Сертифицируемая бортовая операционная система реального времени JetOS для российских проектов воздушных судов

Ю.А. Солоделов <yasolodelov@2100.gosniias.ru>
Н.К. Горелиц <nkgorelits@2100.gosniias.ru >
Государственный научно-исследовательский институт авиационных систем, 125319, Россия, г. Москва, ул. Викторенко, д. 7

Аннотация. JetOS - перспективная бортовая операционная система реального времени (ОСРВ), разработка которой в настоящее время ведется в рамках научно-исследовательской работы ГосНИИАС. В статье указаны предпосылки появления JetOS и рассмотрены работы, ведущиеся по направлению создания ОСРВ. ОСРВ разрабатывается в соответствии с DO-178C и ARINC 653, учтена возможность работы с OpenGL SC. В статье приведены аппаратные аспекты создания ОСРВ – поддержка многоядерности, платформонезависимость и другие. Одной из важнейших задач при разработке ОСРВ является получение сертификационного пакета, соответствующего DO-178C, благодаря чему JetOS можно будет применять при создании и модернизации авионики для гражданской авиации.

Ключевые слова: операционная система реального времени; ОСРВ; интегрированная модульная авионика; ИМА; сертификация; DO-178C; КТ-178C; ARINC 653; авионика.

DOI: 10.15514/ISPRAS-2017-29(3)-10

Для цитирования: Солоделов Ю.А., Горелиц Н.К. Сертифицируемая бортовая операционная система реального времени JetOS для российских проектов воздушных судов. Труды ИСП РАН, том 29, вып. 3, 2017 г., стр. 171-178. DOI: 10.15514/ISPRAS-2017-29(3)-10

1. Введение

Современные комплексы бортового оборудования (КБО) проектируются в соответствии с идеологией, известной как «Интегрированная модульная авионика» (ИМА). [1], [2]. Одной из ключевых особенностей ИМА является возможность исполнения нескольких функциональных приложений (реализующих программную часть той или иной самолетной системы) на одном вычислителе. Необходимым условием при этом является разделение

приложений по времени исполнения и доступным ресурсам (т.е. ограничение вытесняющей многозадачности и контроль доступа к памяти для нескольких приложений).

Такой режим работы приложений обеспечивается операционной системой реального времени (ОСРВ), что делает ОСРВ неотъемлемой и важнейшей частью любого современного вычислительного модуля и комплекса бортового оборудования в целом. Каждое разрабатываемое гражданское воздушное судно (ВС) или комплектующее изделие из состава ВС должно проходить сертификацию в уполномоченной организации. В общем случае объектом сертификации является весь комплекс программных и аппаратных средств, входящих в состав КБО воздушного судна. Чтобы ОСРВ можно было сертифицировать в составе комплектующего изделия или борта гражданской авиации, она должна разрабатываться в соответствии с требованиями DO-178C (в русскоязычной редакции – КТ-178C [3]).

На протяжении длительного времени как в отечественной программе ИМА, так и на разрабатываемых воздушных судах применялись зарубежные ОСРВ (например, VxWorks 653 или Thales MACS2). Однако имеется прецедент, при котором один из разработчиков ОСРВ в одностороннем порядке прекратил сотрудничество с рядом отечественных компаний, что осложнило проводимые работы и поставило вопрос о замене зарубежного продукта на отечественный аналог [4].

В настоящее время ОСРВ, удовлетворяющих требованиям КТ-178C, в нашей стране нет [4]. В такой ситуации задача создания отечественной сертифицируемой ОСРВ стала очень актуальной.

Для решения этой задачи в ГосНИИАС была организована одногодичная научно-исследовательская работа (НИР), в рамках которой был создан задел по основным направлениям, необходимым для создания бортовой ОСРВ. По итогам этой работы была заложена трехлетняя НИР (2017-19), целью которой является разработка основных компонентов ОСРВ под рабочим названием “JetOS”.

2. Разработка ОСРВ

Основная задача, поставленная в рамках НИР – создание работоспособной высокопроизводительной бортовой ОСРВ с сертификационным пакетом, который впоследствии должен быть включен в общий набор сертификационных данных при разработке КБО. К работам привлекается ряд соисполнителей, в частности, ИСП РАН и ДС БАРС.

Необходимо пояснить такую постановку задачи как «создание сертификационного пакета». По действующим в настоящее время международным регламентам сертификации подлежит не ПО, а система (комплектующее изделие), в состав которого упомянутое ПО входит [5]. Поэтому говорить о сертификации ПО в отрыве от разрабатываемого воздушного судна или комплектующего изделия было бы некорректно.

Поскольку данная работа ведется независимо от создания ВС или КИ, ее целью должна являться подготовка всех необходимых материалов (т.е. сертификационного пакета) для последующего включения их в процесс разработки и сертификации комплектующего изделия.

В состав сертификационного пакета входят, помимо прочего, следующие материалы:

1. планы и стандарты, по которым разрабатывается ПО (КТ-178С, гл.11.1 -11.8) [3],
2. требования к разрабатываемому ПО (КТ-178С, гл. 11.9) [3],
3. проект ПО(КТ-178С, гл. 11.10) [3],
4. исходный код самого ПО и тестов (КТ-178С, гл. 11.11, 11.13) [3],
5. широчайший спектр результатов верификации (КТ-178С, гл. 11.14) [3] – от результатов инспекции планов, стандартов, требований и т.п. до результатов прогона тестов, анализа трассируемости между данными жизненного цикла и т.д.

Полный перечень данных жизненного цикла приведен в КТ-178С, гл. 11 [3].

Первоначально задача постановки процессов, удовлетворяющих КТ-178С, ставилась очень широко; планировалась постановка процессов для создания нескольких разнородных компонентов (ядро ОСРВ, графическая система, файловая система, модуль информационной безопасности и т.д.) при участии нескольких компаний-разработчиков. Усугублялась задача тем, что для ее решения планировалось объединить несколько инструментов жизненного цикла линейки IBM Rational, т.е. провести работы по их интеграции и разрешению возникающих проблем для нескольких коллективов сразу. Для решения задачи планировалось применять разработанный ранее в рамках НИР ГосНИИАС инструмент ИСУТ (информационная система управления требованиями), расширяющий возможности продуктов IBM Rational, но работы над ОСРВ показали необходимость расширения требований к данному инструменту.

В результате было принято решение уменьшить объем решаемых задач, и в настоящее время постановка процессов КТ-178С продолжается для двух компаний: ГосНИИАС и ИСП РАН. Забегая вперед, отметим, что объем компонентов, подлежащих разработке в рамках сертификационного процесса, также был сокращен. При более детальном рассмотрении основной акцент в работе перенесен на ядро ОСРВ и основные системные компоненты (например, стандартная библиотека языка С или библиотека, предоставляющая сервисы ARINC 653 [6]).

Параллельно с постановкой процессов велись работы по прототипированию ОСРВ, т.е. разработке пробного кода с целью проверки решений, которые должны лечь в основу проекта ПО. Стоит отметить, что требованиями КТ-178С такой работы не предусмотрено; там код должен разрабатываться исключительно при наличии архитектуры и требований низкого уровня, т.е.

проекта ПО; в свою очередь, проект ПО должен разрабатываться только при наличии требований. Однако практика показала, что разрабатывать архитектуру комплексного ПО, не проведя предварительную проверку решений на практике, очень затруднительно. В связи с этим (и не вступая в противоречие с КТ-178С) был разработан т.н. прототип ОСРВ – работающий код системы, предназначенный для ранней проверки принимаемых архитектурных решений.

Как прототип, так и основной код ОСРВ разрабатывается на языке С. Это обусловлено простотой его синтаксиса сравнительно с более поздними разработками (например, C++) и широкой распространенностью (компиляторы языка С существуют для подавляющего большинства аппаратных платформ в мире). В соответствии с КТ-178С процесс кодирования регламентируется стандартом на кодирование, разработанным в рамках данного проекта и входящим в сертификационный пакет. Стандарт на кодирование в значительной степени базируется на документе MISRA C и синтаксисе языка C99. Стоит отметить, что при создании бортового ПО крайне большое значение уделяется верификации исходного (а зачастую и исполняемого) кода, а широкие синтаксические возможности и развитые парадигмы программирования более современных языков очень усложняют верификацию кода или делают ее вовсе невозможной. Так, в настоящее время применение объектно-ориентированных языков при разработке бортового ПО допускается лишь с существенными ограничениями (см. [7]).

Одним из важных требований к системному бортовому ПО является возможность работы с графической библиотекой OpenGL (как правило, разновидности Safety Critical – OpenGL SC). OpenGL SC является подмножеством стандарта OpenGL для применения в составе критических систем. В настоящее время OpenGL SC представлен версиями 1.0.1 и 2.0. Бортовые функциональные приложения используют сервисы OpenGL для отображения как двухмерных изображений (мнемокадры, таблицы и пр.), так и трехмерных – например, при моделировании рельефа местности.

В работы по ОСРВ с самого начала было заложено графическое направление: создание собственной библиотеки OpenGL SC (причем полностью программной, сертифицируемой по DO-178C и независимой от аппаратуры) и создание графического менеджера, аналогичного компоненту DRM из состава ядра Linux [8]. Для выполнения этих работ были налажены контакты с научными коллективами ИПМ им. М.В. Келдыша РАН и МГТУ им. Н.Э. Баумана; в настоящее время рассматривается вопрос о выделении этих работ в отдельное направление, не привязанное непосредственно к созданию ОСРВ.

Говоря про создание ОСРВ для авионики, нельзя не упомянуть про стандарт ARINC 653 [6]. Данный стандарт, определяющий программный интерфейс и режимы работы бортового функционального ПО, на протяжении ряда лет является общепринятым во всем мире. При разработке JetOS заложена

реализация ARINC 653 наиболее актуальной версии (2015 года), причем не только основных, но и дополнительных сервисов.

Создание ОСРВ как полноценного продукта подразумевает также разработку целого спектра инструментов – в первую очередь, интегрированной среды разработки функционального ПО, а также компонентов ОСРВ, необходимых для отладки, мониторинга и трассировки разрабатываемых приложений.

3. Аппаратные аспекты создания ОСРВ

Специфика архитектуры ИМА (возможность применения разнородных модулей) и современное состояние аппаратных платформ (постоянная модернизация и быстрое развитие аппаратных платформ) диктуют такие требования к ОСРВ, как поддержка многоядерности и легкая переносимость между различными аппаратными платформами с различными архитектурами процессоров.

Вопрос переносимости решается способом, типовым для различных ОСРВ. В архитектуре закладывается разделение на платформо-зависимые и платформо-независимые компоненты, при этом большинство компонентов (и в первую очередь – ядро ОСРВ) должны являться платформо-независимыми, т.е. разрабатываться на языке С и не иметь ассемблерных вставок.

При переносе на новую аппаратную платформу платформо-независимый код ОСРВ компилируется с помощью инструментов, предоставляемых разработчиком данной платформы, и объединяется с низкоуровневым кодом, специально написанным для конкретной аппаратуры. Такой подход позволяет сохранить основную кодовую базу неизменной и одновременно осуществлять поддержку широкого спектра аппаратных решений. При этом сертификационный пакет, полученный в ходе разработки ОСРВ, в значительной части сохраняет свою актуальность (за исключением информации, относящейся к конкретной аппаратной платформе).

Программный продукт может соответствовать требованиям КТ-178С только для определенных аппаратных платформ; при переходе на новую аппаратуру должна проводиться повторная верификация всей портируемой кодовой базы, а весь новый исходный код должен быть разработан в соответствии с процессами КТ-178С. Поэтому в рамках данной НИР была выбрана так называемая основная платформа, для которой будут собираться верификационные данные; ей стал вычислительный модуль МУПД2G на базе процессора P3041 (PowerPC), разработанный компанией НКБ ВС в рамках отечественной программы ИМА, проводившейся под руководством ГосНИИАС. Помимо МУПД2G имеющийся код ОСРВ портирован на платформу на базе P1010 (PowerPC), а также на i.MX6 (на базе архитектуры ARM). Вопрос многоядерности является комплексным. Как указано в CAST-32 (см. [9]), документ DO-178C разрабатывался в то время, когда применение многоядерных процессоров еще не было повсеместным, и в связи с этим специфика применения данного типа платформ в нем (и, соответственно, в КТ-

178С) не отражена. Многоядерность отражается на всех данных жизненного цикла – от планов и стандартов до результатов тестирования, и для ОСРВ данный вопрос находится на стадии проработки. Необходимо отметить также, что вышеупомянутый процессор P3041 является четырехъядерным.

4. Заключение

В настоящее время проект находится на стадии постановки процессов DO-178C и параллельной подготовки артефактов первой версии. Успешно проведено прототипирование на широком спектре компонентов - как системного, так и прикладного уровня (включая графический менеджер, сетевой стек, файловую систему и библиотеку OpenGL). В рамках прототипирования была разработана базовая версия требований высокого уровня и проекта ПО, которая сейчас активно развивается и перерабатывается; параллельно ведутся работы по рефакторингу кодовой базы и созданию инфраструктуры тестирования.

Опыт применения бортовых ОСРВ в мире показывает, что продукт, соответствующий DO-178C может быть сертифицирован и для применения в других отраслях промышленности, что достигается за счет заложенных в стандарте ограничений и жестких требований [10]. Поэтому вопрос адаптации JetOS для индустриальной техники, космоса, транспорта, медицины также является актуальным и активно прорабатывается; при этом основной задачей НИР остается создание работоспособной высокопроизводительной ОСРВ с сертификационным пакетом, который впоследствии можно будет использовать при создании КБО для гражданских самолетов.

Список литературы

[1]. Федосов Е.А. Проект создания нового поколения интегрированной модульной авионики с открытой архитектурой. Полет, №8, 2008 г., стр. 15-22.

[2]. Федосов Е.А. Косьянчук В.В., Сельвесюк Н.И. Интегрированная модульная авионика. Радиоэлектронные технологии, №1, 2015 г., стр. 66-71.

[3]. Квалификационные требования часть 178С, АР МАК, 2014

[4]. Федосов Е.А., Ковернинский И.В., Кан А.В., Волков В.Б., Солоделов Ю.А. Применение операционных систем реального времени в интегрированной модульной авионике. Труды ГосНИИАС: вопросы авионики, №4(24), 2015 г.

[5]. Руководство Р4754 по процессам сертификации высокоинтегрированных сложных бортовых систем воздушных судов гражданской авиации. АР МАК, 2010

[6]. Avionics application software standard interface (ARINC 653). SAE-ITC, 2015

[7]. DO-332: Object-Oriented Technology and Related Techniques. RTCA, December 13, 2011

[8]. Ragav Gopalan. Inside Linux graphics.Intel Embedded, April 2011

[9]. Multi-core Processors (CAST-32A). Certification Authorities Software Team, 2016

[10]. Sven Nordhoff. Successful multicore certification with software-partitiong. SYSGO AG, 2016

Certifiable onboard real-time operation system JetOS for Russian aircrafts design

Yu.A. Solodelov <yasolodelov@2100.gosniias.ru>
N.K. Gorelits <nkgorelits@2100.gosniias.ru >
State Research Institute of Aviation Systems,
125319, Russia, Moscow, Viktorenko Str, 7

Abstract. JetOS is a prospective onboard real-time operating system (RTOS). Nowadays GosNIIAS develops JetOS in the scope of the research and development project. One of the most important tasks during JetOS development is to create the DO-178C certification kit, which will allow JetOS to be used for development and modification of avionics for civil aircraft. Today there is no operating system certified in accordance with DO-178C in Russia, therefore the JetOS creation is the matter of current importance. Using DO-178C requires the developer to have very strict development processes. The arrangement of processes that satisfy the DO-178C requirements is a very responsible and demanding task because of high expectations in the fields of safety and security. JetOS is being developed primarily for onboard equipment based on the integrated modular avionics (IMA). One of the key features of IMA is the ability to execute several functional applications on one target onboard module. The obvious consequence of this feature is a necessity to have a time and resource partitioning of applications. In avionics field application partition along with a host of other features is defined in ARINC 653 international standard, so its support is the significant requirement for JetOS. ARINC 653 defines application programming interface (API) and modes of operation for onboard functional software. JetOS supports the up-to-date version of ARINC 653 (2015) with supplementary services. JetOS also supports the safety-critical graphical library – OpenGL SC; the special implementation of the OpenGL SC library is being developed along with JetOS itself. OpenGL SC services are used to draw two-dimensional and three-dimensional pictures by onboard functional software. JetOS is a certifiable modular cyber-safe real-time operating system, which is designed in order to support several hardware architectures and to be easily adopted for different hardware boards. The scope of the JetOS project also includes creation of the tools necessary for functional software development, especially aircraft systems.

Keywords: real-time operation system; RTOS; integrated modular avionics; IMA; certification; DO-178C; ARINC 653; civil avioncs.

DOI: 10.15514/ISPRAS-2017-29(3)-10

For citation: Solodelov Yu.A., Gorelits N.K. Certifiable onboard real-time operation system JetOS for Russian aircrafts design. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 3, 2017. pp. 171-178 (in Russian). DOI: 10.15514/ISPRAS-2017-29(3)-10

References

[1]. Fedosov E.A. The new generation open architecture IMA project. Polet, №8, 2008 , pp. 15-22 (in Russian).

[2]. Fedosov E.A., Kosyanchuk V.V., Selvesyuk N.I. Integrated Modular Avionics. Radioelektronnye tehnologii, №1, 2015, pp. 66-71 (in Russian).
[3]. Qualification requirements part 178C. IAC, 2014 (in Russian).
[4]. Fedosov E.A., Koverninsky I.V., Kan A.V., Volkov V.B., Solodelov Yu.A. Use of real-time operating systems in the integrated modular avionics. GosNIIAS Proceedings: avionics, №4(24), 2015 г (in Russian).
[5]. R4754. IAC, 2010 (in Russian).
[6]. Avionics application software standard interface (ARINC 653). SAE-ITC, 2015
[7]. DO-332: Object-Oriented Technology and Related Techniques. RTCA, December 13, 2011
[8]. Ragav Gopalan. Inside Linux graphics.Intel Embedded, April 2011
[9]. Multi-core Processors (CAST-32A). Certification Authorities Software Team, 2016
[10]. Sven Nordhoff. Successful multicore certification with software-partitiong. SYSGO AG, 2016