

РОЗДІЛ II. КОМП'ЮТЕРНІ НАУКИ

УДК 004.422

DOI <https://doi.org/10.26661/2413-6549-2020-2-04>

РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ РОЗРОБЛЕННЯ ДОДАТКА ІНТЕГРОВАНОЇ МОДУЛЬНОЇ АВІОНІКИ ЗА СТАНДАРТОМ ARINC653

Коваленко Ю. Б.

*кандидат педагогічних наук,
доцент кафедри безпеки інформаційних технологій,
докторант
Національний авіаційний університет
пр. Любомира Гузара, 1, Київ, Україна
orcid.org/0000-0002-6714-4258
yleejulee22@gmail.com*

Козлюк І. О.

*доктор технічних наук,
професор кафедри телекомунікаційних систем
Національний авіаційний університет
пр. Любомира Гузара, 1, Київ, Україна
orcid.org/0000-0001-8239-8937
avia_ira@ukr.net*

Ключові слова: *інтегрована модульна авіоніка, підтримка програмного забезпечення, архітектура програмного забезпечення, програмний інтерфейс АРЕХ, стандарт ARINC-653, операційні систем реального часу, OpenVZ, періодичний квант часу.*

Сучасні інтегровані модульні системи авіоніки привносять значну гнучкість у розроблення систем авіоніки, але з такою гнучкістю виникає більш складний процес проектування для точного налаштування програмно-апаратної платформи виконання. Це значно збільшує труднощі в проектуванні системи ІМА порівняно з федеративними архітектурою, де прикладне програмне забезпечення статично розподіляється між її виконавчим обладнанням. Метою розроблення програмного комплексу є надання засобів розроблення прикладних програм ІМА і подальший їх запуск на цільовій платформі LynxOS-178 без зміни вихідного коду. Використання цього комплексу дозволить як формувати нові навички для розроблення сучасних модулів авіоніки, так і отримати більш глибокі знання для формування компетенцій у сфері новітніх технологій. У статті пропонується архітектура програмного комплексу розроблення прикладних програм інтегрованої модульної авіоніки (далі – ІМА) з інтерфейсом АРЕХ за стандартом ARINC-653 в операційній системі Linux, особливості її реалізації, а також методи розроблення програмного комплексу. Запропонований підхід спрощує процес розроблення додатків ІМА і зменшує ціну розроблення, включаючи тестування і налагодження. Також використання як загальнодоступної операційної системи реального часу ОСРЧ Linux із відкритим вихідним кодом з інтерфейсом АРЕХ за стандартом ARINC-653 під час розроблення прикладних програм ІМА є рішенням, що лежить у межах програми імпортозаміщення. Запропонований програмний комплекс можна використовувати для забезпечення дисциплін, пов'язаних із вбудованими обчислювальними системами як засіб для розроблення додатків ІМА, у межах освоєння таких компетенцій, як здатність освоювати методики використання програмних

засобів для розв'язання практичних завдань, здатність розробляти компоненти апаратно-програмних комплексів і баз даних, використовуючи сучасні інструментальні засоби і технології програмування, здатність сполучати апаратні й програмні засоби в складі інформаційних і автоматизованих систем, готовність застосовувати основи інформатики та програмування до проектування, конструювання та тестування програмних продуктів, готовність застосовувати основні методи і інструменти розроблення програмного забезпечення, володіння навичками використання різних технологій розроблення програмного забезпечення.

LINUX OS INTEGRATED MODULAR AVIONICS APPLICATION DEVELOPMENT FRAMEWORK WITH APEX API OF ARINC653 SPECIFICATION

Kovalenko Yu. B.

Ph.D. in Pedagogy,

Associate Professor at the Department of Information Technology Security,

Postdoctoral Student

National Aviation University

Liubomyra Huzara ave., 1, Kyiv, Ukraine

orcid.org/0000-0002-6714-4258

yleejulee22@gmail.com

Kozlyuk I. O.

Doctor of Technical Sciences,

Professor at the Department of Telecommunication Systems

National Aviation University

Liubomyra Huzara ave., 1, Kyiv, Ukraine

orcid.org/0000-0001-8239-8937

avia_ira@ukr.net

Key words: *integrated modular avionics, software support, software architecture, APEX software interface, ARINC-653 standard, real-time operating systems, Linux, OpenVZ, maintenance; retrofitting; periodic quantum of time, LXC.*

The study analyzes the functional principles and approaches to determining the main features of the software used to operate aircraft. This software package provides tools for developing IMA applications and their subsequent launch on the target platform LynxOS-178 without changing their source code. The use of such a complex makes it possible to form new skills for the development of modern avionics modules, as well as acquire in-depth knowledge and skills for the formation of competencies in the field of the latest technologies. The article discusses the architecture of a software package for the development of integrated modular avionics (IMA) applications with the APEX interface according to the ARINC-653 standard in the Linux operating system, reveals the features of its implementation, as well as methods for developing the software package. The proposed approach simplifies and reduces the cost of the development of IMA applications, including testing and debugging. Also, the use of an open source Linux RTOS as a public operating system with an APEX interface according to the ARINC-653 standard in the development of IMA application programs is the basis of the import substitution program. The proposed software package can be used for disciplines related to embedded computing systems, such as tools for developing IMA applications, within the framework of mastering the following competencies: the ability to master the methods of using software for solving practical problems; the ability to develop components of hardware and software systems and databases using modern tools, tools and programming technologies; the ability to combine hardware and software as part of information and automated systems; willingness to apply the basics of computer science and programming in the design, construction and testing of software products; the ability to apply the basic methods and tools of software development, possession of the skills to use various software development technologies.

Вступ. У сфері додатків обчислювальної техніки сучасного світу стали широко застосовуватися бортові обчислювальні комплекси (далі – БОК). Цей факт пояснюється їх величезною роллю у виконанні різних завдань в авіації, космонавтиці, робототехніці, автомобілебудуванні, управлінні морськими автономними об'єктами тощо.

Натепер для нашої країни гострим є питання розвитку власних досліджень і розробок у сфері новітніх технологій.

Сьогодні бортовий обчислювальний комплекс включає такі групи бортових інформаційних систем: збирання даних, відображення інформації, радіонавігації, автономні системи радіонавігації, радіозв'язку, автоматичного пілотування та інші. Кожна група систем спрямована на виконання певної групи завдань. Наприклад, система збирання даних призначена для збирання сигналів, що надходять, вимірювання фізичних показників, за якими можна визначити статус літального апарата. А система відображення інформації видає членам екіпажу інформацію, зібрану з інших систем. Ці системи виконуються на обчислювальних модулях, які розробляють відповідно до певних стандартів, як і програмне забезпечення для цих обчислювальних модулів. Наразі розроблення бортових комплексів ведеться за такими двома основними концепціями побудови авіоніки, як федеративна авіоніка й інтегрована модульна авіоніка (далі – ІМА).

Аналіз стану проблеми. До ІМА в основі архітектури авіоніки лежали принципи федеративної структури. У федеративній авіоніці (англ. Federated Avionics) для виконання одного завдання розроблялася система з одним обчислювальним модулем [1]. Із розвитком авіоніки системи почали виконувати більше функцій, самі функції стали складнішими, а також необхідно було вирішити проблему зменшення вартості й тривалості

розроблення комплексу систем [2]. Паралельно з розвитком авіоніки відбувався розвиток обчислювальної техніки, за якого досягалося зменшення фізичного розміру зі збільшенням продуктивності без збільшення вартості. Сукупність цих факторів призвела до виникнення нової архітектури авіоніки – інтегрованої модульної авіоніки. ІМА представила нову архітектуру системи, надаючи можливість використовувати один і той же обчислювальний модуль для одночасного обслуговування різних систем за допомогою ізольованих розділів, які мають на увазі ізоляцію процесів систем, розподіл процесорного часу і пам'яті між системами [3; 4; 5]. Для формалізації вимог до розроблення систем і додатків авіоніки нового покоління було розроблено стандарт ARINC-653 [3]. Він визначає вимоги до операційних систем (далі – ОС), до ізольованих розділів, на яких виконуються додатки ІМА, а також до планування розділів і процесів розділів і надає інтерфейс програмування додатків APEX. Програмний інтерфейс APEX стандарту ARINC-653 включає 6 сервісів: управління розділами, управління процесами, керування часом, управління пам'яттю, комунікація між розділами, комунікації між процесами розділу, сервер моніторингу і оброблення помилок [3].

Стандарт передбачає виконання додатків в ізольованих розділах на одному обчислювальному модулі, при цьому розділи повинні мати жорстке розділення процесорного часу. Для роботи програми повинні користуватися функціями програмного інтерфейсу APEX. Архітектуру інтегрованого модуля, що працює відповідно до стандарту ARINC-653, наведено на рис. 1.

Мета і завдання дослідження. Метою статті є розроблення програмного комплексу, який надає засоби розроблення прикладних програм ІМА і подальший їх запуск на цільовій платформі LynxOS-178 без зміни їх вихідного коду.

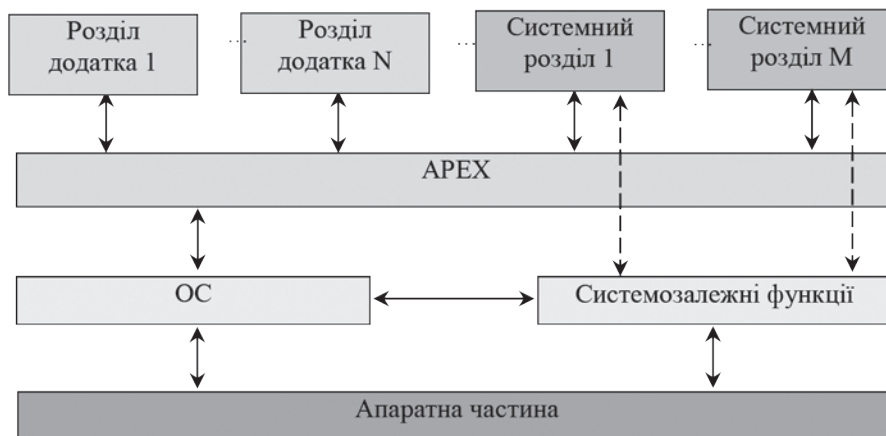


Рис. 1. Архітектура модуля за стандартом ARINC-653

Завдання дослідження – проаналізувати та виявити основні принципи розвитку структур комплексу бортового обладнання; дослідити характерні особливості комплексів, що проектуються на основі інтегральної модульної авіоніки, та зміни, які необхідно внести в процес проектування, для отримання більш оптимальних рішень структур комплекс бортового обладнання на основі інтегральної модульної авіоніки; визначити місце процедури проектування структури функцій у загальному процесі проектування комплексу бортового обладнання.

Виклад основного матеріалу.

Способи вирішення проблеми розроблення додатків ІМА. Натепер є безліч комерційних операційних систем реального часу (далі – ОСРЧ) і дослідницьких проектів, які реалізують стандарт ARINC-653 із програмним інтерфейсом APEX.

До комерційних ОСРЧ належать такі: ОСРЧ LynxOS-178, ОСРЧ PikeOS, ОСРЧ VxWorks 653 [6; 7; 8; 9; 10]. Ці проекти є закордонними розробками без вільного доступу до вихідного коду. Ці ОСРЧ підтримують стандарт POSIX і відповідають стандарту DO-178. Також вони включають засоби розроблення прикладних програм ІМА для подальшого їх використання. Наразі вони активно використовуються (наприклад ОСРЧ LynxOS-178 застосовується в літаку Bombardier Challenger 300).

До дослідницьких проектів належать такі: Patmos і проект корейського університету Konkuk з реалізації програмного інтерфейсу APEX в операційній системі Linux [11; 12]. Patmos – одна з ОСРЧ, що працює за стандартом ARINC-653, є ОСРЧ для процесора архітектури Patmos. Ця архітектура процесорів є розробкою Технічного університету Данії, мало поширена на ринку, а проект є закритим продуктом без доступу до вихідного коду.

Оскільки кошти для розроблення програмних додатків ІМА не поставляються окремо від комерційної ОСРЧ, то розроблення додатків стає дорогим заходом, тому що вже на початковому етапі розроблення додатків ІМА виникає необхідність використання таких ОСРЧ. За наявності коштів для розроблення прикладних програм ІМА доцільно використовувати загальнодоступні ОСРЧ на етапі, коли тимчасова відповідність не є значущою. Таким чином, можна зменшити вартість розроблення цих додатків, тому що буде достатньо однієї системи з ОСРЧ для налагодження тимчасових характеристик розроблених додатків. Також зникає залежність від цільової ОС, тому що завдяки підтримці програмного інтерфейсу APEX загальнодоступною ОС розроблені на ній прикладні програми ІМА будуть готові для запуску на будь-якій цільовій ОСРЧ. Тому виникла необхідність у розробленні про-

грамного комплексу (далі – ПК), який би був засобом розроблення програмних додатків ІМА для загальнодоступної ОС. У цій статті цільовою ОС авторами використовується ОСРЧ LynxOS-178.

Результати розроблення програмного комплексу прикладних програм ІМА з інтерфейсом APEX за стандартом ARINC-653 в операційній системі Linux. Для виконання поставленого завдання ПК повинен підтримувати такі функції:

- надавати реалізацію програмного інтерфейсу APEX стандарту ARINC-653;
- підтримувати ізольовані розділи;
- симулювати розділи відповідно до стандарту ARINC-653.

Вибір ОС РЧ Linux пов'язаний із низкою таких її особливостей, як відкритість доступу до вихідного коду цієї ОС і наявність сертифікованих ОС в Україні, що є необхідною умовою в межах програми імпортозаміщення. Також ОС РЧ Linux (порівняно з іншими ОС) має такі переваги:

- вбудовані методи віртуалізації;
- cross-platform (кросплатформність);
- невеликі потреби в ресурсах.

На основі обраної ОС розроблено власну архітектуру ПК для розроблення прикладних програм ІМА з інтерфейсом APEX за стандартом ARINC-653, що використовує переваги цієї ОС. Розроблену архітектуру представлено на рис. 2.

Архітектура програмного комплексу містить такі компоненти:

- програму-сервер, що керує роботою ПК. Для правильної роботи програми-сервера необхідні конфігураційні файли;
- бібліотеку APEX, що реалізує програмний інтерфейс APEX;
- допоміжні бібліотеки, необхідні роботи ПК і симуляції розділів за стандартом ARINC-653.

Оскільки ПК повинен симулювати розділи за стандартом ARINC-653, необхідно домогтися ізолювання розділів один від одного, а також диспетчеризації розділів і процесів відповідно до стандарту. Для реалізації розділів вирішено використовувати технологію віртуалізації, а саме віртуалізацію на рівні ОС. У цій технології віртуалізації гостьовими ОС є контейнери, які використовують одне і те ж ядро разом із хостовою ОС. Також відбувається ізолювання процесів, файлових систем тощо між контейнерами та хостовою ОС. Перевагами цієї технології є:

- відсутність віртуалізованого обладнання (контейнери мають безпосередній доступ до апаратної частини комп'ютера);
- проста підтримка через використання єдиної ОС;
- гнучке налаштування і масштабування (можливість керування доступом до обладнання комп'ютера).

Унаслідок наявності у контейнерів властивостей з ізолювання, які задовольняють вимоги стандарту ARINC-653, вирішено реалізувати розділи за допомогою контейнерів. Авторами розглянуто такі дві технології реалізації контейнерів в ОС Linux, як LXC і OpenVZ. У результаті аналізу обрано технологію LXC, тому що (порівняно з OpenVZ) вона підтримує роботу в немодифікованому ядрі Linux і доступна прикладному програмному інтерфейсу (API) на мовах програмування C, Lua, Go, Ruby, Python, Haskell [13; 14; 15]. Результати порівняння технологій LXC і OpenVZ представлено в таблиці 1.

Ядром ПК є програма-сервер, що виконує такі функції:

- керує розділами (створення, запуск, зупинка, знищення, диспетчеризація);
- керує процесами (створення, запуск, припинення, зупинення, диспетчеризація);
- забезпечує роботу сервісів програмного інтерфейсу APEX;
- забезпечує роботу ПК відповідно до конфігураційних файлів.

Оскільки LXC контейнери є ізольованими один від одного, як і від системи, на якій вони запущені, забезпечення взаємодії між ними виконано за архітектурою «клієнт-сервер», де сервером є програма-сервер, а клієнтами – розділи. Взаємодія/обмін даними відбувається за допомогою повідомлень.

Таблиця 1

Порівняння технологій віртуалізації OpenVZ і LXC

	OpenVZ	LXC
Робота в немодифікованому ядрі Linux		•
Керування виділенням пам'яті	•	•
Керування використанням пам'яті ядром	•	•
Доступність прикладного програмного інтерфейсу (API) на мовах програмування	C	C, Lua, Go, Ruby, Python, Haskell

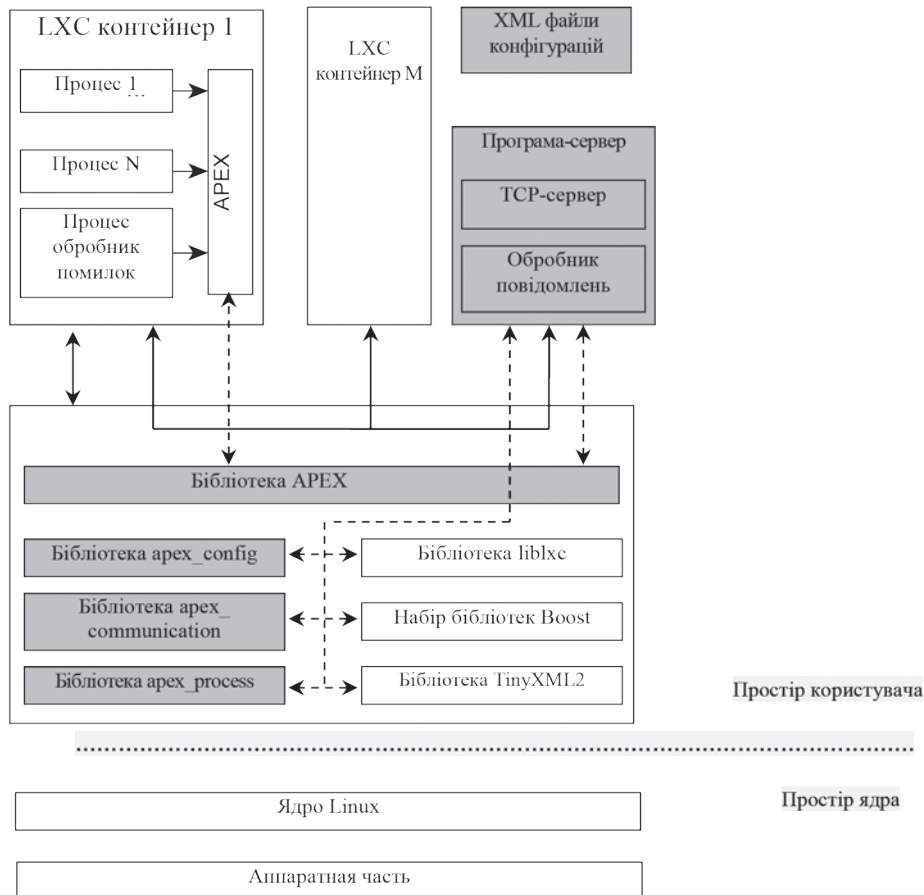


Рис. 2. Архітектура ПК для розроблення прикладних програм IMA з інтерфейсом APEX за стандартом ARINC-653

Для управління розділами програма-сервер підключає бібліотеку `liblxc`, яка реалізує програмний інтерфейс `LXC`. Ця бібліотека є сторонньою розробкою і поставляється разом з ОС Linux. У цих ізольованих розділах виконують процеси, які є додатками ІМА. Цими процесами також керує програма-сервер, яка займається їх запуском і диспетчеризацією. Оскільки симуляція розділів за стандартом ARINC-653 включає диспетчеризацію розділів і процесів, що виконуються в них, у програмі-сервері реалізовано диспетчер розділів і процесів. Стандарт ARINC-653 описує диспетчеризацію так: є основний квант часу, який розділений на більш дрібні кванти часу розділів, які розділені на кванти часу процесів, що показано на рис. 3 [3]. Розділи плануються циклічно, а процеси – за принципом «перший прийшов – перший виконав» із пріоритетом. Для диспетчеризації розділів досить використовувати таймер, який після закінчення кванта часу розділу призупиняє поточний розділ і запускає наступний. Для диспетчеризації процесів усередині розділу необхідно вибрати алгоритм диспетчеризації, тому що (відповідно до стандарту) процеси можуть бути періодичними й аперіодичними.

Отже, обрано алгоритм, який пропонує виділити спеціальний періодичний квант часу (далі – СПКЧ) для аперіодичних процесів, роботу в часі якого представлено на рис. 4 [16].

Цей алгоритм виділяє СПКЧ, який впливає на диспетчеризацію періодичних процесів. Пріоритет СПКЧ дорівнює пріоритету поточного аперіодичного процесу. Цей спосіб переважає в тому, що диспетчеризація процесів є єдиною для всіх типів, а тому не вимагає розроблення додаткової логіки диспетчера. Недоліком цього алгоритму є штучне обмеження часу виконання аперіодичного процесу. Значення квантів часу розділів і проце-

сів, а також іншу необхідну інформацію програма-сервер зчитує і зберігає протягом роботи ПК в конфігураційних файлах, які в ПК можуть бути трьох типів:

- основний конфігураційний файл, визначений стандартом ARINC-653, що включає основні налаштування (таблицю розділів, налаштування для моніторингу сервера й оброблення помилок тощо);
- файл конфігурації процесів, у якому міститься таблиця процесів;
- файл конфігурації засобів міжпроцесної комунікації, в якому задається таблиця коштів міжпроцесної комунікації.

Для розроблення прикладних програм ІМА, а також подальшого їх запуску в ПК розроблено бібліотеку, яка реалізує програмний інтерфейс `APEX`. Більшість функцій цієї бібліотеки залежать від програми-сервера, тому що вона зберігає всю необхідну інформацію про систему, на основі якої відбувається необхідна дія. Наприклад, під час виконання функції `GET_PARTITION_STATUS` відбувається надсилання повідомлення програмі-серверу, яка надсилає необхідні дані обернено до викликального процесу.

Оскільки цільовою ОС обрано ОСРЧ LynxOS-178, то виникає необхідність у трансляції конфігураційного файла формату XML у формат VCT, що застосовується в ОСРЧ LynxOS-178. Для цього розроблено і реалізовано спеціальну програму, яка на основі шаблону файла формату VCT транслює файл формату XML, запропонований стандартом. Конфігураційний файл, отриманий за допомогою цієї програми, готовий до використання в ОСРЧ LynxOS-178 без будь-яких змін.

Обговорення результатів тестування розробленого ПК. Тестування розробленого ПК передбачає розроблення і тестування приклад-



Рис. 3. Диспетчеризація розділів і процесів

них програм у ПК на основі ОС Linux із подальшим запуском розроблених додатків в ОСРЧ LynxOS-178.

Для тестування ПК системи було налаштовано так:

- два розділи з квантом часу, що дорівнює 500 000 наносекунд;
- у кожному розділі єдиний аперіодичний процес;
- на один розділ виділено два порти, що працюють у режимі черги: один порт типу «джерело», інший – типу «одержувач». Ці порти мають такі параметри: максимальна довжина черги дорівнює 10 повідомленням, максимальний розмір повідомлення дорівнює 512 байтам.

Як прикладні програми ІМА реалізовано дві програми, алгоритми роботи яких представлено в лістингу 1.

Лістинг 1. Алгоритми роботи програм тестування.

Програма 1:

1. Початок.
2. Надіслати повідомлення програмі розділу 2.
3. Засікти час відправки повідомлення програмі розділу 2.
4. Отримати повідомлення від програми розділу 2.
5. Витягнути час отримання повідомлення програмою з розділу 2 з отриманого повідомлення.
6. Обчислити різницю між часом отримання і часом відправлення.
7. Вивести його на екран.
8. Кінець.

Програма 2:

1. Початок.
2. Отримати повідомлення від програми розділу 1.
3. Засікти час отримання повідомлення від програми розділу 1.

4. Записати час отримання в повідомленні.
5. Надіслати повідомлення програмі розділу 1.
6. Кінець.

Кожна програма виконується на окремому розділі, обмінюючись один з одним повідомленнями, при цьому відбувається підрахунок часу між відправленням і отриманням кожного повідомлення.

Це тестування включає кілька етапів:

- розроблення вищеприписаних програм у ПК;
- запуск і тестування розроблених програм у ПК;
- генерування конфігураційного файлу для цільової ОСРЧ за допомогою програми, яка транслює конфігураційний файл стандарту ARINC-653 у файл конфігурації для LynxOS-178 формату VCT;
- запуск розроблених програм у ПК на основі цільової ОСРЧ;
- порівняння й аналіз результатів виконання програм.

Тестування показало, що:

- програмний комплекс має можливість розробляти додатки ІМА без зміни вихідного коду програм;
- програмний комплекс симулює розділи за стандартом ARINC-653;
- диспетчеризацію розділів реалізовано за стандартом ARINC-653;
- диспетчеризація процесів досить близько реалізована до стандарту ARINC-653, що допустимо на першому етапі розроблення додатків ІМА;
- програма, яка транслює файли конфігурації програмного комплексу у файл конфігурації формату VCT, формує синтаксично правильні файли конфігурації й вихідні коди ініціувальних процесів, які безпомилково компілюються з підключенням бібліотеки libarinc653 ОС LynxOS-178.

Висновки. ПК надає засіб розроблення прикладних програм ІМА в ОС Linux для подальшого їх використання на цільових ОСРЧ. При цьому на

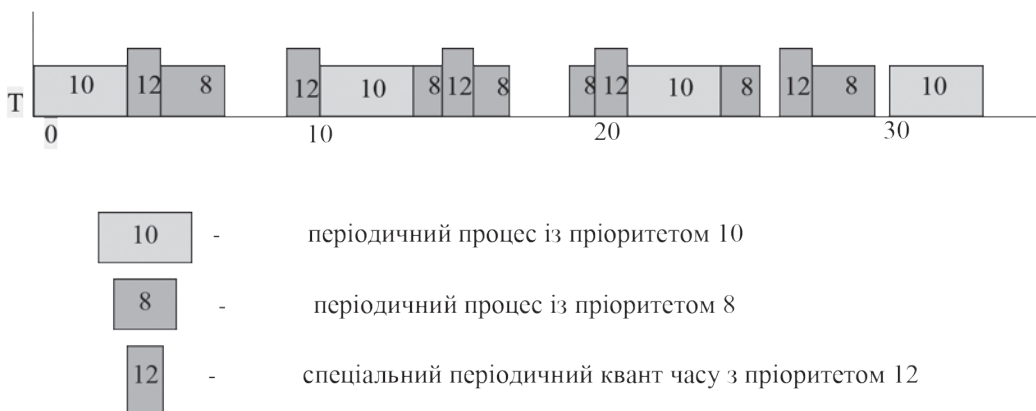


Рис. 4. Одночасна диспетчеризація аперіодичних і періодичних процесів

Безперервна лінія відповідає точці запуску періодичного процесу з пріоритетом 10, пунктирна – точці запуску періодичного процесу з пріоритетом 8, точкова – точці запуску СПКЧ

етапі розроблення і налагодження прикладних програм ІМА розроблений ПК дозволяє виробляти основне розроблення в загальнодоступних ОС, які не вимагають великих витрат, і, як наслідок, відмовитися від необхідності використання дорогих ОСРЧ, таких як LynxOS-178, VxWorks-653 тощо. Також під час розроблення прикладних програм ІМА використані як загальнодоступної ОС ОСРЧ Linux із відкритим вихідним кодом з інтерфейсом APEX за стандартом ARINC-653 є рішенням, що лежить у межах програми імпортозаміщення.

Реалізований програмний комплекс має перспективи розвитку, які полягають у:

- модифікуванні диспетчера процесів для підтримки останніх термінів виконання;
- реалізації інтегрованого середовища розроблення на основі програмного комплексу для швидкого та ефективного розроблення додатків ІМА;
- реалізації текстового інтерфейсу користувача для керування роботою
- програмного комплексу, а також відстеження стану системи;

– реалізації можливості поширювати програмний комплекс у вигляді єдиного установчого пакета для полегшення установки програмного комплексу на систему розробника.

Запропонований програмний комплекс можна використовувати для забезпечення дисциплін, пов'язаних із вбудованими обчислювальними системами, як засіб для розроблення додатків ІМА, в межах освоєння таких компетенцій, як здатність освоювати методики використання програмних засобів для виконання практичних завдань, здатність розробляти компоненти апаратно-програмних комплексів і баз даних, використовуючи сучасні інструментальні засоби і технології програмування, здатність сполучати апаратні й програмні засоби в складі інформаційних і автоматизованих систем, готовність застосовувати основи інформатики та програмування до проектування, конструювання та тестування програмних продуктів, готовність застосовувати основні методи й інструменти розроблення програмного забезпечення, володіння навичками використання різних технологій розроблення програмного забезпечення.

ЛІТЕРАТУРА

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). С. 4–6.
3. Конахович Г.Ф., Козлюк І.О., Коваленко Ю.Б., Оптимізація показників ефективності організаційної системи технічної експлуатації та оновлення радіоелектронних систем повітряних суден. *Системні дослідження та інформаційні технології*. 2020. № 3. С. 41–54.
4. Aeronautical Radio Inc. Avionics application software standard interface part 1 required services. ARINC Specification 653P1-2. Aeronautical Radio Inc. 2005.
5. Роль и место бортового оборудования воздушных судов на современном этапе развития авиации. URL: <http://www.modern-avionics.ru/analytics/2014/modern-role-of-avionics-aircraft/> (дата звернення 04.09.2020).
6. Gitsuzo B.S. Tagawa, Marcelo Lopes de Oliveira e Souza. An overview of the integrated modular avionics (IMA) concept. *DINCON*. 2011.
7. Золотарев С. LynxOS-178 – коммерческая ОСРЧ для авиации. *PCWeek*. 2005. № 22.
8. Baumann C., Bormer T. Verifying the PikeOS Microkernel: First Results in the Verisoft XT Avionics Project. *Aachener Informatik Berichte*. 2009. pp. 20–22.
9. Benjamin Ip. Performance Analysis of VxWorks and RTLinux. Languages of Embedded Systems Department of Computer Science. 2001.
10. Ramesh Yerraballi. Real-Time Operating Systems: An Ongoing Review. The 21'st IEEE Real-Time Systems Symposium, WIP Section, Orlando Fl, 2000.
11. Wind River VxWorks 653 Операционная система реального времени для ИМА (Интегрированной Модульной Авионики) и средства разработки ПО критических для безопасности встраиваемых компьютерных систем. URL: http://www.vxworks.ru/V653flyer_rus.pdf (дата звернення 04.09.2020).
12. A real-time operating system for Patmos. Исходный код. URL: <https://github.com/t-crest/ospat> (дата звернення 04.09.2020).
13. 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.
14. Kovalenko Y., Konakhovych H., Kozlyuk I. Specificity of optimization of performance indicators of technical operation and updating of radio electronic systems of aircraft. *International Journal of Engineering Research and Applications (IJERA)*. 2020. Т. 10, № 9. P. 48–58.
15. Документация по ядру Linux. URL: <http://git.kernel.org/cgi/linux/kernel/git/torvalds/linux.git/tree/documentation> (дата звернення 04.09.2020).

16. Козлюк І.О., Коваленко Ю.Б. Функціональні засади розробки та експлуатації програмного забезпечення в авіоніці. *Проблеми інформатизації та управління*, № 63, 2020, С. 49-63.
17. Зеленев С.В. Планирование строго периодических задач в системах реального времени. *Труды Института системного программирования РАН*. 2011, Т. 20.

REFERENCES

1. Harish K. (2014) *An Approach to Electrical Integration: Integrated Modular Avionics*. FAVIT.
2. Fedosov E.A., Kvochur A.N. (2013) Avionika blizhayshey perspektivy. *AvioPanorama*, no. 4 (100), pp. 4–6 (in Russ.).
3. Konakhovych H., Kozlyuk I., Kovalenko Y. (2020) Specificity of optimization of performance indicators of technical operation and updating of radio electronic systems of aircraft. *System research and information technologies*, no. 3, pp. 41–54.
4. Aeronautical Radio Inc. Avionics application software standard interface part 1 required services. ARINC Specification 653P1-2. Aeronautical Radio Inc. 2005.
5. Rol' i mesto bortovogo oborudovaniya vozdushnykh sudov na sovremennom etape razvitiya aviatsii. URL: <http://www.modern-avionics.ru/analytics/2014/modern-role-of-avionics-aircraft/> (accessed 04.09.2020).
6. Gitsuzo B.S. Tagawa, Marcelo Lopes de Oliveira e Souza (2011) *An overview of the integrated modular avionics (IMA) concept*. DINCON.
7. Zolotarev S. (2005) LynxOS-178 – kommercheskaya OSRV dlya aviatsii. *PCWeek*, no. 22.
8. Baumann C., Borner T. (2009) Verifying the PikeOS Microkernel: First Results in the Verisoft XT Avionics Project. *Aachener Informatik Berichte*, pp. 20–22.
9. Benjamin Ip. (2001) *Performance Analysis of VxWorks and RTLinux. Languages of Embedded Systems Department of Computer Science*.
10. Ramesh Yerraballi (2000) Real-Time Operating Systems: An Ongoing Review. *The 21st IEEE Real-Time Systems Symposium, WIP Section, Orlando FL*.
11. 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. URL: http://www.vxworks.ru/V653flyer_rus.pdf (accessed 04.09.2020).
12. A real-time operating system for Patmos. Iskhodnyy kod. URL: <https://github.com/t-crest/ospat> (accessed 04.09.2020).
13. Sanghyun Han, Hyun-Wook Jin. (2011) Kernel-Level Arinc 653 Partitioning for Linux. *SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 1632–1637. DOI:10.1145/2245276.
14. Kovalenko Y., Konakhovych H., Kozlyuk I. (2020) Specificity of optimization of performance indicators of technical operation and updating of radio electronic systems of aircraft. *International Journal of Engineering Research and Applications (IJERA)*, vol. 10 (09), pp. 48–58.
15. Dokumentatsiya po yadru Linux. URL: <http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation> (accessed 04.09.2020).
16. Kozlyuk I., Kovalenko Y. (2020) Functional bases of the software development and operation in avionics. *Problems of Informatization and Management*, no. 63, pp. 49–63.
17. Zelenov S.V. (2011) Planirovanie strogo periodicheskikh zadach v sistemakh real'nogo vremeni. *Trudy Instituta sistemnogo programmirovaniya RAN*, vol. 20 (in Russ.).