описанном выше, предусмотрена конфигурация, в которой HDD 104 сохраняет Java-программы. Поскольку HDD 104 представляет собой область, которая используется для хранения данных, он может
 45 считываться или записываться и имеет наибольшую вероятность изменения. В первом варианте осуществления, описанном выше, имеется проблема в том, что доступность снижается, поскольку активация многофункционального периферийного устройства 100 прекращается, когда существует изменение. Соответственно, в настоящем варианте

осуществления, приводится описание относительно способа, при использовании которого доступность не снижается, даже если Java-программа, сохраненная на HDD 104, изменяется.

[0040] *Процедура активации*

5 Далее, со ссылкой на фиг. 3С, приводится описание для каждой программы, подписи, местоположения хранения для открытого ключа и порядка активации в варианте осуществления. Как проиллюстрировано посредством фиг. 3С, процедура для активации каждого модуля является аналогичной фиг. 3В, которая описывается в первом варианте осуществления, описанном выше. Тем не менее, в настоящем варианте осуществления, в отличие от первого варианта осуществления, описанного выше, Java-программа также сохраняется во флэш-памяти 114, помимо HDD 104. Кроме того, подпись 330, которая является идентичной подписи 310 Java-программы 214, сохраняется во флэш-памяти 114 вместе с Java-программой 331. При условии, что подпись 330 не изменяется, она является идентичной подписи 310, сохраненной на HDD 104.

15 [0041] *Процедура обработки*

Далее, со ссылкой на фиг. 5, приводится описание для процедуры для обработки во время активации многофункционального периферийного устройства 100 согласно настоящему варианту осуществления. Поскольку этапы S401-S407, этап S409, этап S410 и этап S412 имеют обработку, аналогичную обработке в первом варианте осуществления, описанном выше, ее описание опускается.

20 [0042] На этапе S407, когда ядро 212 активируется, выполняется различная обработка инициализации, и обработка переходит к этапу S501. Кроме того, блок 205 обнаружения изменения программ, включенный в ядро 212, считывает собственные программы 213 и Java-программу 214 из флэш-памяти 114 в RAM 103. Кроме того, блок 205 обнаружения изменения программ считывает открытый ключ 308 верификации Java-программы, подпись 309 собственной программы и подпись 330 Java-программы 331 из флэш-памяти 114 в RAM 103.

[0043] Далее, на этапе S501, блок 205 обнаружения изменения программ использует открытый ключ 307 верификации собственной программы для того, чтобы выполнять верификацию подписи 309 собственных программ 213. Кроме того, блок 205 обнаружения изменения программ использует открытый ключ 308 верификации Java-программы для того, чтобы выполнять верификацию подписи 330 Java-программы 331 и определять то, успешно выполнены или нет обе верификации подписей. Когда обе верификации подписей успешно выполняются, обработка переходит к этапу S409. 35 Между тем, когда любая верификация подписей завершается неудачно, обработка переходит к этапу S412.

[0044] Когда обработка этапа S409 завершается, и на этапе S410 определяется то, что верификация подписей Java-программы 214 завершена неудачно, обработка переходит к этапу S502. На этапе S502, блок 206 обнаружения изменения Java-программ удаляет Java-программу 214 из HDD 104 и выполняет обработку повторного развертывания для записи Java-программы 331, сохраненной во флэш-памяти 114, и ее подписи 330 на HDD 104. Когда обработка повторного развертывания завершается, блок 206 обнаружения изменения Java-программ завершает обработку, обработка переходит к этапу S411, и Java-программа 214 активируется.

45 [0045] Как описано выше, на основе настоящего варианта осуществления, Java-программы хранятся во флэш-памяти 114 в дополнение к HDD 104. Вследствие этого, даже если Java-программа HDD 104, которая имеет высокую вероятность изменения, изменяется, можно не допускать снижения доступности, поскольку изменение может

быть восстановлено посредством повторного развертывания Java-программы 331, которая находится флэш-памяти 114.

[0046] *Третий вариант осуществления*

Ниже приводится пояснение для третьего варианта осуществления настоящего изобретения. Как проиллюстрировано посредством фиг. 3D, предусмотрен случай, в котором рассматриваемая конфигурация представляет собой конфигурацию, в которой многофункциональное периферийное устройство 100 хранит множество ядер и программ, и ядра и программы, которые должны активироваться, заменяются посредством загрузчика. При такой конфигурации, в случае попытки активировать ядро В 220, которое не представляет собой ядро 212 согласно конфигурации первого или второго вариантов осуществления, описанных выше, имеется проблема в том, что поскольку отсутствует подпись для ядра В 220, она должна обнаруживаться как измененная, несмотря на отсутствие изменения, и не активируется. Соответственно, в настоящем варианте осуществления, приводится описание относительно способа для выполнения обнаружения изменения и активации даже в конфигурации, в которой хранятся различные ядро и различная программа.

[0047] *Программная конфигурация*

Во-первых, со ссылкой на фиг. 2В, приводится описание для примера конфигурации программных модулей, хранимых посредством многофункционального периферийного устройства 100 согласно настоящему варианту осуществления. Поскольку ссылки с номерами 201-214 являются идентичными ссылкам с номерами на фиг. 2А, их описание опускается.

[0048] Загрузчик 223 представляет собой программу, выполняемую посредством CPU 101 после того, как обработка для BIOS 210 завершается, и помимо выполнения обработки, связанной с активацией, имеет блок 224 обнаружения изменения ядра для обнаружения изменения ядра. Загрузчик 223 заменяет ядро, которое следует активировать в соответствии с пользовательским вводом, через функциональную панель 110.

[0049] Ядро В 220 представляет собой программу, выполняемую посредством CPU 101, которая отличается от ядра 212, выполняет обработку, связанную с активацией, и также имеет блок 221 обнаружения изменения программ для выполнения обнаружения изменения собственной программы В 222. Собственная программа В 222 представляет собой программу, которая выполняется посредством CPU 101, и предоставляет функцию обновления для многофункционального периферийного устройства 100. Она вызывается посредством ядра В 220 и предоставляет функцию для обновления ядра 212, собственных программ 213 или Java-программы 214. Следует отметить, что собственная программа В 222 не ограничена функцией обновления и может представлять собой программу, которая предоставляет другую функцию.

[0050] *Процедура активации*

Далее, со ссылкой на фиг. 3Е, приводится описание для последовательности операций обработки, в которой цель для обнаружения изменения может заменяться и активироваться в соответствии с тем, представляет собой ядро, которое следует активировать посредством загрузчика 223, ядро 212 или ядро В 220.

[0051] Предполагается, что загрузчик 223 включает в себя подпись 304 загрузчика 223, открытый ключ 305 для верификации ядра 212 и открытый ключ 340 для верификации ядра В 220. Помимо этого, подпись 341 для ядра В и открытый ключ 342 для верификации собственной программы В включены в ядро В 220, и подпись 343 для собственной программы В включена в собственную программу В 222. Желательно,

чтобы эти открытые ключи и подписи назначались заранее программам до заводской поставки многофункционального периферийного устройства 100. Таким образом, загрузчик 223 включает в себя соответствующие открытые ключи (верифицирующую информацию) для множества ядер, которые могут активироваться следующими.

5 [0052] *Процедура обработки*

Далее, со ссылкой на фиг. 6, приводится описание для процедуры для обработки во время активации многофункционального периферийного устройства 100 согласно настоящему варианту осуществления. Поскольку обработка этапов S401-S404 и этапов S406-S412 является аналогичной обработке на фиг. 4, ее описание опускается.

10 [0053] На этапе S600, когда загрузчик 223 активируется, различная обработка инициализации выполняется. Далее, на этапе S601, загрузчик 223 определяет то, выбрано или нет ядро В 220 в качестве цели активации через функциональную панель 110. Если ядро В 220 не выбрано в качестве цели активации, выполняется переход к обработке этапа S406. Между тем, когда ядро В 220 выбирается, блок 224 обнаружения изменения
15 ядра, включенный в загрузчик 223, считывает ядро В 220, открытый ключ 342 для верификации собственной программы В и подпись 341 ядра В из флэш-памяти 114 в RAM 103. На этапе S602, блок 224 обнаружения изменения ядра использует открытый ключ 342 для верификации ядра В, чтобы выполнять верификацию подписи 341 ядра В и определять то, успешно выполнена или нет верификация. Если верификация подписи
20 завершается неудачно, обработка переходит к этапу S412. Между тем, когда верификация подписи успешно выполняется, блок 221 обнаружения изменения ядра завершает обработку, и на этапе S603, загрузчик 223 активирует ядро В 220, которое считано в RAM 103.

[0054] Когда ядро В 220 активируется, выполняется различная обработка инициализации, и блок 221 обнаружения изменения программ, включенный в ядро В
25 220, считывает собственную программу В 222 и подпись 343 для собственной программы В из флэш-памяти 114 в RAM 103. На этапе S604, блок 221 обнаружения изменения программ использует открытый ключ 342 для верификации собственной программы В, чтобы выполнять верификацию подписи 343 собственной программы В и определять
30 то, успешно выполнена или нет верификация. Если верификация подписи завершается неудачно, обработка переходит к этапу S412. Между тем, когда верификация подписи успешно выполняется, блок 221 обнаружения изменения программ завершает обработку и активирует собственную программу В 222 на этапе S605. При активации, собственная программа В 222 предоставляет пользователю функцию обновления.

35 [0055] Как описано выше, на основе настоящего варианта осуществления, даже в конфигурации, в которой хранятся множество ядер и множество программ, можно выполнять активацию после выполнения обнаружения изменения для них, и можно достигать преимущества, аналогичного преимуществу первого и второго вариантов осуществления, описанных выше.

40 [0056] *Варьирование*

Настоящее изобретение не ограничено вариантами осуществления, описанными выше, и возможны различные варьирования. В первом-третьем вариантах осуществления, описанных выше, открытые ключи описаны как отличающиеся друг от друга, но они могут быть идентичными. Помимо этого, описание приведено с ROM
45 102, флэш-памятью 114 и HDD 104 в качестве местоположения хранения для соответствующих программ, но это не ограничивает местоположение хранения, и местоположение хранения может представлять собой другой носитель хранения данных. Помимо этого, местоположение хранения программы не должно обязательно

представлять собой описанное местоположение, и, например, может осуществляться конфигурация для того, чтобы сохранять загрузчик 223 в ROM 102.

[0057] *Другие варианты осуществления*

Вариант(ы) осуществления настоящего изобретения также могут реализовываться
 5 посредством компьютера системы или устройства, которое считывает и выполняет
 машиноисполняемые инструкции (например, одну или более программ), записанные
 на носителе хранения данных (который также более полно может называться
 "энергонезависимым машиночитаемым носителем хранения данных"), с тем чтобы
 10 выполнять функции одного или более вышеописанных вариантов осуществления, и/
 или которое включает в себя одну или более схем (например, специализированную
 интегральную схему (ASIC)) для выполнения функций одного или более вышеописанных
 вариантов осуществления, и посредством способа, осуществляемого посредством
 компьютера системы или устройства, например, посредством считывания и выполнения
 15 машиноисполняемых инструкций из носителя хранения данных с возможностью
 выполнять функции одного или более вышеописанных вариантов осуществления, и/
 или управления одной или более схем с возможностью выполнять функции одного или
 более вышеописанных вариантов осуществления. Компьютер может содержать один
 или более процессоров (например, центральный процессор (CPU), микропроцессор
 (MPU)) и может включать в себя сеть отдельных компьютеров или отдельных
 20 процессоров, чтобы считывать и выполнять машиноисполняемые инструкции.
 Машиноисполняемые инструкции могут предоставляться в компьютер, например, из
 сети или с носителя хранения данных. Носитель хранения данных может включать в
 себя, например, одно или более из жесткого диска, оперативного запоминающего
 устройства (RAM), постоянного запоминающего устройства (ROM), устройства хранения
 25 распределенных вычислительных систем, оптического диска (такого как компакт-диск
 (CD), универсальный цифровой диск (DVD) или Blu-Ray-диск (BD)TM), устройства флэш-
 памяти, карты памяти и т.п.

[0058] Хотя настоящее изобретение описано со ссылкой на примерные варианты
 осуществления, следует понимать, что изобретение не ограничено раскрытыми
 30 примерными вариантами осуществления. Объем прилагаемой формулы изобретения
 должен соответствовать самой широкой интерпретации, так что он включает в себе
 все такие модификации и эквивалентные структуры и функции.

(57) Формула изобретения

35 1. Устройство обработки информации, выполненное с возможностью последовательно
 активировать множество программных модулей после активации загрузочной
 программы посредством встроенного контроллера, причем каждый программный
 модуль, кроме программного модуля, который должен быть активирован последним,
 содержит:

40 блок обнаружения, который использует верифицирующую информацию для
 верификации подписи программного модуля, который должен активироваться
 следующим, чтобы обнаруживать изменение программного модуля, который должен
 активироваться следующим, и

45 блок активации, который активирует программный модуль, который должен
 активироваться следующим, в случае если верификация подписи посредством блока
 обнаружения успешно выполняется, и

при этом каждый программный модуль хранит заранее верифицирующую
 информацию и собственную подпись, и

при этом обнаружение блоком обнаружения изменения программного модуля, который должен активироваться следующим, содержащегося в программном модуле, при успешном выполнении верификации, выполняется повторно до тех пор, пока не будет активирован программный модуль, который должен быть активирован последним.

2. Устройство обработки информации по п. 1, в котором:
встроенный контроллер для выполнения загрузочной программы:
содержит блок обнаружения и блок активации, и
хранит заранее верифицирующую информацию.

3. Устройство обработки информации по п. 1, в котором блок активации прекращает активацию устройства обработки информации в случае, если верификация подписи посредством блока обнаружения завершается неудачно.

4. Устройство обработки информации по п. 1, в котором:
множество программных модулей включают в себя BIOS (базовую систему ввода-вывода), загрузчик, ядро, первую программу и вторую программу, и
загрузочная программа и BIOS сохраняются в ROM (постоянном запоминающем устройстве), загрузчик, ядро и первая программа сохраняются во флэш-памяти, и вторая программа сохраняется на HDD (жестком диске).

5. Устройство обработки информации по п. 4, в котором:
флэш-память также сохраняет вторую программу, и
в случае если верификация подписи посредством блока обнаружения для второй программы, сохраненной на HDD, завершается неудачно, блок активации программного модуля, который активирует вторую программу, повторно развертывает вторую программу, сохраненную во флэш-памяти, на HDD и активирует вторую программу.

6. Устройство обработки информации по п. 1, в котором:
блок активации заменяет и активирует программный модуль, который должен активироваться следующим, в соответствии с пользовательским вводом,
каждый программный модуль хранит заранее верифицирующую информацию для каждого из множества программных модулей, которые могут активироваться следующими, и собственную подпись.

7. Устройство обработки информации по п. 1,
дополнительно содержащее главный контроллер, который управляет каждым программным модулем,
при этом встроенный контроллер имеет запоминающее устройство и процессор, отдельные от главного контроллера.

8. Способ управления устройством обработки информации, выполненным с возможностью последовательно активировать множество программных модулей после активации загрузочной программы посредством встроенного контроллера, при этом:
каждый программный модуль, кроме программного модуля, который должен быть активирован последним, включает в себя этапы, на которых:

используют верифицирующую информацию для верификации подписи программного модуля, который должен активироваться следующим, чтобы обнаруживать изменение программного модуля, который должен активироваться следующим, и
активируют программный модуль, который должен активироваться следующим, когда верификация подписи успешно выполняется, и

при этом каждый программный модуль хранит заранее верифицирующую информацию и собственную подпись, и

при этом обнаружение блоком обнаружения изменения программного модуля, который должен активироваться следующим, содержащегося в программном модуле,

при успешном выполнении верификации, выполняется повторно до тех пор, пока не будет активирован программный модуль, который должен быть активирован последним.

9. Машиночитаемый носитель хранения данных, сохраняющий компьютерную программу для инструктирования компьютеру выполнять каждый этап способа управления устройством обработки информации, выполненным с возможностью последовательно активировать множество программных модулей после активации загрузочной программы посредством встроенного контроллера, при этом:

каждый программный модуль, кроме программного модуля, который должен быть активирован последним, включает в себя:

использование верифицирующей информации для верификации подписи программного модуля, который должен активироваться следующим, чтобы обнаруживать изменение программного модуля, который должен активироваться следующим, и

активацию программного модуля, который должен активироваться следующим, когда верификация подписи успешно выполняется, и

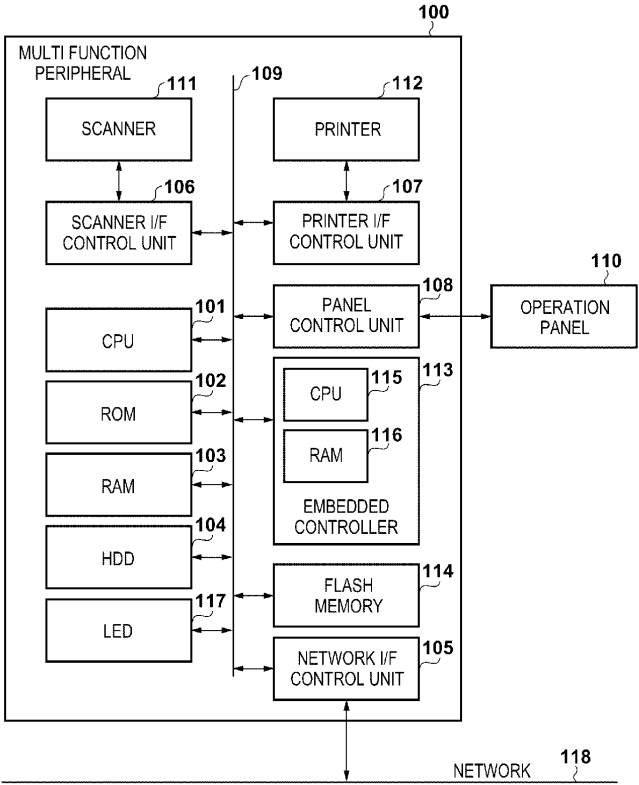
при этом каждый программный модуль хранит заранее верифицирующую информацию и собственную подпись, и

при этом обнаружение блоком обнаружения изменения программного модуля, который должен активироваться следующим, содержащегося в программном модуле, при успешном выполнении верификации, выполняется повторно до тех пор, пока не

будет активирован программный модуль, который должен быть активирован последним.

1

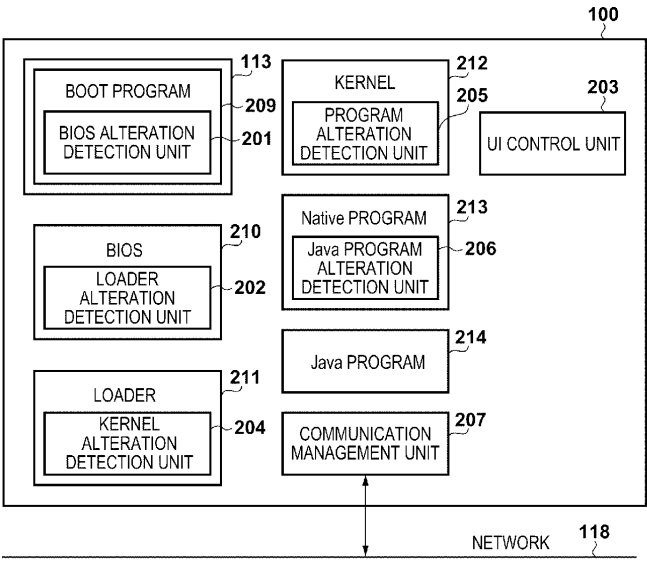
1/11



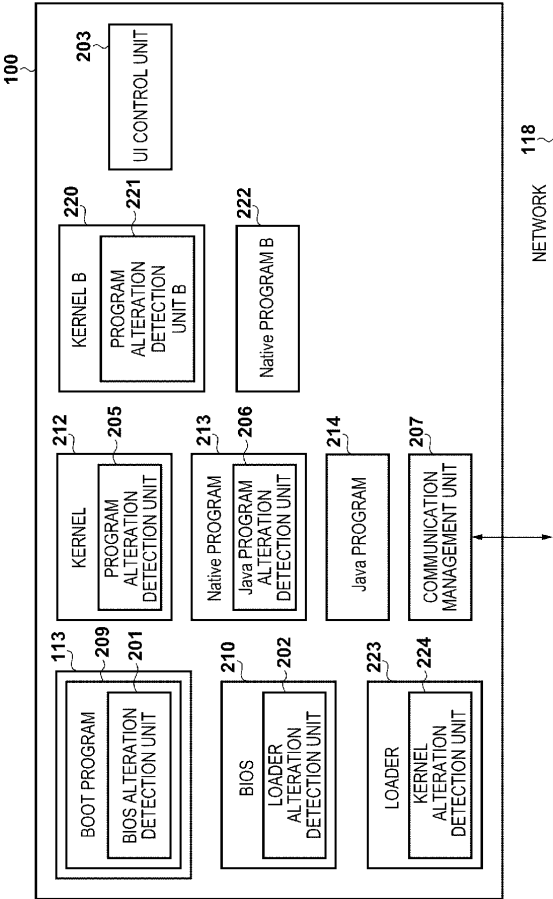
Фиг. 1

2

2/11

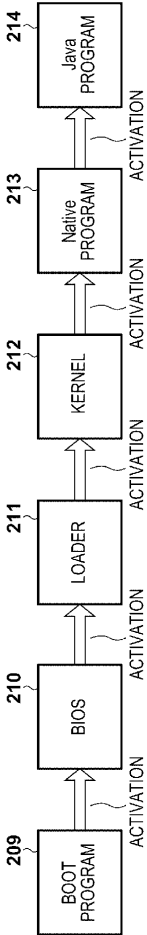


Фиг. 2А

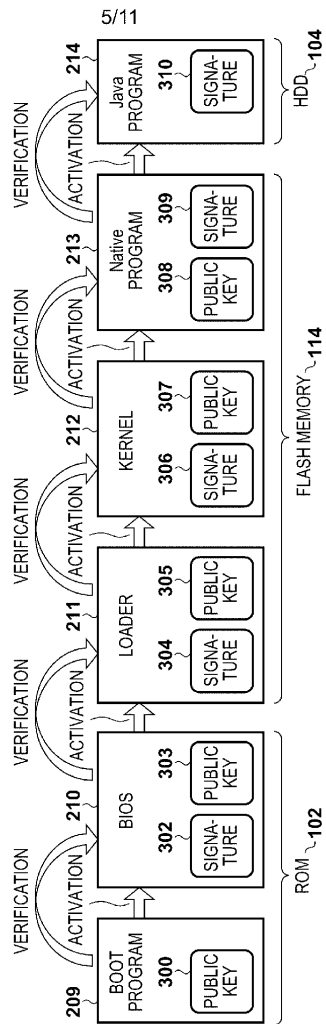


Фиг. 2В

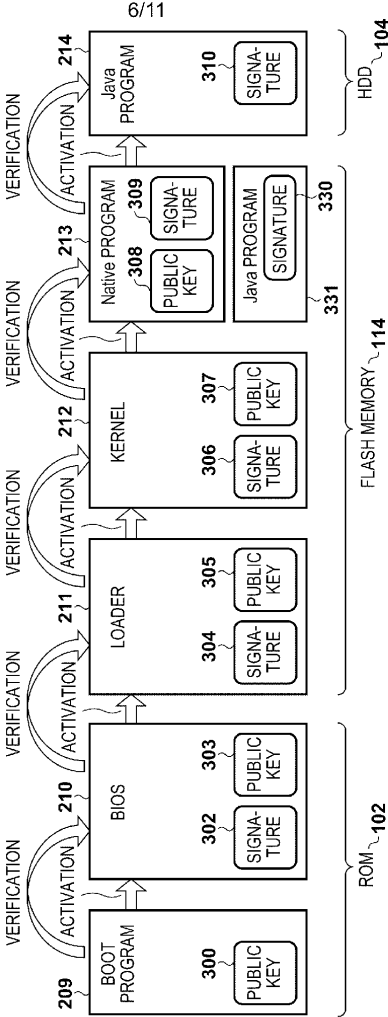
4/11



Фиг. 3А

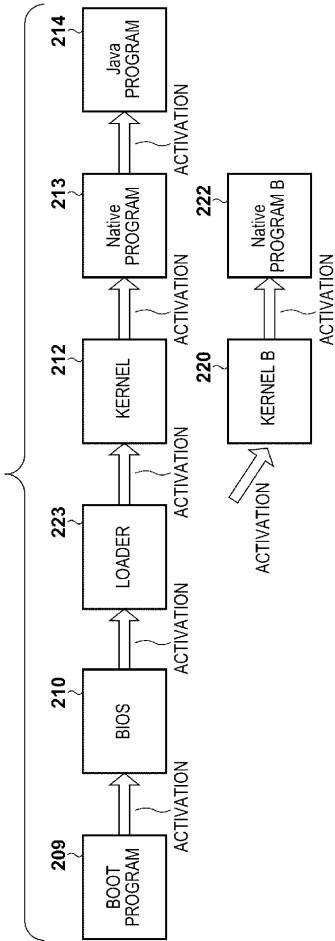


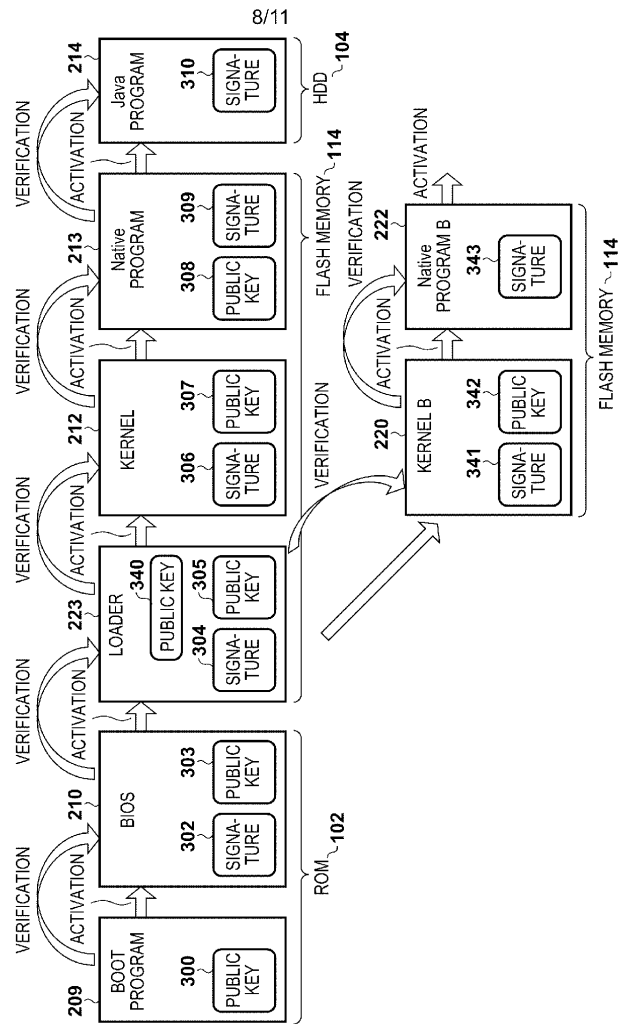
Фиг. 3В



Фиг. 3С

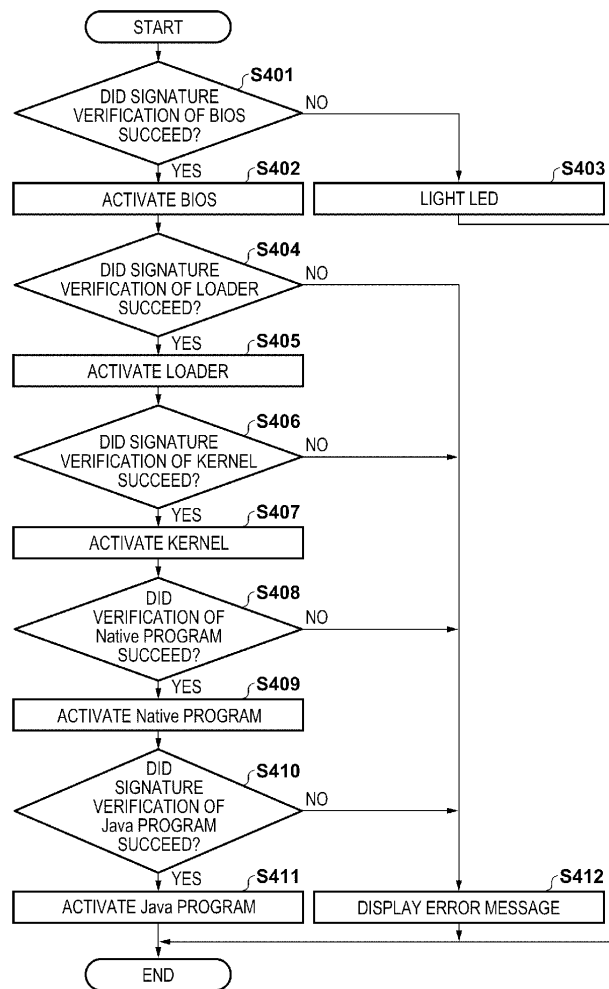
Фиг. 3D





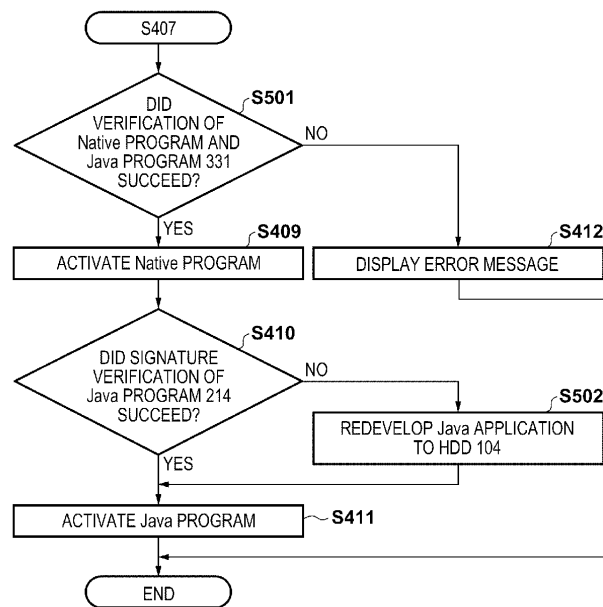
Фиг. 3Е

9/11



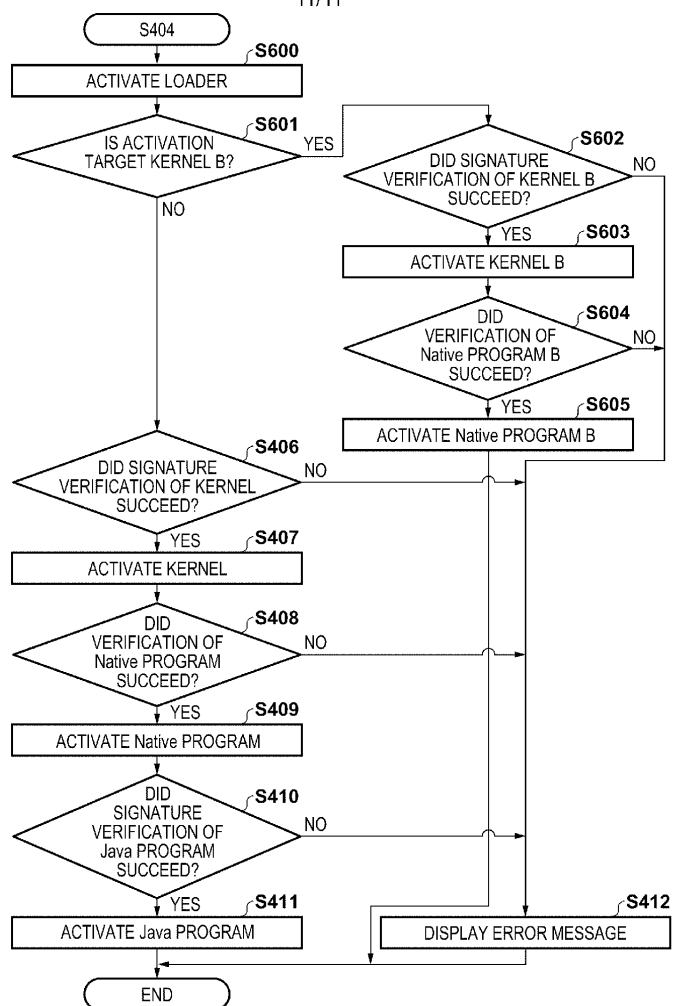
Фиг. 4

10/11



Фиг. 5

11/11



Фиг. 6