УДК 004.054

Самарин Н.Н.

Научно-исследовательский институт «Квант», г. Москва, Россия

РАЗВИТИЕ ПРОГРАММНОГО ИНСТРУМЕНТАРИЯ ОЦЕНКИ НАДЕЖНОСТИ ПРОГРАММНЫХ ПРОДУКТОВ БЕЗ ИСХОДНЫХ ТЕКСТОВ

Аннотация

В статье описывается развитие отечественного программного инструментария, позволяющего оценивать надежность, качество и безопасность программных продуктов без исходных текстов. Указаны результаты его работы. Приведены сведения о действующих нормативно-методических документах, которые потеряли свою актуальность на фоне развития информационных технологий и аппаратно-программных решений. Приведены альтернативные комплексы анализа программных продуктов, имеющие сертификаты ФСТЭК России.

Ключевые слова

Надежность, программный продукт, программный инструментарий, информационная система, системы высокой доступности, области оперативной памяти, потенциально опасные возможности, гипервизор.

Samarin N.N.

Technology Federal State Unitary Enterprise "Research Institute Kvant", Moscow, Russia

DEVELOPMENT OF SOFTWARE INSTRUMENTATION FOR ASSESSMENT OF RELIABILITY OF SOFTWARE PRODUCTS WITHOUT INITIAL TEXTS

Abstract

The article describes the development of domestic software tools, that allows you to assess the reliability, quality and security of software products without source code access. The results of this work are given. Information is provided on the current regulatory and methodological documents that have lost their relevance against the backdrop of the development of information technology and hardware-software solutions. The alternative software analysis complexes with FSTEC certificates are given.

Keywords

Reliability, software product, software tools, information system, high availability systems, main storage region, potentially dangerous, the hypervisor.

Введение

В Российской Федерации автоматизированные системы управления стратегическими объектами, тепловыми электростанциями, гидроэлектростанциями, объектами авиационного, железнодорожного, речного и морского транспорта, космической отрасли, а также системы высокой доступности (СВД), обеспечивающие суверенность государства, зачастую функционируют под управлением импортного программного обеспечения, которое может содержать дефекты, уязвимости, что гарантированно, может привести к сбоям. На территории нашего государства действует система сертификации программных продуктов (ПП) на соответствие требованиям надежности и безопасности. Уполномоченным органом является Федеральная служба по техническому и экспортному контролю (ФСТЭК России), которая на основании ведомственных нормативных документов выдает сертификаты. Однако действующие нормативные документы морально устарели [1], процедура сертификации является сложной и ресурсоемкой, требующей больших временны́х затрат. При этом в нашей стране отсутствует система классификации, систематизации и учета выявленных уязвимостей.

В литературе информацию об исследовании ПП по требованиям надёжности, устойчивости и безопасности найти очень сложно, все сводится к

поиску потенциальных уязвимостей в исходных текстах, не говоря уже о программах, исходный текст которых отсутствует, либо к математической оценке программ, аппаратно-программных средств и информационных систем (ИС) на надёжность [2-4]. При этом, стоит отметить, что наибольшую опасность представляют уязвимости, заложенные в программное обеспечение из-за невнимательности разработчиков на ранних этапах его жизненного цикла.

С ростом сложности ПП динамический анализ становится неразрешимой задачей и превращается в формальную процедуру. Многие ПП, используемые экспертами в повседневной работе, имеют списки потенциально опасных конструкций, а требования к ним руководящей документацией не выдвигаются, что делает неэффективными методы сигнатурного анализа. Не упоминается сигнатурный анализ и в требованиях к испытаниям программ ниже второго уровня контроля (т.е. программ, обрабатывающих секретную и конфиденциальную информацию) [5]. Нет механизмов выявления ошибок кодирования, связанных с переполнением буфера, вызовом функций из чужого адресного пространства, отсутствует очистка памяти и т.д. Отсутствуют требования по построению перечня маршрутов при выполнении функциональных ветвей программы, нет механизмов определения полноты покрытия кода и его достаточности при динамическом анализе.

Стоит обратить внимание, что на отечественном рынке существуют российские разработки, а некоторые из них имеют сертификат ФСТЭК России. Анализатор исходных текстов «АИСТ-С» [6], позволяет получать в автоматическом режиме информацию о структуре и ряде характеристик исследуемого программного обеспечения, а также обеспечивает проведение экспертом в интерактивном режиме операций по анализу получаемой информации с привязкой к анализируемым исходным текстам. Программный комплекс «IRIDA 2.0» [7] предназначен для проведения статического и динамического анализа потоков управления в исходных кодах программ на языке Си++ для проектов Microsoft Visual Studio.NET. Анализатор «АК-ВС 2» [7] предназначен для проведения сертификационных испытаний на отсутствие недекларированных возможностей (программных закладок) и анализа безопасности программного кода. Это единственное сертифицированное средство проведения сертификационных испытаний по 2-му и 1-му уровням контроля отсутствия недекларированных возможностей по требованиям руководящих документов.

Подходы к решению задачи оценки надежности ПП

Особый интерес представляют работы, проводимые не столько с исследованием исходных текстов, сколько с контролем обращений ПП к аппаратным средствам ИС, где они могут реализовывать скрытые функциональные возможности, которые невозможно обнаружить на этапе анализа после проведения дизассемблирования.

Введем такое понятие, как *«код с потенциально опасными возможностями».* Данным определением назовем код, реализующий одно или несколько действий конкретного вида, последствия выполнения которых могут либо непосредственно представлять угрозу целостности, достоверности и конфиденциальности данных, либо приводить к выполнению кода, представляющего данную угрозу [8]. К таким видам относится:

- передача управления в область модифицированных данных;
- самомодификация или изменение кода других программ в оперативной памяти или на внешних носителях;
- самодублирование, подмену собой других программ или перенос своих фрагментов в область оперативной или внешней памяти, не принадлежащие программе;
- сохранение информации из областей оперативной памяти, не принадлежащих программе;
- искажение, блокирование или подмену информации, являющейся результатом работы других программ;
- скрытие своего присутствия в программной среде.

Указанные потенциально опасные возможности рассматриваются для оперативной памяти ввиду того, что большая часть уязвимостей реализуется через её области. Для реализации задачи контроля был выбран гипервизор Bochs [9], который создает журнал работы и формирует список команд работы процессора вычислительной системы. Далее разработан ПИ, состоящий из двух независимых модулей [10-11]. Первый модуль обеспечивает выборку из журнала работы гипервизора только те позиции, которые содержат сведения об обращениях процессора к областям памяти по командам чтения, записи и исполнения, при этом строки содержат физический адрес области памяти, по которому можно узнать ее тип и определить имеет ли право исследуемый ПП обращаться в эту область. Фрагмент журнала гипервизора приведен на рисунке 1.

Второй модуль позволяет визуализировать процессы работы изучаемого ПП в реальном масштабе времени. Визуализация обусловлена тем, что:

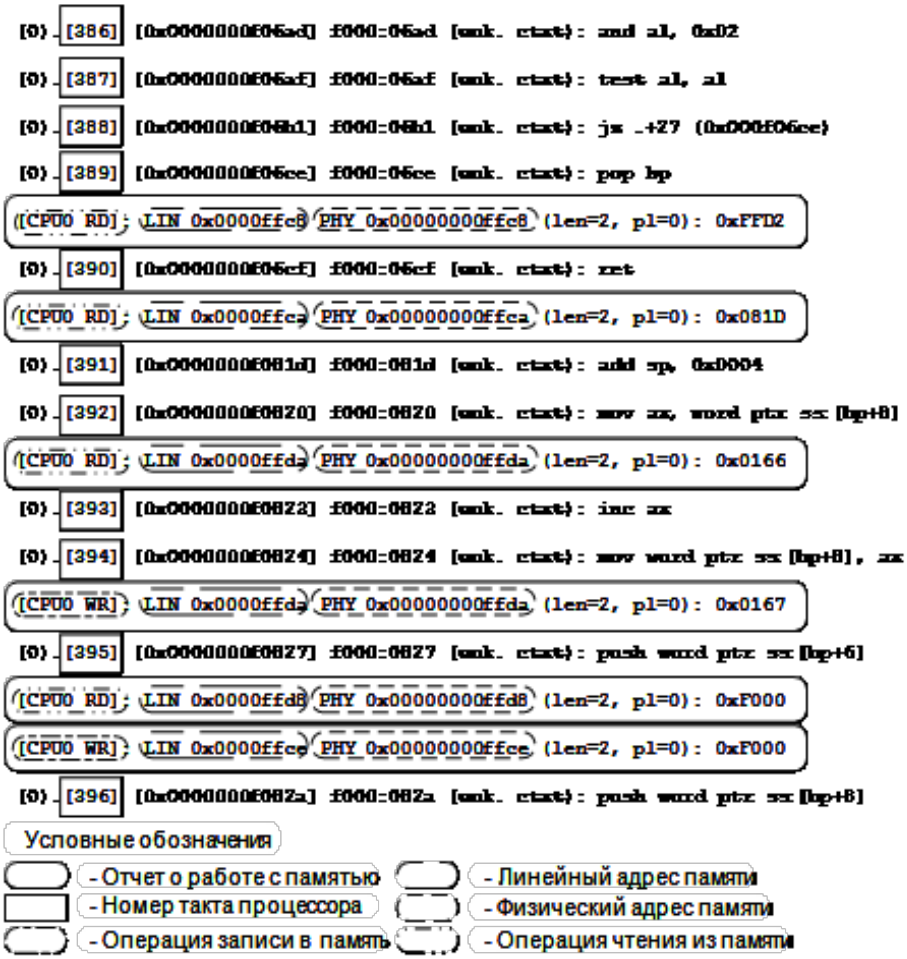


Рисунок 1. Фрагмент журнала гипервизора

1. исчезает необходимость исследования всего объема продукта, а только тех функциональных возможностей, который вызывают сомнения у эксперта;
2. наглядно показывает все области памяти ИС, которые выделены и задействованы во время работы;
3. позволяет обучать в процессе работы молодых сотрудников;
4. освобождает «дорогое» время опытных специалистов, которое нужно использовать для решения других задач;
5. позволяет привлекать к процессу исследования большее количество специалистов, смежных областей (программисты, системные архитекторы, аналитики и пр.).

Задача решалась под управлением операционной системы MS DOS 6.22, при этом результат может быть получен и под операционными системами семейства MS Windows.

Выбор данной операционной системы был сделан с целью:

- проверки выдвинутой теории о возможности контроля обращения исследуемого программного продукта к областям памяти вычислительных средств;
- разработки программного инструментария в разумные сроки без отрыва от работы коллективом до трех человек;
- выполнения экспериментов с рядом программ без исходных текстов и подтверждения их обращения к областям памяти, куда они не должны иметь доступ;
- выполнение исследований в реальном масштабе времени.

Основные инструментальные особенности

Первая рабочая версия, полученная в конце 2013 года и получившая условное наименование broute_2013, выполняла следующие функции:

- визуализация всех областей памяти ИС;
- контроль за выполнением операций чтение, запись и исполнение, а также «прорисовка» на областях памяти поадресно данные операции;
- формирование событий на визуализаторе в реальном масштабе времени;
- поадресное изменение контролируемых областей памяти;
- изменение визуализируемых областей памяти в пикселях;
- выполнение мгновенных снимков (snapshot) для дальнейшего детального

анализа;
- изменение количества выводимых строк на рабочую область визуализатора;
- изменение количества операций процессора на выводимую строку (масштабирование).

Диалоговое окно модуля визуализации приведено на рисунке 2.

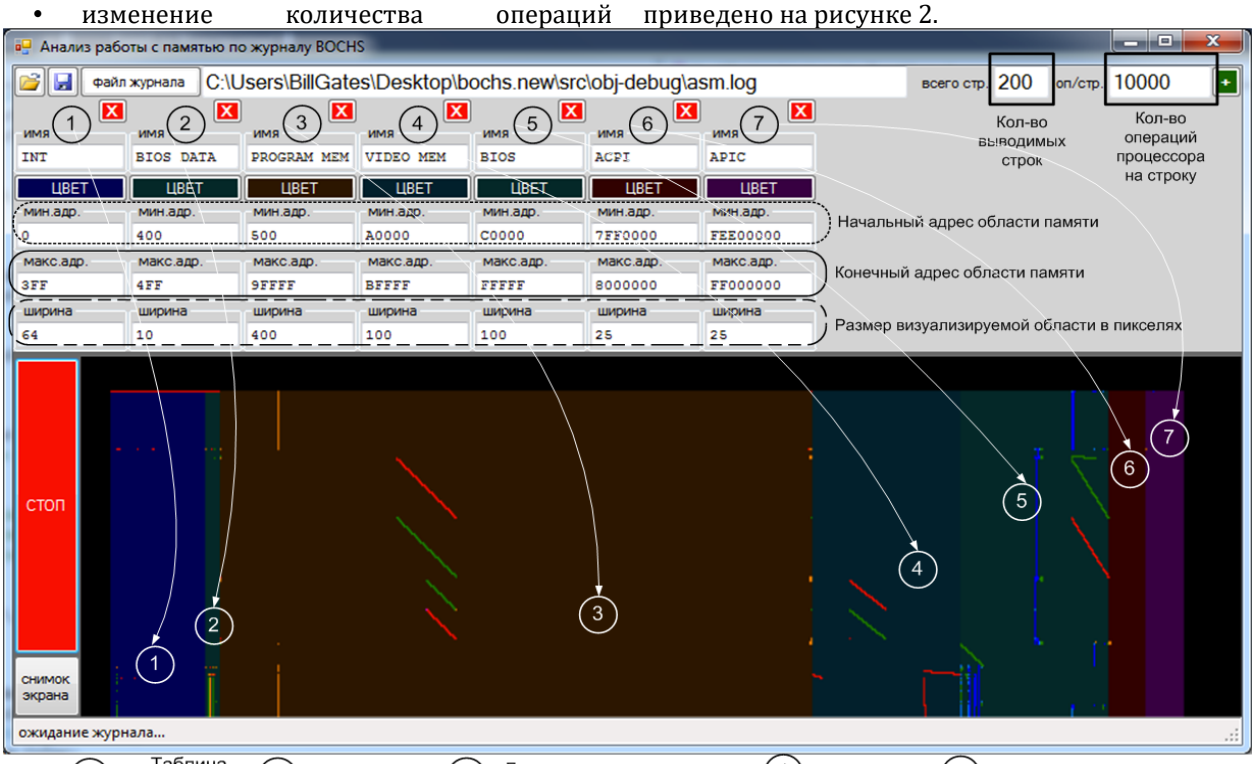
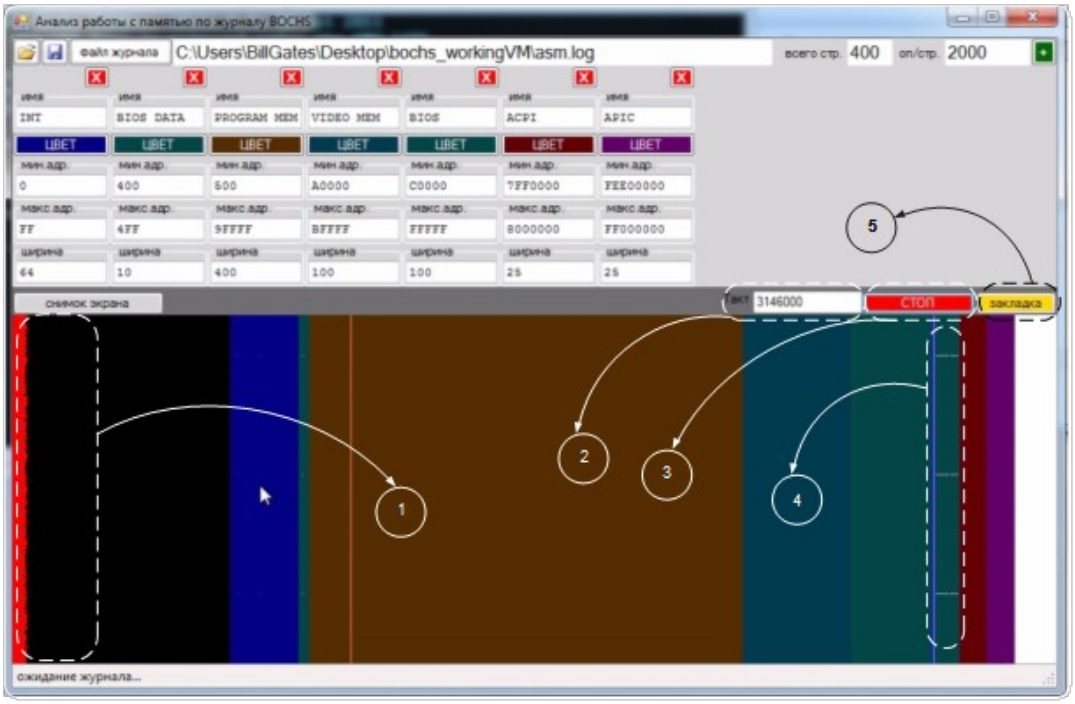


Рисунок 2. Диалоговое окно модуля визуализации версия broute_2013



1 - оценка нагрузки на процессор
2 - контроль тактов процессора
3 - функция «старт-стоп»
4 - трассировщик
5 - закладки

Рисунок 3. Диалоговое окно модуля визуализации версия broute_2015

Дальнейшее развитие программного инструментария [12] (рисунок 3) было направлено на более тонкую настройку и расширение функционала визуализатора – версия broute_2015:

- оценка нагрузки на процессор относительно обрабатываемых данных;
- визуальный контроль количества тактов процессора;
- функция буферизации, позволяющая осуществлять просмотр не только в реальном масштабе времени, но и в режиме «старт-стоп»;
- введение функции трассировщика, которые позволяет «соединять» в одну линию точки в областях памяти, отвечающие за команды чтение, запись и исполнение (необходимо для отслеживания перемещения по областям памяти при минимальном масштабировании);
- возможность установки «закладок», тех мест на визуализаторе, которые интересны эксперту, для того чтобы иметь возможность позже вернуться к более детальному просмотру, а не останавливать процесс работы программы.

Результаты проверки надежности и функциональной безопасности ПП

Эксперименты были проведены на двенадцати программах (10 вирусов и 2 штатные программы для DOS), где проверялись шесть потенциально опасных возможностей, указанные в начале статьи. Результаты указаны в таблице 1.

Таблица 1. Результаты экспериментов

№ п/п Тип ОВ	1	2	3	4	5	6
AVV	+	-	+	-	+	-
Abbas	+	+	+	-	-	-
Adi	+	+	+	+	+	+
Ah	+	-	+	+	+	+
Bomzh	+	+	-	+	+	-
Dir2	+	+	-	+	+	-
Green	+	+	-	+	-	-
Omsk622	+	-	-	+	+	-
Ukraine	+	+	-	-	+	+
Yosha	+	+	+	+	+	+
Keyrus	+	+	+	+	-	-
qbasic	+	-	+	-	-	-

Общий алгоритм проведения работы следующий:

1. создание рабочего образа операционной системы с исследуемым ПП;
2. запуск гипервизора;
3. запуск модуля ПИ выборки команд процессора из журнала гипервизора;
4. загрузка операционной системы с исследуемым ПП под управлением гипервизора;
5. запуск модуля визуализатора ПИ;
6. работа с ПП под управлением гипервизора;
7. контроль за работой ПП на визуализаторе.

Выводы

В настоящее время ПИ направлен на решение узкоспециализированных задач, применительно к СВД и критически значимым системам с целью оценки надежности используемых ПП, при этом он хорошо зарекомендовал себя в последние годы использования. Готовится к выходу версия broute_2017. Планируется, что она позволит исследовать приложения, работающие с сетью, и оценивать влияние их на сетевые интерфейсы. Рассматриваются задачи о переводе ПИ под управление операционной системы семейства MS Windows, для решения которых потребуется не более полутора лет и коллектив из шести человек (потребуется работа коллектива из шести человек на протяжении полутора лет).

Благодарности

За подготовку статьи благодарю Синицына Игоря Николаевича, доктора тех. наук, профессора, заведующего отделом ФИЦ ИУ РАН.

Литература

1. Руководящий документ. Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей. М.: Гостехкомиссия России, 1998.
2. Королев В.Ю., Соколов И.А. Основы математической теории надежности модифицируемых систем / М.: ИПИ РАН. 2006. – 102с.
3. Майерс Г. Дж. Надежность программного обеспечения. М.: 1980.
4. Карповский Е.Я., Чижов С.А. Надежность программной продукции. / Киев. 1990.
5. Выявление уязвимостей программного обеспечения в процессе сертификации / А.С. Марков, С.В. Миронов, В.Л. Цирлов // Известия Южного федерального университета. Технические науки. 2006. Т. 62. № 7. С. 82-87.
6. ЦБИ [Электронный ресурс] Режим доступа: http://www.cbi-info.ru/groups/page-343.htm
7. Система сертификации средств защиты информации по требованиям безопасности информации № РОСС RU.0001.01БИ00 //Государственный реестр сертифицированных средств защиты информации. [Электронный ресурс]. Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/153-sistema-sertifikatsii/591-gosudarstvennyj-reestr-sertifitsirovannykh-sredstv-zashchity-informatsii-n-ross-ru-0001-01bi00.
8. Кубрин С.С., Самарин Н.Н., Современное состояние инструментальных средств анализа программного обеспечения на уязвимость //Горный информационно-аналитический бюллетень (научно-технический журнал), 2013,№ 5, с. 92-102.
9. Тексы программ. [Электронный ресурс]. Режим доступа: http://bochs.sourceforge.net.
10. Самарин Н.Н. Программный макет системы визуализации тракта данных в электронной вычислительной системе «Путь» / Баженов А.С., Борисов А.В., Кудяков К.И., Самарин Н.Н. // Свидетельство о государственной регистрации программы для ЭВМ №2013614133 от 24.04.2013г.

11. Самарин Н.Н. Программный комплекс контроля и визуализации областей памяти электронной вычислительной системы. / Баженов А.С., Борисов А.В., Самарин Н.Н. // Свидетельство о государственной регистрации программы для ЭВМ №2013660975 от 26.11.2013г.
12. Самарин Н.Н. Программный комплекс определения циклов в областях памяти электронной вычислительной системы с их автоматической регистрацией // Борисов А.В., Кубрин С.С., Самарин Н.Н. // Свидетельство о государственной регистрации программы для ЭВМ №2015615141 от 08.05.2015г.

References

1. Rukovodjashhij dokument. Zashhita ot nesankcionirovannogo dostupa k informacii. Chast' 1. Programmnoe obespechenie sredstv zashhity informacii. Klassifikacija po urovnju kontrolja otsutstvija nedeklarirovannyh vozmozhnostej. M.: Gostehkomissija Rossii, 1998.
2. Korolev V.Ju., Sokolov I.A. Osnovy matematicheskoj teorii nadezhnosti modificiruemyh sistem / M.: IPI RAN. 2006. – 102s.
3. Majers G. Dzh. Nadezhnost' programmnogo obespechenija. M.: 1980.
4. Karpovskij E.Ja., Chizhov S.A. Nadezhnost' programmnoj produkcii. / Kiev. 1990.
5. Vyjavlenie ujazvimostej programmnogo obespechenija v processe sertifikacii / A.S. Markov, S.V. Mironov, V.L. Cirlov // Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki. 2006. T. 62. № 7. S. 82-87.
6. CBI [Jelektronnyj resurs] Rezhim dostupa: http://www.cbi-info.ru/groups/page-343.htm
7. Sistema sertifikacii sredstv zashhity informacii po trebovanijam bezopasnosti informacii № ROSS RU.0001.01BI00 //Gosudarstvennyj reestr sertificirovannyh sredstv zashhity informacii. [Jelektronnyj resurs]. Rezhim dostupa: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/153-sistema-sertifikatsii/591-gosudarstvennyj-reestr-sertifitsirovannykh-sredstv-zashchity-informatsii-n-ross-ru-0001-01bi00.
8. Kubrin S.S., Samarin N.N., Sovremennoe sostojanie instrumental'nyh sredstv analiza programmnogo obespechenija na ujazvimost' //Gornyj informacionno-analiticheskij bjulleten' (nauchno-tehnicheskij zhurnal), 2013,№ 5, s. 92-102.
9. Teksy programm. [Jelektronnyj resurs]. Rezhim dostupa: http://bochs.sourceforge.net.
10. Samarin N.N. Programmnyj maket sistemy vizualizacii trakta dannyh v jelektronnoj vychislitel'noj sisteme «Put'» / Bazhenov A.S., Borisov A.V., Kudjakov K.I., Samarin N.N. // Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM №2013614133 ot 24.04.2013g.
11. Samarin N.N. Programmnyj kompleks kontrolja i vizualizacii oblastej pamjati jelektronnoj vychislitel'noj sistemy. / Bazhenov A.S., Borisov A.V., Samarin N.N. // Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM №2013660975 ot 26.11.2013g.
12. Samarin N.N. Programmnyj kompleks opredelenija ciklov v oblastjah pamjati jelektronnoj vychislitel'noj sistemy s ih avtomaticheskoj registraciej // Borisov A.V., Kubrin S.S., Samarin N.N. // Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM №2015615141 ot 08.05.2015g.

Поступила: 26.04.2017

Об авторе:
Самарин Николай Николаевич, начальник научно-исследовательского отделения, Научно-исследовательский институт «Квант», samarin_nik@mail.ru

Note on the author:
Samarin Nikolay, Head of Department, Technology Federal State Unitary Enterprise "Research Institute Kvant", samarin_nik@mail.ru