УДК 004.41(075.8)

Вестник СибГАУ
Том 18, № 1. С. 58–61

К ВОПРОСУ РЕАЛИЗАЦИИ МУЛЬТИВЕРСИОННОЙ СРЕДЫ ИСПОЛНЕНИЯ БОРТОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОНОМНЫХ БЕСПИЛОТНЫХ ОБЪЕКТОВ СРЕДСТВАМИ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

И. В. Ковалев, В. В. Лосев*, М. В. Сарамуд, Д. И. Ковалев, М. О. Петросян

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
*E-mail: basilos@mail.ru

Рассматриваются вопросы функциональной и алгоритмической реализации мультиверсионной среды исполнения модулей как компонентов бортового программного обеспечения автономных беспилотных объектов средствами операционной системы реального времени. Приведен один из подходов к реализации мультиверсионной среды исполнения – реализация принципа псевдопараллельности, а именно, имитация параллельного исполнения задач (тредов) путем разделения времени их исполнения. Определено, что функциональный потенциал рассматриваемой операционной системы, выраженный в наличии планировщика тредов и временной базы как инструмента выполнения действий через строго выделенные интервалы времени, механизма очередей, обмена сообщениями, способен быть использован в качестве функциональной поддержки изыскиваемой среды. Также озвучена возможность портирования, т. е. адаптирования к исполнению на однокристальных микроЭВМ (микроконтроллерах), что, в свою очередь, способствует возможности апробации идеи на доступных инструментальных средствах. Циклограммой реализован процесс обмена сообщениями между несколькими тредами как процедуры возврата результата голосования посредством механизма очередей, который является безопасным способом взаимодействия тредов друг с другом и решает проблему совместного доступа нескольких тредов к одному аппаратному ресурсу, роль которого в данном случае играет память. Также приведены основные API-функции, сопровождающие процесс алгоритмизации обмена сообщениями, такие как запуск планировщика, создание тредов, создание очереди, запись в очередь, чтение из нее и пр. Рассматриваемый механизм обмена сообщениями совместно с планировщиком и системой приоритетов, реализуемые средствами операционной системы реального времени, позволяют выстраивать более гибкие алгоритмы вотирования (голосования), способные варьировать весами N-версионных модулей и, как следствие, влиять на показатели надежности мультиверсионного программного обеспечения, в том числе для реализации мультиверсионной среды исполнения бортового программного обеспечения автономных беспилотных объектов.

Ключевые слова: мультиверсионная среда исполнения, голосование, задача, циклограмма, сообщения, очереди, надежность, модуль.

Sibirskii Gosudarstvennyi Aerokosmicheskii Universitet
imeni Akademika M. F. Reshetneva. Vestnik
Vol. 18, No. 1, P. 58–61

TO THE QUESTION OF IMPLEMENTATION OF MULTI-VERSION EXECUTION ENVIRONMENT SOFTWARE OF ONBOARD AUTONOMOUS PILOTLESS OBJECTS BY MEANS OF REAL-TIME OPERATING SYSTEM

I. V. Kovalev, V. V. Losev*, M. V. Saramud, D. I. Kovalev, M. O. Petrosyan

Reshetnev Siberian State Aerospace University
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russian Federation
*E-mail: basilos@mail.ru

The article deals with the functional and algorithmic implementation of multi-version execution environment of modules as components of the onboard software of autonomous pilotless objects by means of real-time operating system. One of the approaches to implement multi-version execution environment implementation of the principle of a pseudo-parallelism (imitation of concurrent execution of tasks (threads) by dividing the time of their execution) are given. It was determined that the functional capacity of the operating system, expressed in the presence of threads scheduler and time base, as a tool for the implementation of actions through a strictly selected intervals, queuing mechanism, messaging, capable of being used as a functional support of sought environment. The article also announced the possibility of porting, which is adapting to the execution on the single-chip microcomputers (microcontrollers), which, in turn, contributes to capability of testing the idea on available workbench. Messaging

process between multiple threads has been implemented by cyclogram, as the procedure of returning of voting result by queuing mechanism, which is a safe way of interaction of threads with each other and solves the problem of sharing multiple threads to the same hardware resources, whose role in this case played by memory. The main API-functions accompanying process of algorithmization of the exchange of messages, such as start scheduler, the creation of threads, creating of the queue, entry in the queue, reading out and others are shown. In the article it is viewed messaging mechanism, in conjunction with the scheduler and priority system implemented by real-time operating system, allow building more flexible algorithms of voting, that can vary the weights of N-versioned modules, and as a result, affect the reliability indices of multi-version software, including for implementation multi-version execution environment of onboard software of autonomous pilotless objects.

Keywords: multi-version execution environment, voting, tasks, cyclogram, messages, queue, reliability, module.

Описание проблемы. Задача проектирования и программной реализации модулей обеспечения функционирования бортового программного обеспечения автономных беспилотных объектов в контексте методологии мультиверсионного программного обеспечения не является тривиальной [1; 2]. При этом не менее важной является задача формирования мультиверсионной среды исполнения программно-реализованных модулей, поскольку минимально необходимыми требованиями обеспечения функционирования подобной среды является соблюдение ряда условий. Во-первых, необходимо обеспечить реализацию адекватного механизма вотирования [3], т. е. механизма принятия решения о достоверности возвращаемого модулем результата путем голосования; во-вторых, обеспечить обмен данными между модулями; в-третьих, обеспечить механизм вытеснения модуля в случае принятия подобного решения арбитром – программным компонентом, реализующим механизм вотирования, а также последующим добавлением нового модуля для сохранения заданного показателя надежности бортового программного обеспечения автономных беспилотных объектов (АБО).

Поиск решения. Одним из подходов к реализации мультиверсионной среды исполнения, в том числе бортового программного обеспечения АБО, является реализация принципа псевдопараллельности, а именно, имитации параллельного исполнения тредов путем разделения времени их исполнения. Подобный функциональный принцип реализуем операционной системой жесткого реального времени RTOS (Real-Time Operating System), а именно, одной из версий – FreeRTOS [4] – портированной, т. е. адаптированной к исполнению на SoC (System On a Chip), что способствует возможности апробации идеи на доступных инструментальных средствах [5].

Рассмотрим циклограмму (см. рисунок) реализации процесса обмена сообщениями между несколькими тредами как процедуры возврата результата голосования посредством механизма очередей, реализуемых FreeRTOS.

Тред-приемник – это задача, реализующая сбор данных от *N*-версионных модулей (Тред-1, Тред-2, …, Тред-*N*) с целью последующего принятия решения арбитром, однако в описываемом процессе выполняющая только функцию приемника [5; 6].

Тред-1, -2 – это задачи, интерпретирующие *N*-версионные модули [7; 8], реализующие функции бортового программного обеспечения управления АБО.

Алгоритмизация процесса обмена сообщениями. Временная составляющая данной циклограммы распределена по равным квантам времени.

Момент времени «0» – инициализируется запуск планировщика FreeRTOS, который задействует состояние «Выполнение» треда с наиболее высоким приоритетом, в данном случае это Тред-приемник.

Примечание. За запуск планировщика отвечает API-функция vTaskSrartScheduler(), за создание треда – API-функция xTaskCreate().

Момент времени «1» – Тред-приемник инициирует попытку произвести чтение элемента из очереди, но переходит в состояние «Блокирование», поскольку очередь пуста в момент ее создания. В данном состоянии Тред-приемник находится до момента возникновения данных в очереди или до момента истечения тайм-аута 120 мс. Следующим этапом является переход в состояние «Выполнение» одного из тредов-передатчиков (Тред-1 или Тред-2). Однозначно декларировать, какой из тредов перейдет в данное состояние, нельзя, поскольку каждый из них имеет равный приоритет. Предположим, это Тред-1.

Примечание. За переход в состояние «Блокирование» и выход из него отвечает API-функция vTaskSuspend() и vTaskResume() соответственно.

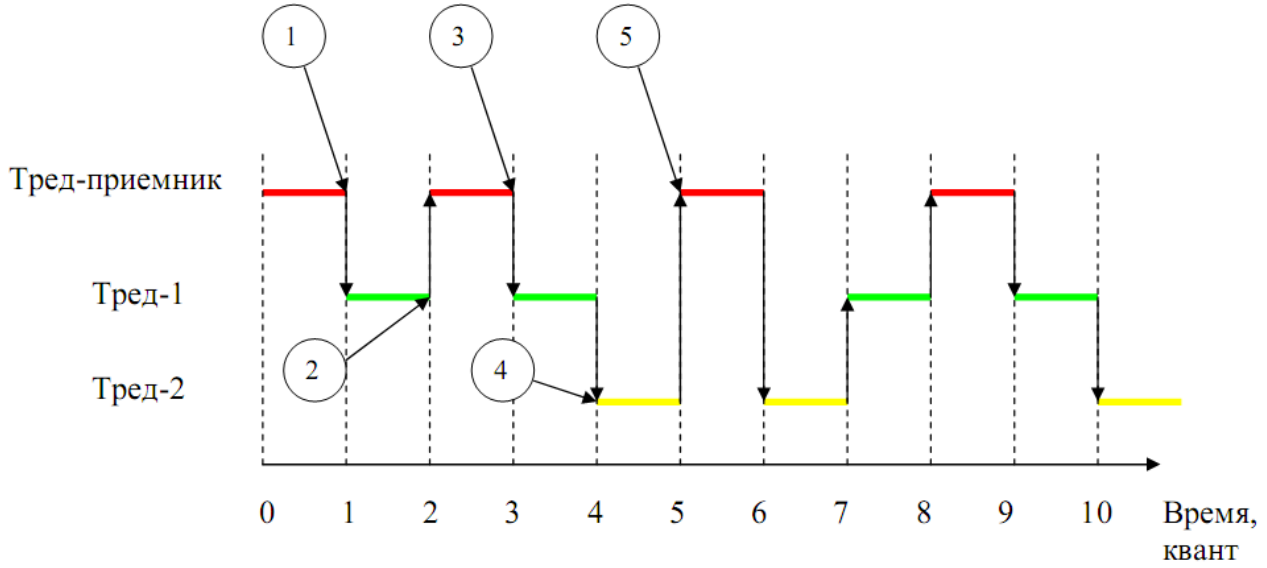
Момент времени «2» – Тред-1 записывает значение «25» в созданную пустую очередь. В этот момент происходит возврат из состояния «Блокирование» Треда-приемника, поскольку функцией данного треда является захват данных из очереди, и приоритет его наиболее высокий.

Примечание. За создание очереди отвечает API-функция xQueueCreate(), для записи элемента в конец очереди – реализация принципа FIFO, используется API-функция xQueueSendToBack(), для записи элемента в начало очереди – реализация принципа LIFO, используется API-функция xQueueSendToFront().

Момент времени «3» – получив доступ к очереди и считав данные из нее, вновь происходит его блокирование, так как очередь теперь пуста. Управление возвращается к прерванному Треду-1, который выполняет API-функцию вызова планировщика *taskYIELD ()*.

Примечание. За чтение элемента с последующим удалением его из очереди отвечает API-функция xQueueReceive().

Момент времени «4» – планировщик переводит в состояние «Выполнение» равноприоритетный Тред-2, который, в свою очередь, записывает значение «50» в очередь.



Циклограмма процесса обмена сообщениями между тредами

Момент времени «5» – выходит из состояния «Блокирование» высокоприоритетный Тред-приемник и производит считывание данных из очереди. Далее происходит следующая итерация цикла.

Примечание. Текущее значение счетчика квантов времени может быть получено с помощью API-функции xTaskGetTickCount().

Заключение. Данный пример демонстрирует механизм обмена сообщениями, согласно которому значение, помещаемое в очередь задачей, интерпретирующей *N*-версионные модули (Тред-1, -2), представлено литерой (константой). Истинное значение, возвращаемое модулем как результат голосования, требует дополнительной алгоритмизации *N*-версионного модуля [9; 10], при этом значение, формируемое модулем, может принимать различный формат, в том числе формат булевых и целочисленных переменных [11], в зависимости от механизма вотирования [12]. Таким образом, механизм обмена сообщениями совместно с планировщиком и системой приоритетов, реализуемые средствами FreeRTOS, позволяют выстраивать более гибкие алгоритмы вотирования (голосования), способные варьировать весами *N*-версионных модулей и, как следствие, влиять на показатели надежности мультиверсионного программного обеспечения [13–15], в том числе для реализации мультиверсионной среды исполнения бортового программного обеспечения АБО.

Благодарности. Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований, Правительства Красноярского края, Красноярского краевого фонда поддержки научной и научно-технической деятельности в рамках научного проекта № 16-47-242143.

Acknowledgements. The work was financially supported by Russian Foundation for Basic Research, Government of Krasnoyarsk Territory, Krasnoyarsk Region Science and Technology Support Fund to the research project № 16-47-242143.

Библиографические ссылки

1. Ковалев И. В., Семенько Т. И., Царев Р. Ю. Методология оценки и повышения надежности программно-информационных технологий и структур : монография / Федер. агентство по образованию ; Краснояр. гос. техн. ун-т. Красноярк, 2005. 160 с.

2. Multiversion environment creation for control algorithm execution by autonomous unmanned objects / I. V. Kovalev [et al.] // IOP Conference Series : Materials Science and Engineering V International Workshop on Mathematical Models and their Applications – 2016 (7–9 November 2016, Krasnoyarsk). 2017. Vol. 173. P. 012025.

3. Ковалев И. В. Анализ проблем в области исследования надежности программного обеспечения: многоэтапность и архитектурный аспект // Вестник СибГАУ. 2012. Вып. № 3 (55). С. 78–92.

4. Barry R. Using the FreeRTOS Real Time Kernel: ARM Cortex-M3 Edition. Real Time Engineers, 2010. P. 196.

5. Инструментальные средства формирования мультиверсионной архитектуры отказоустойчивых программных систем / И. В. Ковалев [и др.] / М-во сельского хоз-ва Российской Федерации ; Краснояр. гос. аграрный ун-т. 2011. С. 152.

6. Стельмах В. О., Ковалев И. В. Построение отказоустойчивых систем управления на основе мультиверсионного подхода // Информационно-телекоммуникационные системы и технологии (ИТСИТ-2012) : материалы Всерос. молодеж. конф. 2012. С. 172–173.

7. Ковалёв П. В. Графоаналитический метод анализа мультиверсионных архитектур программного обеспечения // Международный журнал прикладных и фундаментальных исследований / Академия естествознания. 2009. № 6. С. 70.

8. Kovalev I., Zelenkov P., Ognerubov S. The minimization of inter-module interface for the achievement of reliability of multi-version software // IOP Conference Series : Reshetnev Readings : Materials Science and

Engineering XVII International Scientific Conference. 2015. P. 012006.

9. Kovalev I. V., Zelenkov P. V., Tsarev M. Y. The control of developing a structure of a catastrophe-resistant system of information processing and control // IOP Conference Series : Reshetnev Readings : Materials Science and Engineering XVII International Scientific Conference. 2015. P. 012008.

10. Ковалев И. В., Юнусов Р. В. Мультиверсионный метод повышения программной надежности информационно-телекоммуникационных технологий в корпоративных структурах // Телекоммуникации и информатизация образования. 2003. № 2. С. 50–55.

11. Kovalev I. V., Dgioeva N. N., Slobodin M. Ju. The mathematical system model for the problem of multiversion software design // International Conference on Modelling and Simulation, MS'2004. AMSE, French Research Council, CNRS, Rhone-Alpes Region, Hospitals of Lyon. Lyon-Villeurbanne, 2004.

12. Ковалев И. В., Слободин М. Ю., Ступина А. А. Математическая постановка задачи проектирования *n*-версионных программных систем // Проблемы машиностроения и автоматизации. 2005. № 3. С. 16–23.

13. Engel E. A., Kovalev I. V. Information processing using intelligent algorithms by solving wcci 2010 tasks // Вестник СибГАУ. 2011. № 3 (36). С. 4–8.

14. Лосев В. В., Ковалев И. В. Реинжиниринг информационного обеспечения интегрированных систем управления производством // Приборы. 2010. № 3 (117). С. 31–36.

15. Оценка надежности АСУ с блокирующими модулями защиты / И. В. Ковалев [и др.] // Приборы. 2013. № 6. С. 20–23.

References

1. Kovalev I. V., Semenko T. I., Tsarev R. Yu. *Metodologiya otsenki i povysheniya nadezhnosti programmno-informatsionnykh tekhnologiy i struktur* [The assessment of methodology and improvement of the reliability of software and information technologies and structures]. Krasnoyarsk, Feder. Education Agency, Krasnoyarsk state technical University Publ., 2005, 160 p.

2. Kovalev I. V. et al. Multiversion environment creation for control algorithm execution by autonomous unmanned objects. IOP Conference Series: Materials Science and Engineering V International Workshop on Mathematical Models and their Applications 2016 7–9 November 2016. 2017, Vol. 173, P. 012025.

3. Kovalev I. [Analysis of problems in the field of research of software reliability: the multistage and the architectural aspect]. *Vestnik SibGAU.* 2012, No 3 (55), P. 78–92 (In Russ.).

4. Barry R. Using the FreeRTOS Real Time Kernel: ARM Cortex-M3 Edition. Real Time Engineers, 2010, P. 196.

5. Kovalev I. V. et al. *Instrumental'nye sredstva formirovaniya mul'tiversionnoi arkhitektury otkazo-ustoichivykh programmnykh system* [Tools multiversioning formation of fault-tolerant architecture of software systems]. Krasnoyarsk, M-vo sel'skogo khoz-va Rossiiskoi Federatsii, Krasnoyarskii gos. agrarnyi un-t Publ., 2011, P. 152.

6. Stel'makh V. O., Kovalev I. V. [Building on the basis of fault-tolerant control systems multiversionnyh approach]. Materialy vserossiiskoi molodezhnoi konferentsii "Informatsionno-telekommunikatsionnye sistemy i tekhnologii (ITSIT-2012)" [Information and Telecommunication Systems and Technologies (ITSIT 2012): Proc. youth]. 2012, P. 172–173.

7. Kovalev P. V. [Graphic-analytical method for the analysis of software architectures multiversionnyh]. *Mezhdunarodnyi zhurnal prikladnykh i fundamental'nykh issledovanii, Akademiya estestvoznaniya*. 2009, No. 6, P. 70 (In Russ.).

8. Kovalev I. V., Zelenkov P. V., Ognerubov S. The minimization of inter-module interface for the achievement of reliability of multi-version software. *IOP Conference Series: Materials Science and Engineering 17. Cep. XVII International Scientific Conference "Reshetnev Readings"* 2015, P. 012006.

9. Kovalev I. V., Zelenkov P. V., Tsarev M. Y. The control of developing a structure of a catastrophe-resistant system of information processing and control. *IOP Conference Series: Materials Science and Engineering 17. Cep. XVII International Scientific Conference "Reshetnev Readings"*. 2015, P. 012008.

10. Kovalev I. V. [Multiversioned views method for increasing software reliability information and telecommunication technologies in corporate structures]. *Telekommunikatsii i informatizatsiya obrazovaniya.* 2003, No. 2, P. 50–55 (In Russ.).

11. Kovalev I. V., Dgioeva N. N., Slobodin M. Ju. The mathematical system model for the problem of multiversion software design. *Proceedings of Modelling and Simulation, MS'2004 AMSE : Intern. Conf. on Modelling and Simulation, MS'2004.* AMSE, French Research Council, CNRS, Rhone-Alpes Region, Hospitals of Lyon. Lyon-Villeurbanne, 2004.

12. Kovalev I. V., Slobodin M. Ju., Stupina A. A. [Mathematical formulation of the problem of designing *n*-version software systems]. *Problemy mashinostroeniya i avtomatizacii*. 2005, No. 3, P. 16–23 (In Russ.).

13. Engel E. A., Kovalev I. V. [Information processing using intelligent algorithms by solving wcci 2010 tasks]. *Vestnik SibGAU.* 2011, No. 3 (36), P. 4–8 (In Russ.).

14. Losev V. V., Kovalev I. V. [Reengineering information support of integrated systems of production management]. *Pribory.* 2010, No. 3 (117), P. 31–36 (In Russ.).

15. Kovalev I. et al. Evaluation of the reliability of ACS with blocking protection modules. *Pribory.* 2013, No. 6, P. 20–23 (In Russ.).

© Ковалев И. В., Лосев В. В., Сарамуд М. В., Ковалев Д. И., Петросян М. О., 2017