

Д.И. Приходько, А.В. Мокряков, В.В. Горшков

ОБЗОР ПРИНЦИПИАЛЬНЫХ СХЕМ МУЛЬТИРАЗРЯДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И МЕТОДИКИ ЕГО РАЗРАБОТКИ

Аннотация. Рассматривается особый класс приложений для операционной системы, которая предназначена для работы на нескольких режимах микропроцессора, – многозарядные (мультизарядные) приложения. Назначение рассматриваемых приложений – способность работать на нескольких режимах микропроцессора, который используется в экстремальных условиях окружающей среды. Указанный эффект в микропроцессоре достигается путем программного отделения поврежденных частей кристалла от целой части. Поэтому микропроцессор, режим работы которого определяется накопленными повреждениями, используется преимущественно в экстремальных условиях, к которым относятся условия космоса. В связи с этим для работы вычислительной системы в космических условиях дополнительно необходим не только специальный микропроцессор, но и особое программное обеспечение, которое способно работать на различных режимах микропроцессора. Рассмотрены базовые механизмы этого особого программного обеспечения на примере типа приложения на основе Windows исполняемого файла.

Ключевые слова: мультизарядное программное обеспечение, контейнерные программы, мультизарядные операционные системы.

D.I. Prihodko, A.V. Mokryakov, V.V. Gorshkov

REVIEW OF CIRCUIT DIAGRAMS OF MULTI-BIT SOFTWARE AND METHODS FOR ITS DEVELOPMENT

Abstract. The article describes a special class of applications for operating systems: multi-bit (multi-bit) applications, which are applications that can work in several modes on a microprocessor used in extreme environmental conditions. At the technical level, by using logical separation of damaged parts of the microprocessor and changing the mode of operation of the microprocessor from more to less at the cost of reducing the accuracy of calculations, reduce the likelihood of failure of both programs and the operating system itself. There fore, for a microprocessor whose mode of operation is determined by accumulated damage, and therefore is used mainly in extreme conditions, which include space conditions. There fore, for the operation of a computer system in space conditions, not only a special microprocessor is additionally required, but also special software that is capable of operating in various modes of the microprocessor. The article discusses the basic mechanisms of this special software using the example of the type of application based on the Windows executable file.

Keywords: multi-bit operating systems, changing the bit size of calculations, using disk cache.

Введение

При работе вычислительной системы в экстремальных условиях окружающей среды, как-им является космическое пространство, возникает необходимость в дополнительном увеличении устойчивости системы к данным условиям. Для этого используются два способа [1; 2]:

- дублирование функционала – все компоненты задублированы;
- сжимание имеющегося функционала – в зависимости от степени повреждений изменяется функциональное наполнение.

Приходько Дмитрий Игоревич

магистрант, инженер вычислительных и электронно-вычислительных машин, Научно-исследовательский институт системных исследований Российской академии наук, Москва. Сфера научных интересов: информационные системы, вычислительные системы, операционные системы, эксплуатация вычислительных систем. Автор более 80 опубликованных научных работ. Электронный адрес: mitry1205@mail.ru

Мокряков Алексей Викторович

кандидат физико-математических наук, научный сотрудник Московского авиационного института; научный сотрудник Российского государственного университета имени А.Н. Косыгина «Технологии. Дизайн. Искусство», Москва. Сфера научных интересов: информационные системы, вычислительные системы, математическое моделирование. Автор более 80 опубликованных научных работ. Электронный адрес: ali.latex@gmail.com

Горшков Владимир Владимирович

доктор технических наук, профессор, заведующий кафедрой прикладной математики и системного программирования, Российский государственный университет имени А.Н. Косыгина «Технологии. Дизайн. Искусство», Москва. Сфера научных интересов: информационные системы, вычислительные системы, операционные системы, эксплуатация вычислительных систем. Автор более 200 опубликованных научных работ. Электронный адрес: gorshkov-vv@rguk.ru

Мультиразрядное программное обеспечение (далее – ПО) предназначено для работы в нескольких режимах микропроцессора, поэтому относится автоматически ко второму способу повышения устойчивости вычислительной системы.

Для каждого режима работы микропроцессора требуется отдельная версия системы, поэтому и приложения в простейшем случае будут выглядеть как набор версий в одном файле. Прежде чем перейти к структуре мультиразрядного ПО, необходимо определить особенности рассматриваемых приложений.

Методология

Мультиразрядные приложения в упрощенном виде выглядят как набор версий для каждого режима работы микропроцессора, который считается приемлемым по качеству для выполнения задач. Поскольку исполняемые файлы в каждом семействе систем различны, для описания упрощения реализации необходимо рассмотреть мультиразрядное ПО, основанное на исполняемых файлах операционных систем Windows и MS-DOS, причем последняя взята с учетом того, что она работает в режиме i386 – самом младшем режиме работы микропроцессоров архитектуры AMD 64. Выбор указанной архитектуры объясняется тем же фактом, что и выбор операционной системы, – открытость документации и доступность со стороны пользователей.

Более того, система Windows XP использовалась активно на МКС, а также для одной из миссий на Марс в качестве операционной системы была выбрана Windows 98 SE и микропроцессор Pentium.

Для выбранной архитектуры и семейства операционных систем, взятых за основу для построения вычислительной системы со способностью работать на нескольких режимах, возможны следующие способы построения программ:

- приложение dual exe [3], представляет собой объединенные версии Windows и Dos в один исполняемый файл;
- приложение с технологией кэширования, подразумевает использование технологии дополнительного сохранения данных в специальной области оперативной памяти или жесткого диска;
- приложение с технологией горячего переключения режима работы, данная технология подразумевает использование подмены функций одного режима другим через специальный набор правил (Thunk [4] скрипт);
- мультиверсионная программа, которая собирается из нескольких версий, и при запуске версия достается по некоторому условию из файла;
- контейнерная программа – специальная технология проектирования, заключающаяся в комбинировании предыдущих технологий с учетом выявленных особенностей.

Прежде чем перейти к краткому рассмотрению описанных технологий, необходимо отметить, что достоинства и недостатки всех технологий будут собраны в отдельную единую кросс-таблицу.

dual exe – единая программа, которая создается из приложений dos и Windows NT по правилам PE формата (см. Рисунок 1).

MS-DOS Заголовок
MS-DOS Приложение
Сигнатура PE
PE Заголовок
Оptionальный PE Заголовок
Таблица Секций
... Набор Данных ...

Рисунок 1. Структура исполняемого файла dual exe, использующегося в Windows NTи MS_DOS

Технология инициализации в кэш. Способ организации работы приложения, обладающий следующими особенностями:

- в выделенную область оперативной памяти и (или) жесткого диска, программным способом обозначенную системой как скрытая, выгружаются критические компоненты вычислительной системы;
- помимо критических данных также выгружаются очень легкие приложения, которые могут быть очень быстро приспособлены для нового режима микропроцессора.

Технология горячего переключения разрядности. Главная идея основана на факте использования приложением всех возможностей, являющихся доступными в каждом режиме микропроцессора и версии операционной системы. Задача изменения режима работы приложением заключается только в необходимости осуществлять контроль вычислений, а с точки зрения языков высокого уровня – проверять типизацию данных.

Эта технология может быть использована в крайне ограниченном варианте, поскольку использование специальных вставок на языке assembler может привести к проблемам.

Технология мультиверсионного программного обеспечения. Подразумевает приложение из нескольких версий, при этом каждая версия описывается специализированными требованиями со стороны системы и процессора. Примером для операционных систем Windows выступает приложения dual-exe.

Технология контейнерной программы. Отличается от остальных тем, что используются все доступные выше технологии плюс использование специальных механизмов сжатия программы, то есть операционная система не просто выбирает из файла версию приложения, а создает новую по указанным секциям.

Достоинства и недостатки всех технологий сведены в Таблицу 1.

Таблица 1

Обзорная таблица достоинств и недостатков технологий

Технология	Достоинства	Недостатки
dual-exe [9]	Легко решается вопрос переноса приложения между защищенным и реальным режимом	Технология устарела и требует доработки – не учитывается технология режима Long Mod микропроцессора AMD64 формат PE64 для системы Windows 64 bit
		Итоговый размер прогаммы для системы реального режима работы оказывается тяжелым
Технология инициализации в кэш	Надежность вычислений не страдает при условии целостности данных в специальном кэше	Необходимость держать полное приложение с избыточным размером в специально организованном хранилище
	Технология потребляет крайне малое количество ресурсов оперативной памяти за счет подгружаемой небольшой части, которая практически соответствует объему приложения, спроектированного под конкретную разрядность	Со стороны системы дополнительно требуются механизмы по управлению хранилищем, которое в зависимости от технологии ОС может быть как в специальной области оперативной памяти, так и в области дискового пространства
Горячее переключение разрядности	Высокая скорость переключения в различных режимах работы микропроцессора	Узким местом является низкоуровневая часть программ, или ассемблерные вставки, которые практически не поддаются оптимизации
	В большинстве случаев удается сохранить точность вычислений на необходимом уровне. Со стороны программиста требуется лишь правильный контроль типа данных, который был предложен Microsoft на примере Thunk-технологии	
Мультиверсионная программа	Высокая производительность – распаковывается необходимая версия по аналогии с портфелем	Представление не самое компактное, поэтому требуются избыточные ресурсы для хранения
Контейнерная программа	Относительная компактность представления данных	Сравнительно сложная конструкция, так как версии для каждого режима хранятся не целиком, а частями, из которых идет сборка версии
	Программа в целом обладает высоким быстродействием	Необходимость механизмов адресации

Поэтому контейнерная программа является принципиальной реализацией мультиразрядных приложений, которые представляют собой частную и компактную реализацию мультиверсионных приложений для работы на разных режимах работы микропроцессора.

Результаты исследования

После рассмотрения технологии особенностей построения мультиразрядных приложений необходимо рассмотреть техническую реализацию указанной технологии. Для выбранного семейства операционных систем и архитектуры микропроцессора структура контейнерной программы состоит из следующих особенностей:

- исполняемый файл построен на основе технологии dual exe как частном случае мультиверсионного программного обеспечения;
- критические места кода объединены в обертки специальных процессорных команд – контейнер;
- некритические места меняются по двум правилам:
 - идет подмена функций со стороны операционной системы через смену ядра функций;
 - идет понижение разрядности аргументов в зависимости от требований к работе программы.

Поэтому контейнерная программа не исключает использование описанных решений, но они комбинируются особым образом для выбора оптимального способа для каждого участка программного обеспечения с учетом специфики кода приложения.

Прежде чем перейти к техническому описанию деталей, необходимо определить понятие «контейнер». Под контейнером подразумевается часть программы, вызываемая в процессе сборки приложения для режимов микропроцессора. Контейнер бывает универсальный, когда используется на нескольких режимах микропроцессора, или специализированный – для конкретного режима микропроцессора.

Поэтому секции исполняемого файла у этих приложений состоят из контейнеров. Причем в зависимости от наполнения секций по типам контейнеров можно выделить следующие секции:

универсальные – секции, использующиеся на нескольких режимах микропроцессора; например, такой секцией может быть секция .data, в которой обычно хранятся данные;

специализированная – секция, которая является специфической для каждой из выбранных версий операционных систем и работы микропроцессора.

Поэтому секции могут содержать как минимум 1 контейнер, но чаще всего их несколько, как и секций. Например, контейнером в программе может быть код определения типа архитектуры процессора. Простейшая схема, отображающая принципы построения контейнерной программы, показана на Рисунке 2.

В рамках доступных режимов работы для построения структуры были выбран режим i8086 и MS-DOS для поддержки реального режима работы, режим i386 и система Windows NT для IA32 команд, для режима AMD64 – Windows NT [3].

Согласно Рисунку 2 дублированные участки кода объединяются в контейнеры, которые работают на нескольких режимах работы процессора. Основное назначение такой структуры – сократить время переноса программы при смене режима работы микропроцессора (смену разрядности вычислений). При изменении разрядности поведение программы будет следующим:

1. При смене режима работы микропроцессора секции распадаются на набор контейнеров.
2. Универсальный контейнер, если он работает на данном режиме микропроцессора, продолжает работать.

3. Локальный контейнер меняется на соответствующий конкретному режиму. Если смена невозможна по причине физического отсутствия контейнера в программе, программа выгружается средствами системы из памяти и кэша.

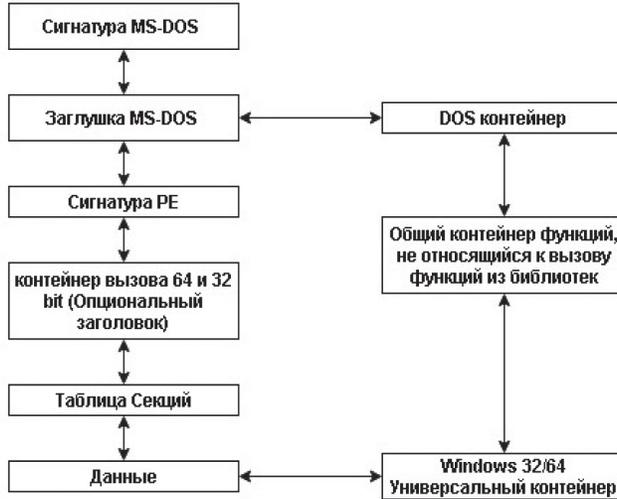


Рисунок 2. Структура контейнерной программы, работающей под DOS-16 bit, Windows 32 bit и Windows 64 bit

Механизмы кэширования данных в контейнерной программе на момент публикации статьи используются следующие:

кэширование данных на жесткий диск – используется специально выделенная область на жестком диске.

кэширование данных в оперативную память – используется специально выделенная область в оперативной памяти.

Важно отметить, что механизмы кэширования целиком и полностью зависят от используемых возможностей в операционной системе [4], которая относится к классу мультиразрядных.

Главное назначение механизмов кэширования и их использование заключается в следующих пунктах:

1. Использование кэширования позволяет ускорить процессы смены режимов работы микропроцессора. Это связано с тем, что при инициализации система из секций приложения собирает целиком и полностью не только версию для текущего режима, но и для всех остальных.

2. Предыдущий пункт автоматически будет означать избыточность потребления ресурсов, доступных для каждого режима работы системы. Для сохранения работоспособности каждого из режимов используется кэширование лишних данных в скрытые области.

Например, стандартный механизм адресации 32 бит составляет 4 Гб, следовательно, используя области более 5 Гб, можно добиться того, что система не будет видеть использования избыточного ресурса ПК, и работа системы не нарушится.

Так как контейнерная программа работает на нескольких режимах микропроцессора, то допускается возможность указать способ развертки с точки зрения приоритетов. Например, для работы драйверов используется преимущественно оперативная память в ка-

честве кэширования, тогда как для громоздких вычислительных пакетов это может быть реализация Matlab, который отвечает за вычисления, поэтому для него используется дисковый кэш, и версии кэшируются на диск. Но это зависит от настроек самой системы, поэтому поведение программ в зависимости от технологии кэша будет аналогичным Рисунку 3 для дискового и Рисунку 4 – для оперативного кэша.

С учетом приведенных структур использования механизмов кэширования на Рисунках 3 и 4 возникает серьезный технологический вопрос: как именно передавать аргументы функций при смене разрядности вычислений.

Для этого можно использовать разработанную систему взаимодействия Windows 16 bit и Windows 32 bit с подсистемами Universal Thunk [4]. Эта технология позволяет приложениям Windows 16 bit вызвать 32 bit функции, и наоборот. Однако она требует доработки для Windows 64 bit.

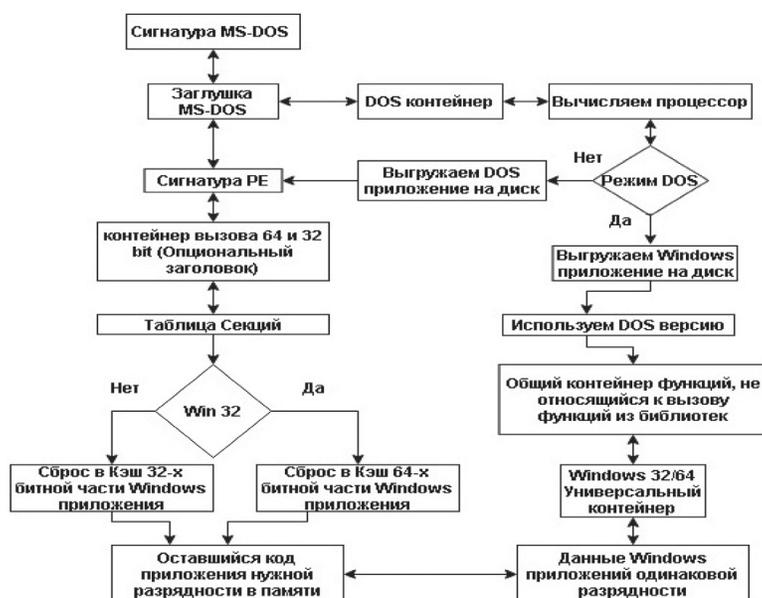


Рисунок 3. Структура контейнерной программы с кэшированием на жесткий диск

Также можно заранее в контейнерной программе указать специальные методы преобразования числовых типов данных из целых чисел в числа с плавающей точкой. Это позволяет для базовых регистров микропроцессора сохранить необходимую для работы точность вычислений.

Поэтому данное ПО пригодно для использования только в экстремальных условиях окружающей среды, которой является космическое пространство. Как правило, механизмы, работающие в экстремальных условиях, являются крайне сложными [6; 7], и поэтому разработка ПО требует особой методики разработки.

Например, для проектирования данного программного обеспечения можно использовать методику, представленную в Таблице 2.

Следует отметить важную особенность: этот тип программного обеспечения зависит от проекта, и внесение поправок в проект требует аккуратности. Преимущества и недостатки предложенной методологии проектирования ПО представлены в Таблице 3.

Обзор базовых механизмов мультиразрядного программного обеспечения ...

Учитывая рассмотренные достоинства и недостатки, предложенную методологию разработки ПО можно использовать для создания мультиразрядных приложений, которые, в свою очередь, могут повысить надежности вычислительной системы. Использование принципа построения на основе понятия «контейнер» позволяет повысить эффективность эксплуатации программ.



Рисунок 4. Использование кэша в оперативной памяти

Таблица 2

Методика проектирования программного обеспечения

Этап	Средства
Разработка идеи информационной системы	Формирование идеи проекта. На данном этапе разработка не идет
Анализ эксплуатационных требований	Оценка технических требований и соответствия внешних условий ИС
Создание концептуальной модели на основе рассмотренного ранее анализа	Выполнение структурно-функционального и/или объектно ориентированного моделирования
Оценка спроектированной концептуальной модели	Оценка построенной концепции с точки зрения эксплуатации, уточнение возникших нюансов
Создание математической модели	Моделирование по типу информационного (логического) и математического
Оценка спроектированной математической модели	Оценка построенной математической или логической модели с точки зрения эксплуатации, уточнение возникших нюансов
Создание первичной реализации ИС	Создание базового функционала ИС, который обязателен для итоговой версии
Экстремальная эксплуатация первичной реализации ИС	Тестирование основного функционала в требуемых условиях эксплуатации. Оценка полученных результатов

Окончание Таблицы 2

Этап	Средства
Разработка итоговой физической модели и итогового программного обеспечения ИС	Реализация выбранной в логической модели логики, разработка итогового аппаратного обеспечения и программного кода на основе математической модели
Экстремальная эксплуатация готовой реализации ИС	Проверка готового технического решения на соответствие ТЗ, корректировки реализации перед передачей заказчику
Эксплуатация ИС	Передача изделия заказчику. Осуществление технической поддержки и гарантийного обслуживания по договоренности с заказчиком

Таблица 3

Достоинства и недостатки предложенной методологии проектирования

Достоинства	Недостатки
Улучшенная модель проектирования позволяет сразу оценить внешние условия	Сложность в проектировании. Не всегда можно однозначно предусмотреть все внешние условия систем
Модель позволяет гибко определить набор испытаний еще на этапе проектирования	Требует периодической корректировки исходного плана реализации
На момент составления проекта решений можно однозначно определить детали и нюансы решения	–

Обсуждение

Принципиальные различия между мультиразрядными и мультиверсионными приложениями заключаются в том, что первые используются для обеспечения работоспособности в различных режимах микропроцессора и поэтому требуют специальной методологии разработки и, более того, эксплуатации. Необходимость обеспечить работоспособность приложений на нескольких режимах микропроцессора обусловлена использованием программного обеспечения орбитальных станций, особенно когда физически произвести замену вышедших из строя компонентов невозможно. Решение аппаратной части проблемы было приведено в патенте [4], однако если ПО не сможет использовать несколько режимов микропроцессора, то возможности его применения существенно ограничиваются.

Ранее рассматривались несколько работ по мультиверсионным приложениям. Например, в работе [8] указывается общее построение мультиверсионного программного обеспечения, но так как не учитывается возможность изменения режимов работы ПО, то гарантировать работоспособность мультиверсионных приложений в общем случае нельзя. В работе [9] рассмотрен вопрос реализации мультиверсионной операционной системы как набора нескольких параллельно работающих версий, однако при этом не рассматривается вопрос о том, что произойдет, когда текущий режим микропроцессора не позволит использовать возможности текущего режима работы, и необходимо будет использовать предыдущий режим. В работе [10] рассматривается модельный аспект мультиверсионного ПО, однако вопрос повреждения микропроцессора там тоже не рассматривается в явном виде.

В связи с этим для эксплуатации ПО в условиях космического пространства требуется особый тип приложений – мультиразрядные приложения, которые работают на нескольких режимах работы.

Заключение

Мультиразрядное приложение в общем случае повышает надежность работы вычислительной системы в экстремальных условиях окружающей среды, к которым относится космическое пространство. При этом данное ПО является подмножеством мультиверсионных приложений и обладает особой спецификой – перестраивает свою работу в зависимости от режима работы микропроцессора.

Приведенный пример построения приложений для мультиверсионной операционной системы на основе систем Windows и MS-DOS является частным случаем. Для построения мультиразрядных приложений в других семействах операционных систем, например Linux, необходим детальный анализ программного обеспечения, которое было выпущено для этой системы.

Технология контейнерного построения мультиразрядных приложений является оптимальной как по размеру версий за счет использования универсальных секций, так и по функционалу за счет подбора режима работы. Но это программное обеспечение может использоваться только в специализированных внешних условиях, к которым относится космическое пространство.

Литература

1. *Артюхова М.А., Кулыгин В.Н.* Разработка автоматизированной системы обеспечения радиационной стойкости бортовой аппаратуры космических аппаратов // *Инновации на основе информационных и коммуникационных технологий.* 2011. № 1. С. 323–324.
2. *Полесский С., Жаднов В., Артюхова М., Прохоров В.* Обеспечение радиационной стойкости аппаратуры космических аппаратов при проектировании // *Компоненты и технологии.* 2010. № 9 (110). С. 93–98.
3. PE-файлы: заголовок и заглушка DOS [Электронный ресурс]. URL: <http://cs.usu.edu.ru/docs/re/dos.html> (дата обращения: 26.09.2022).
4. US6763518B2. Automatic client/server translation and execution of non-native applications: Google Patents. URL: <https://patents.google.com/patent/US6763518> (дата обращения: 26.09.2022).
5. *Приходько Д.И., Мокряков А.В., Горшков В.В.* Обзор базовых механизмов мультиразрядных операционных систем, предназначенных для эксплуатации в космических условиях // *Авиакосмическое приборостроение.* 2022. № 12. С. 39–48.
6. *Мокряков А. В., Приходько Д.И.* Общие концепции и идеи теории эксплуатации в контексте вычислительной техники // *Современная наука: актуальные проблемы теории и практики.* Серия: Естественные и технические науки. 2020. № 12.
7. *Приходько Д.И.* Разработка и оптимизация методов эксплуатации информационных систем как этапа их жизненного цикла с учетом воздействия агрессивной внешней среды / РГУ им. А.Н. Косыгина. М., 2021. 204 с.
8. *Карцан И.Н., Ефремова С.В.* Мультиверсионная модель программного обеспечения систем управления космическим аппаратом с ранжированием принятия решения // *Сибирский аэрокосмический журнал.* 2021. Т. 22, № 1. С. 32–46. DOI: 10.31772/2712-8970-2021-22-1-32-46
9. *Поздняков Д.А.* Компонентная программная архитектура мультиверсионных систем обработки информации и управления // *Современные наукоемкие технологии.* 2006. № 4. С. 62–64.
10. К вопросу реализации мультиверсионной среды исполнения бортового программного обеспечения автономных беспилотных объектов средствами операционной системы реального време-

ни / И.В. Ковалев, В.В. Лосев, М.В. Сарамуд [и др.] // Вестник СибГАУ. 2017. № 1 (18). С. 58–61. EDN: YKCCWL.

Literature

1. Artyukhova M.A., Kulygin V.N. (2011) [Development of an automated system for ensuring radiation resistance of on-board equipment of spacecraft]. *Innovacii na osnove informacionnyh i kommunikacionnyh tehnologij*, 2011, No. 1, Pp. 323–324 (in Russian).
2. Polesky S., Zhadnov V., Artyukhova M., Prokhorov V. (2010) [Ensuring radiation resistance of spacecraft equipment during design]. *Komponenty i tehnologii*, 2010, No. 9 (110), Pp. 93–98 (in Russian).
3. PE files: DOS header and stub. Available at: <http://cs.usu.edu.ru/docs/pe/dos.html> (accessed: 26.09.2022).
4. US6763518B2. Automatic client/server translation and execution of non-native applications: Google Patents. Available at: <https://patents.google.com/patent/US6763518> (accessed: 26.09.2022).
5. Prikhodko D.I., Mokryakov A.V., Gorshkov V.V. (2022) [Review of the basic mechanisms of multi-bit operating systems designed for operation in space conditions]. *Aviakosmicheskoe priborostroenie*, 2022, No. 12, Pp. 39–48 (in Russian).
6. Mokryakov A.V., Prikhodko D.I. (2020) [General concepts and ideas of operation theory in the context of computing]. *Sovremennaja nauka: aktual'nye problemy teorii i praktiki. Serija: Estestvennye i tehniczeskie nauki*, 2020, No. 12 (in Russian).
7. Prikhodko D.I. (2021) *Razrabotka i optimizacija metodov jekspluatacii informacionnyh sistem kak jetapa ih zhiznennogo cikla s uchetom vozdejstvija agressivnoj vneshnej sredy* [Development and optimization of methods for operating information systems as a stage of their life cycle, taking into account the impact of an aggressive external environment]. RGU im. A.N. Kosygina, Moscow, 2021, 204 p. (in Russian).
8. Kartsan I.N., Efremova S.V. (2021) [Multiversion software model of spacecraft control systems with decision ranking]. *Sibirskij ajerokosmicheskij zhurnal*, 2021, Vol. 22, no. 1, Pp. 32–46. DOI: 10.31772/2712-8970-2021-22-1-32-46 (in Russian).
9. Pozdnyakov D.A. [Component software architecture of multiversion information processing and control systems]. *Sovremennye naukoemkie tehnologii*, 2006, No. 4, Pp. 62–64 (in Russian).
10. Kovalev I.V., Losev V.V., Saramud M.V. [On the issue of implementing a multi-version execution environment for on-board software of autonomous unmanned objects using a real-time operating system]. *Vestnik SibGAU*, 2017, No. 1 (18), pp. 58–61 (in Russian). EDN: YKCCWL.