

Реализация программного комплекса разработки программных приложений интегрированной модульной авионики по стандарту ARINC653

Целью разработки программного комплекса является предоставление средства разработки прикладных приложений ИМА и дальнейший их запуск на целевой платформе LynxOS-178 без изменения их исходного кода. Использование данного комплекса позволит как формировать у студентов новые навыки для разработки современных модулей авионики, так и получить более глубокие знания для формирования компетенций в области технического творчества.

В статье предлагается архитектура программного комплекса (ПК) разработки прикладных программ интегрированной модульной авионики (ИМА) с интерфейсом APEX по стандарту ARINC-653 в операционной системе Linux, особенности ее реализации, а также методы разработки программного комплекса.

Предлагаемый подход упрощает процесс разработки приложений ИМА и уменьшает цену разработки, включая тестирование и отладку. Также использование в качестве общедоступной операционной системы ОСРВ Linux с открытым исходным кодом с интерфейсом APEX по стандарту ARINC-653 при разработке прикладных приложений ИМА является решением, лежащим в рамках программы импортозамещения.

Предлагаемый программный комплекс возможно использовать для обеспечения дисциплин, связанных с встраиваемыми вычислительными системами, в качестве средства для разработки приложений ИМА, в рамках освоения следующих компетенций: способность осваивать методики использования программных средств для решения практических задач, способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования, способность сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем, готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов, готовность применять основные методы и инструменты разработки программного обеспечения, владение навыками использования различных технологий разработки программного обеспечения.

Ключевые слова: интегрированная модульная авионика, APEX, ARINC-653, ОСРВ, Linux, OpenVZ, LXC.

Anna V. Korneenkova, Rinat A. Dobrokhотов

Moscow Aviation Institute (National Research University), Moscow, Russia

Linux OS integrated modular avionics application development framework with apex API of ARINC653 specification

The framework is made to provide tools to develop the integrated modular avionics (IMA) applications, which could be launched on the target platform LynxOs-178 without modifying their source code. The framework usage helps students to form skills for developing modern modules of the avionics. In addition, students obtain deeper knowledge for the development of competencies in the field of technical creativity by using of the framework.

The article describes the architecture and implementation of the Linux OS framework for ARINC653 compliant OS application development.

The proposed approach reduces ARINC-653 application development costs and gives a unified tool to implement OS vendor independent code that meets specification. To achieve import substitution free and open-source Linux OS is used as an environment for developing IMA applications.

The proposed framework is applicable for using as the tool to develop IMA applications and as the tool for development of the following competencies: the ability to master techniques of using software to solve practical problems, the ability to develop components of hardware and software systems and databases, using modern tools and programming techniques, the ability to match hardware and software tools in the information and automated systems, the readiness to apply the fundamentals of informatics and programming to designing, constructing and testing of software products, the readiness to apply basic methods and tools of software development, knowledge of various technologies of software development.

Keywords: integrated modular avionics, APEX, ARINC-653, RTOS, Linux, OpenVZ, LXC.

1. Введение

В современном мире, в области приложений вычислительной техники, стали широко использоваться бортовые вычислительные комплексы (БВК). Данный факт объясняется их огромной ролью при решении различных задач в авиации, космонавтике, робототехнике, автомобилестроении, управлении морскими автономными объектами и др.

В настоящее время для нашей страны острым становится вопрос развития собственных исследований и разработок в области новейших технологий.

Сегодня бортовой вычислительный комплекс включает в себя такие группы бортовых информационных систем как: системы сбора данных, системы отображения информации, системы радионавигации, автономные системы радионавигации, системы радиосвязи, системы автоматического пилотирования и другие. Каждая группа систем направлена на выполнение определённой группы задач. Например, система сбора данных предназначена для сбора поступающих сигналов, измерение физические показатели, по которым можно определить статус летательного аппарата. А система отображения информации выдает членам экипажа информацию, собранную с других систем. Данные системы исполняются на вычислительных модулях, которые разрабатываются в соответствии с определенными стандартами, также как и программное обеспечение для этих вычислительных модулей. На данный момент разработка бортовых комплексов ведется по двум основным концепциям построения авионики: федеративная авионика и ИМА.

До ИМА в основе архитектуры авионики лежали принципы федеративной структуры. В федеративной авионике (англ. Federated Avionics) для выполнения одной задачи разрабатывалась система с одним вычислительным модулем [1]. С развитием авионики системы начали выполнять больше функций, а сами функции стали сложнее, также необходимо было решить

проблему уменьшения стоимости и продолжительности разработки комплекса систем [2]. Параллельно развитию авионики, происходило развитие вычислительной техники, при котором достигалось уменьшение физического размера с увеличением производительности без увеличения стоимости. Совокупность данных факторов привело к возникновению новой архитектуры авионики – интегрированная модульная авионика. ИМА представила новую архитектуру системы, предоставляя возможность использовать один и тот же вычислительный модуль для одновременного обслуживания различных систем с помощью изолированных разделов, которые подразумевают изоляцию процессов систем, разделение процессорного времени и памяти между системами [3], [4], [5]. Для формализации требований к разработке систем и приложений авионики нового поколения был разработан стандарт ARINC-653 [3]. Данный стандарт определяет требования к ОС, к изолированным разделам, на которых исполняются приложения ИМА, а также к планированию разделов и процессов разделов и предоставляет интерфейс программирования приложений APEX. Программный интерфейс APEX стандарта ARINC-653 включает в себя 6 сервисов: управление разделами; управление процессами; управление временем; управление памятью; коммуникация между разделами; коммуникации между процессами раздела; сервер мониторинга и обработки ошибок [3].

Стандарт предполагает исполнение приложений в изолированных разделах на одном вычислительном

модуле, при этом разделы должны иметь жесткое разделение процессорного времени. Для работы приложения должны пользоваться функциями программного интерфейса APEX. Архитектура интегрированного модуля, работающего в соответствии со стандартом ARINC-653, приведена на рис. 1.

2. Способы решения проблемы разработки приложений ИМА

На данный момент существует множество коммерческих операционных систем реального времени (ОСРВ) и исследовательских проектов, реализующие стандарт ARINC-653 с программным интерфейсом APEX.

К коммерческим ОСРВ относятся: ОСРВ LynxOS-178, ОСРВ PikeOS, ОСРВ VxWorks 653 [6], [7], [8], [9], [10]. Все данные проекты являются зарубежными разработками без свободного доступа к исходному коду. Данные ОСРВ поддерживают стандарт POSIX и соответствуют стандарту DO-178. Также они включают в себя средства разработки прикладных приложений ИМА, для дальнейшего их использования. На данный момент они активно используются, например, ОСРВ LynxOS-178 используется в самолете Bombardier Challenger 300.

К исследовательским проектам относятся: Patmos и проект корейского университета Konkuk по реализации программного интерфейса APEX в операционной системе Linux [11], [12]. Patmos – одна из ОСРВ, работающая по стандарту ARINC-653, является ОСРВ для

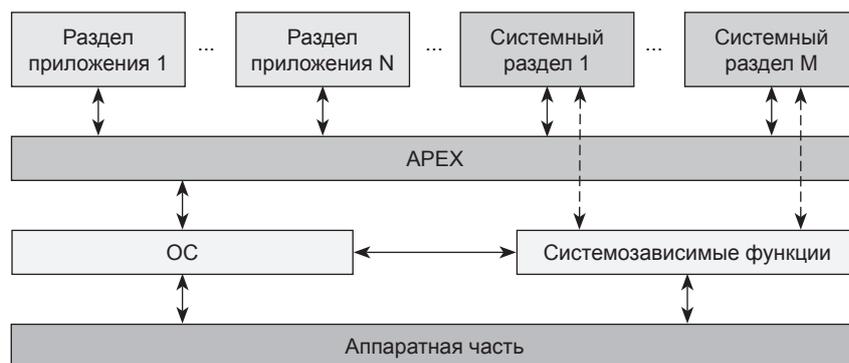


Рис. 1. Архитектура модуля по стандарту ARINC-653

процессора архитектуры Patmos. Данная архитектура процессоров является разработкой Технического Университета Дании, мало распространена на рынке, а проект является закрытым продуктом без доступа к исходному коду.

Поскольку средства разработки программных приложений ИМА не поставляются отдельно от коммерческой ОСРВ, разработка приложений становится дорогостоящим мероприятием, т.к. уже на начальном этапе разработки приложений ИМА возникает необходимость использования таких ОСРВ. При наличии средств разработки прикладных приложений ИМА целесообразно использовать общедоступные ОСРВ на этапе, когда временное соответствие не является значимым. Таким образом, можно в разы уменьшить стоимость разработки данных приложений, т.к. будет достаточно одной системы с ОСРВ для отладки временных характеристик разработанных приложений. Также пропадает зависимость от целевой ОС, т.к. благодаря поддержке программного интерфейса APEX на общедоступной ОС, разработанные на ней прикладные приложения ИМА будут готовы для запуска на любой целевой ОСРВ. В следствии этого, возникла необходимость в разработке программного комплекса (ПК), который бы представлял средство разработки программных приложений ИМА для общедоступной ОС. В данной статье в качестве целевой ОС авторами используется ОСРВ LynxOS-178.

3. Разработка программного комплекса разработки прикладных программ ИМА с интерфейсом APEX по стандарту ARINC-653 в операционной системе Linux

Для решения поставленной задачи, ПК должен поддерживать следующие функции:

- Предоставлять реализацию программного интерфейса APEX стандарта ARINC-653
- Поддерживать изолированные разделы

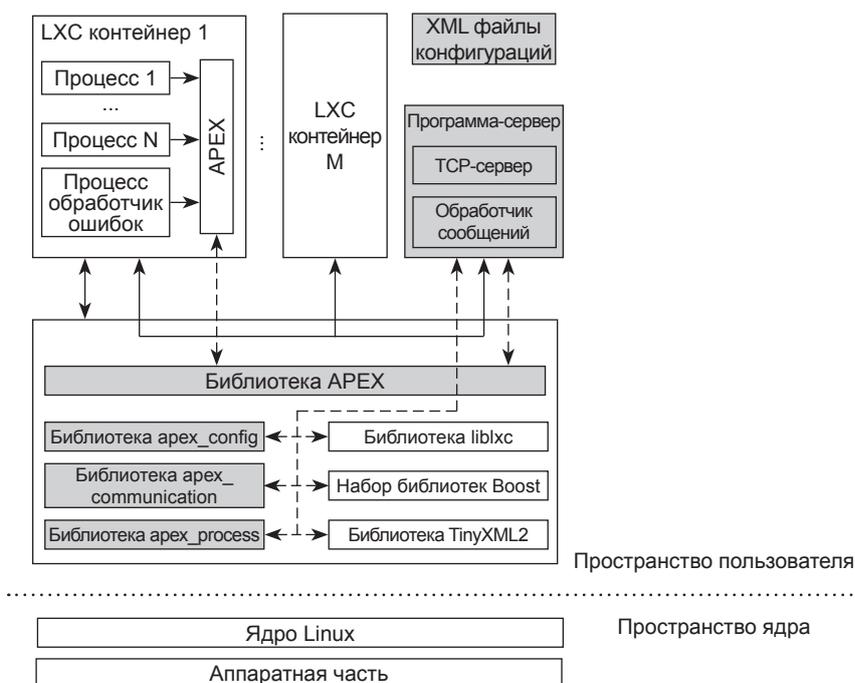


Рис. 2. Архитектура ПК разработки прикладных программ ИМА с интерфейсом APEX по стандарту ARINC-653

- Симулировать разделы согласно стандарту ARINC-653

Выбор ОС РВ Linux связан с рядом ее таких особенностей, как: открытость доступа к исходному коду данной ОС и наличие сертифицированных ОС в Российской Федерации, что является необходимым условием в рамках программы импортозамещения. Также ОС РВ Linux по сравнению с другими ОС имеет следующие преимущества:

- встроенные методы виртуализации;
- кроссплатформенность;
- небольшие потребности в ресурсах.

На основе выбранной ОС была разработана собственная архитектура ПК разработки прикладных программ ИМА с интерфейсом APEX по стандарту ARINC-653, использующая преимущества данной ОС. Разработанная архитектура представлена на рис. 2.

Архитектура программного комплекса включает следующие компоненты:

- программа-сервер – управляет работой ПК. Для правильной работы программы-сервера необходимы конфигурационные файлы.
- библиотека APEX – реализует программный интерфейс APEX.

- вспомогательные библиотеки – необходимы работы ПК и симуляции разделов по стандарту ARINC-653.

Т.к. ПК должен симулировать разделы по стандарту ARINC-653, необходимо было добиться изолирования разделов друг от друга, а также диспетчеризации разделов и процессов согласно стандарту. Для реализации разделов было решено использовать технологию виртуализации, а именно виртуализацию на уровне ОС – контейнеры. В данной технологии виртуализации гостевыми ОС являются контейнеры, которые используют одно и то же ядро вместе с хостовой ОС. Также происходит изолирование процессов, файловых систем и т.д. между контейнерами и хостовой ОС. Преимуществами данной технологии являются:

- Отсутствие виртуализованного оборудования: контейнеры имеют прямой доступ к аппаратной части компьютера
- Простая поддержка из-за использования единой ОС
- Гибкая настройка и масштабируемость: возможность управления доступа к оборудованию компьютера.

В следствии наличия у контейнеров свойств по изолированию,

Таблица

Сравнение технологий виртуализации OpenVZ и LXC

	OpenVZ	LXC
Работа в не модифицированном ядре Linux		•
Управление выделением памяти	•	•
Управление использованием памяти ядром	•	•
Доступность прикладного программного интерфейса (API) на языках программирования	C	C, Lua, Go, Ruby, Python, Haskell

которые удовлетворяют стандарту ARINC-653, было решено реализовать разделы с помощью контейнеров. Авторами были рассмотрены две технологии реализации контейнеров в ОС Linux: LXC и OpenVZ. В результате анализа была выбрана технология LXC, т.к. по сравнению с OpenVZ она поддерживает работу в не модифицированном ядре Linux и имеет доступность прикладного программного интерфейса (API) на языках программирования C, Lua, Go, Ruby, Python, Haskell [13], [14], [15]. Результаты сравнения технологий LXC и OpenVZ представлены в таблице.

Ядром ПК является программа-сервер, которая выполняет следующие функции:

- управляет разделами: создание, запуск, остановка, уничтожение, диспетчеризация;
- управляет процессами: создание, запуск, приостановка, остановка, диспетчеризация;
- обеспечивает работу сервисов программного интерфейса APEX;
- обеспечивает работу ПК в соответствии с конфигурационным файлом.

Т.к. LXC контейнеры являются изолированными друг от друга, также, как и от системы, на которой они запущены, обеспечение взаимодействия между ними выполнено по архитектуре клиент-сервер, где сервером является программа-сервер, а клиентами – разделы. Взаимодействие/обмен данными происходит с помощью сообщений.

Для управления разделами программа-сервер подключает библиотеку liblxc, которая реализует программный интерфейс LXC. Данная библиотека является сторонней разработкой и поставляется вместе с ОС Linux. В данных изолированных разделах исполняются процессы, которые представляют собой приложения ИМА. Данными

процессами также управляет программа-сервер, которая занимается их запуском и диспетчеризацией. Т.к. симуляция разделов по стандарту ARINC-653 включает в себя диспетчеризацию разделов и процессов, исполняющихся в них, в программе-сервере был реализован диспетчер разделов и процессов. Стандарт ARINC-653 описывает диспетчеризацию следующим образом: существует основной квант времени, который разделён на другие более мелкие кванты времени разделов, которые в свою очередь разделены на кванты времени процессов, что показано на рис. 3 [3]. Разделы планируются циклически, а процессы по принципу «первый пришел – первый выполнен» (ПППВ) с приоритетом. Для диспетчеризации разделов достаточно использовать таймер, который по

истечению кванта времени раздела приостанавливает текущий раздел и запускает следующий по очереди. Для диспетчеризации процессов внутри раздела необходимо было выбрать алгоритм диспетчеризации, т.к. согласно стандарту, процессы могут быть двух типов: периодическими и аperiodическими.

Авторами статьи был выбран алгоритм, который предлагает выделить специальный периодический квант времени (СПКВ) для аperiodических процессов, работа во времени которого представлена на рис. 4 [16].

Данный алгоритм наделяет СПКВ, который влияет на диспетчеризацию периодических процессов. Приоритет СПКВ равен приоритету текущего аperiodического процесса. Данный способ имеет преимущество в том, что диспетчеризация процессов является единой для всех типов, а значит не требует разработки дополнительной логики диспетчера. Недостатком данного алгоритма является искусственное ограничение времени исполнения аperiodического процесса. Значения квантов времени разделов и процессов, а также другую необ-

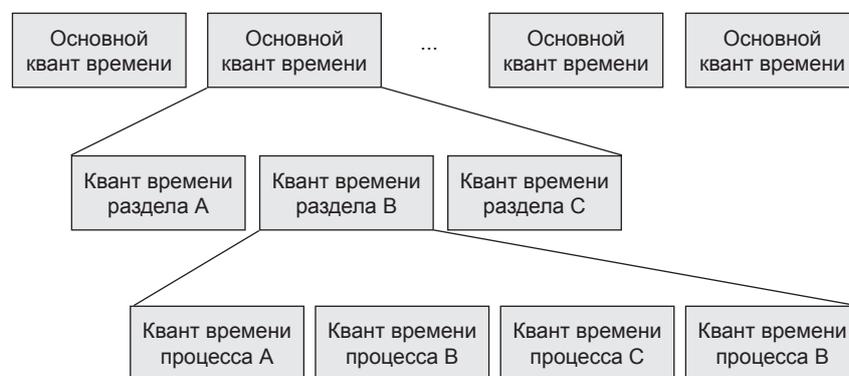


Рис. 3. Диспетчеризация разделов и процессов

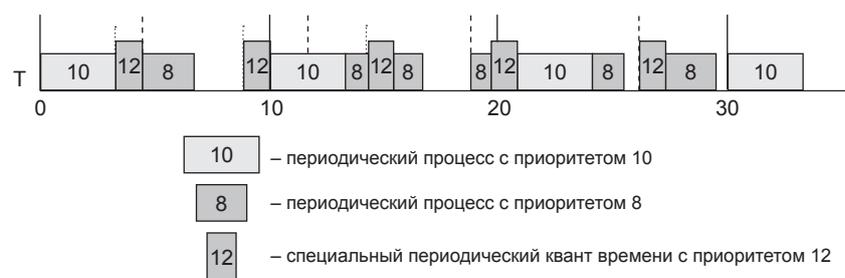


Рис. 4. Одновременная диспетчеризация аperiodических и периодических процессов. Непрерывная линия соответствует точке запуска периодического процесса с приоритетом 10, пунктирная – точке запуска периодического процесса с приоритетом 8, точечная – точке запуска СПКВ

ходимую информацию программа-сервер считывает и хранит в течение работы ПК в конфигурационных файлах, которые в ПК могут быть трёх типов:

- основной конфигурационный файл, определенный стандартом ARINC-653, включающий все основные настройки: таблицу разделов, настройки для мониторинга сервера и обработки ошибок и др.;

- файл конфигурации процессов, в котором находится таблица процессов;

- файл конфигурации средств межпроцессной коммуникации, в котором задается таблица средств межпроцессной коммуникации.

Для разработки прикладных приложений ИМА, а также последующего их запуска в ПК была разработана библиотека, реализующая программный интерфейс АРЕХ. Большинство функций данной библиотеки зависят от программы-сервера, т.к. программа-сервер хранит всю необходимую информацию о системе, на основе которой происходит необходимое действие. Например, при вызове функции GET_PARTITION_STATUS, происходит отправление сообщения программе-серверу, которая в свою очередь отправляет необходимые данные обратно вызывавшему процессу.

Т.к. в качестве целевой ОС была выбрана ОСРВ LynxOS-178, появляется необходимость в трансляции конфигурационного файла формата XML в формат VCT, применяющийся в ОСРВ LynxOS-178. Для этого была разработана и реализована специальная программа, которая на основе шаблона файла формата VCT, транслирует файл формата XML, предложенный стандартом. Конфигурационный файл, полученный с помощью данной программы, готов к использованию в ОСРВ LynxOS-178 без каких-либо изменений.

4. Тестирование разработанного ПК

Тестирование разработанного ПК предполагает разработку и тестирование прикладных приложений в ПК на основе ОС Linux

с дальнейшим запуском разработанных приложений в ОСРВ LynxOS-178.

Для тестирования ПК системы были настроены следующим образом:

- два раздела с квантом времени равным 500000 наносекунд

- в каждом разделе единственный аperiodический процесс

- на один раздел выделено два порта, работающих в режиме очереди: один порт типа «источник», другой порт типа «получатель». Данные порты имеют следующие параметры: максимальная длина очереди равно 10 сообщениям, максимальный размер сообщения равно 512 байтам.

В качестве прикладных приложений ИМА были реализованы две программы, алгоритмы работы которых представлены в листинге 1.

Листинг 1. Алгоритмы работы программ тестирования.

Программа 1.

1. Начало

2. Отправить сообщение программе раздела 2

3. Засечь время отправки сообщения программе раздела 2

4. Получить сообщение от программы раздела 2

5. Извлечь время получения сообщения программой из раздела 2 из полученного сообщения

6. Вычислить разность между временем получения и временем отправки

7. Вывести его на экран

8. Конец

Программа 2.

1. Начало

2. Получить сообщение от программы раздела 1

3. Засечь время получения сообщения от программы раздела 1

4. Записать время получения в сообщение

5. Отправить сообщение программе раздела 1

6. Конец

Каждая программа исполняется на отдельном разделе, обмениваясь друг с другом сообщениями, при этом происходит подсчет времени между отправкой и получением каждого сообщения.

Данное тестирование включает несколько этапов:

- Разработка выше описанных программ в ПК;

- Запуск и тестирование разработанных программ в ПК;

- Генерирование конфигурационного файла для целевой ОСРВ с помощью программы, транслирующей конфигурационный файл стандарта ARINC-653 в файл конфигурации для LynxOS-178 формата VCT;

- Запуск разработанных программ в ПК на целевой ОСРВ;

- Сравнение и анализ результатов выполнения программ.

Тестирование показало, что:

- программный комплекс имеет возможность разработки приложений ИМА без изменения исходного кода программ;

- программный комплекс симулирует разделы по стандарту ARINC-653;

- диспетчеризация разделов реализована по стандарту ARINC-653;

- диспетчеризация процессов достаточно близко реализована к стандарту ARINC-653, что допустимо на первом этапе разработки приложений ИМА;

- программа, транслирующая файлы конфигурации программного комплекса в файл конфигурации формата VCT, формирует синтаксически верные файлы конфигурации и исходные коды инициализирующих процессов, которые безошибочно компилируются с подключением библиотеки libarinc653 ОС LynxOS-178.

5. Заключение

ПК предоставляет средство разработки прикладных приложений ИМА в ОС Linux для дальнейшего их использования на целевых ОСРВ. При этом на этапе разработки и отладки прикладных приложений ИМА разработанный ПК позволяет производить основную разработку в общедоступных ОС, которые не требуют крупных затрат, и, как следствие, отказаться от необходимости использования дорогостоящих ОСРВ, таких как LynxOS-178, VxWorks-653 и т.д. Также при разработке прикладных приложений ИМА использование в качестве общедоступной ОС ОСРВ

Linux с открытым исходным кодом с интерфейсом APEX по стандарту ARINC-653 является решением, лежащим в рамках программы импортозамещения.

Реализованный программный комплекс имеет перспективы развития, которые заключаются в:

- модифицировании диспетчера процессов для поддержки крайних сроков исполнения;
- реализации интегрированной среды разработки на основе программного комплекса для быстрой и эффективной разработки приложений ИМА;
- реализации текстового интерфейса пользователя для управления работой программного комплекса,

а также отслеживания состояния системы;

- реализации возможности расширять программный комплекс в виде единого установочного пакета для облегчения установки программного комплекса на систему разработчика.

Предлагаемый программный комплекс возможно использовать для обеспечения дисциплин, связанных с встраиваемыми вычислительными системами, в качестве средства для разработки приложений ИМА, в рамках освоения следующих компетенций: способность осваивать методики использования программных средств для решения практических задач, способность

разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования, способность сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем, готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов, готовность применять основные методы и инструменты разработки программного обеспечения, владение навыками использования различных технологий разработки программного обеспечения.

Литература

1. *Harish K.* An Approach to Electrical Integration: Integrated Modular Avionics. FAVIT. 2014.
2. *Федосов Е.А., Квочур А.Н.* Авионика ближайшей перспективы. АвиоПанорама. 2013, №4 (100), стр.4–6.
3. Aeronautical Radio Inc. Avionics application software standard interface part 1 required services. ARINC Specification 653P1-2. Aeronautical Radio Inc. 2005.
4. Роль и место бортового оборудования воздушных судов на современном этапе развития авиации. Электрон. текстовые дан. Режим доступа: [<http://www.modern-avionics.ru/analytics/2014/modern-role-of-avionics-aircraft/>]
5. *Gitsuzo B.S.* Tagawa, Marcelo Lopes de Oliveira e Souza. An overview of the integrated modular avionics (IMA) concept. DINCON. 2011.
6. *Золотарев С.* LynxOS-178 – коммерческая ОСРВ для авиации. PCWeek. 2005. №22 // LynxOS-178 – сертифицированная ОСРВ для интегрированной модульной авионики. Мир компьютерной автоматизации. 2006, №5.
7. *Baumann C., Bormer T.* Verifying the PikeOS Microkernel: First Results in the Verisoft XT Avionics Project. Aachener Informatik Berichte. 2009, стр. 20–22.
8. *Benjamin Ip.* Performance Analysis of VxWorks and RTLinux. Languages of Embedded Systems Department of Computer Science. 2001.
9. Ramesh Yerraballi. Real-Time Operating Systems: An Ongoing Review. The 21'st IEEE Real-Time Systems Symposium, WIP Section, Orlando FL, 2000.
10. Wind River VxWorks 653 Операционная система реального времени для ИМА (Интегрированной Модульной Авионики) и средства разработки ПО критических для безопасности встраиваемых компьютерных систем. Электрон. текстовые дан. Режим доступа: [http://www.vxworks.ru/V653flyer_rus.pdf].
11. A real-time operating system for Patmos. Исходный код. Режим доступа: [<https://github.com/t-crest/ospat>].

References

1. *Harish K.* An Approach to Electrical Integration: Integrated Modular Avionics. FAVIT. 2014.
2. *Fedosov E.A., Kvochur A.N.* Avionika blizhayshey perspektivy. AvioPanorama. 2013, №4 (100), Pp. 4–6. (in Russ.)
3. Aeronautical Radio Inc. Avionics application software standard interface part 1 required services. ARINC Specification 653P1-2. Aeronautical Radio Inc. 2005.
4. Rol' i mesto bortovogo oborudovaniya vozdushnykh sudov na sovremennom eta-pe razvitiya aviatsii. [Electronic resource]. – Available at: <http://www.modern-avionics.ru/analytics/2014/modern-role-of-avionics-aircraft/> (in Russ.)
5. *Gitsuzo B.S.* Tagawa, Marcelo Lopes de Oliveira e Souza. An overview of the integrated modular avionics (IMA) concept. DINCON. 2011.
6. *Zolotarev S.* LynxOS-178 – kommercheskaya OSRV dlya aviatsii. PCWeek. 2005. №22 // LynxOS-178 – sertifiitsirovannaya OSRV dlya integrirovannoy modul'noy avioniki. Mir komp'yuternoy avtomatizatsii. 2006, №5. (in Russ.)
7. *Baumann C., Bormer T.* Verifying the PikeOS Microkernel: First Results in the Verisoft XT Avionics Project. Aachener Informatik Berichte. 2009, Pp. 20–22.
8. *Benjamin Ip.* Performance Analysis of VxWorks and RTLinux. Languages of Embedded Systems Department of Computer Science. 2001.
9. Ramesh Yerraballi. Real-Time Operating Systems: An Ongoing Review. The 21'st IEEE Real-Time Systems Symposium, WIP Section, Orlando FL, 2000.
10. Wind River VxWorks 653 Operatsionnaya sistema real'nogo vremeni dlya IMA (Integrirovannoy Modul'noy Avioniki) i sredstva razrabotki PO kriticheskikh dlya bezopasnosti vstraivaemykh komp'yuternykh sistem. [Electronic resource]. – Available at: http://www.vxworks.ru/V653flyer_rus.pdf.
11. A real-time operating system for Patmos. Iskhodnyy kod. [Electronic resource]. – Available at: <https://github.com/t-crest/ospat>.

12. Sanghyun Han, Hyun-Wook Jin. Kernel-Level Arinc 653 Partitioning for Linux. SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing. 2011, стр. 1632–1637, doi>10.1145/2245276.

13. Kirill K. Virtualization in Linux. Электрон. текстовые дан. Режим доступа: [<http://download.openvz.org/doc/openvz-intro.pdf>], 2006.

14. Документация по ядру Linux. Электрон. текстовые дан. Режим доступа: [<http://git.kernel.org/cgi/linux/kernel/git/torvalds/linux.git/tree/Documentation>].

15. Наш опыт тестирования LXC (Linux Containers) на примере Debian Wheezy. Электрон. текстовые дан. – Режим доступа: [<http://habrahabr.ru/company/centosadmin/blog/202482/>].

16. Зеленов С.В. Планирование строго периодических задач в системах реального времени. Труды Института системного программирования РАН. 2011, Том 20.

12. Sanghyun Han, Hyun-Wook Jin. Kernel-Level Arinc 653 Partitioning for Linux. SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing. 2011, Pp. 1632–1637, doi>10.1145/2245276.

13. Kirill K. Virtualization in Linux. [Electronic resource]. – Available at: <http://download.openvz.org/doc/openvz-intro.pdf>, 2006. (in Russ.)

14. Dokumentatsiya po yadru Linux. [Electronic resource]. – Available at: <http://git.kernel.org/cgi/linux/kernel/git/torvalds/linux.git/tree/Documentation>. (in Russ.)

15. Nash opyt testirovaniya LXC (Linux Containers) na primere Debian Wheezy. [Electronic resource]. – Available at: <http://habrahabr.ru/company/centosadmin/blog/202482/>. (in Russ.)

16. Zelenov S.V. Planirovanie strogo periodicheskikh zadach v sistemakh real'nogo vremeni. Trudy Instituta sistemnogo programmirovaniya RAN. 2011, Volume 20. (in Russ.)

Сведения об авторах

Анна Викторовна Корнеевкова,

кандидат технических наук, доцент

Кафедра «Вычислительные машины, системы и сети»

Московский авиационный институт (национальный исследовательский университет), Москва, Россия

Эл. почта: ankorn77@gmail.com

Ринат Алексеевич Доброхотов,

Кафедра «Вычислительные машины, системы и сети»

Московский авиационный институт (национальный исследовательский университет), Москва, Россия

Эл. почта: rad021993@hotmail.com

Information about the authors

Anna V. Korneenkova,

Candidate of Engineering Sciences, Associate Professor

Department «Computers, systems and networks»

Moscow Aviation Institute (National Research University), Moscow, Russia

E-mail: ankorn77@gmail.com

Rinat A. Dobrokhotoy,

Department «Computers, systems and networks»

Moscow Aviation Institute (National Research University), Moscow, Russia

E-mail: rad021993@hotmail.com