

УДК 004.42

## РОЗРОБКА АРХІТЕКТУРИ ОПЕРАЦІЙНОЇ СИСТЕМИ РЕАЛЬНОГО ЧАСУ ДЛЯ ВБУДОВАНИХ СИСТЕМ НА БАЗІ RISC-V

Данилов А.Д., студент

*Київський національний університет технологій та дизайну*

Мельник Г.В., кандидат технічних наук, доцент

*Київський національний університет технологій та дизайну*

*Ключові слова:* операційна система реального часу, планувальник завдань, RISC-V, вбудовані системи, контекст процесора.

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням відкритих архітектур наборів команд, серед яких провідне місце посідає RISC-V. Завдяки своїй модульності та відсутності ліцензійних обмежень, RISC-V стає базою для створення спеціалізованих обчислювальних систем. Проте ефективне використання апаратних можливостей цієї архітектури у системах, критичних до часу відгуку, неможливе без розробки оптимізованої архітектури операційної системи реального часу (ОСРЧ).

Основним завданням при проектуванні ОСРЧ для вбудованих систем є забезпечення детермінованості поведінки системи під час виконання багатьох паралельних процесів. Це вимагає розробки багаторівневої архітектури, яка б мінімізувала затримки переривань та забезпечувала ефективне перемикання контексту між задачами з різними пріоритетами.

Запропонована архітектура базується на концепції мікроядра, що дозволяє винести більшість сервісів системи за межі основи коду ядра у вигляді окремих незалежних модулів, підвищуючи загальну відмовостійкість. Центральним елементом керування є блок керування задачами (TaskControlBlock– TCB). Проектування структури TCB є критичним для оптимізації використання обчислювальних ресурсів.

Структура TCB включає:

- `pxTopOfStack`– вказівник на поточну вершину стека задачі;
- `xStateListItem`– елемент списку для відстеження стану задачі (готова, заблокована, очікування);
- `uxPriority`– рівень пріоритету для планувальника;
- `pxStack`– вказівник на початок виділеної області пам'яті під стек.

Процес ініціалізації задачі передбачає підготовку стека таким чином, щоб він імітував стан процесора після виклику переривання. Це дозволяє використовувати єдиний механізм для першого запуску задачі та для її подальшого відновлення після витіснення.

Іншим важливим елементом ядра ОСРЧ є планувальник виконання задач. В основі планувальника лежить алгоритм фіксованих пріоритетів із витісненням. Для архітектури RISC-V реалізація цього механізму потребує специфічної обробки системних викликів та таймерних переривань. При

настанні переривання стан реєстрів загального призначення (від x1 до x31) зберігається у стек поточної задачі.

Важливим аспектом є реалізація програмного переривання для перемикання контексту. У RISC-V це реалізується через маніпуляції з реєстрами керування та стану (CSR), зокрема mstatus та перс. Планувальник аналізує списки готових задач і обирає ту, що має найвищий пріоритет. Якщо пріоритети рівні, для розподілу ресурсів використовується круговий алгоритм (RoundRobin) із квантуванням часу.

Для забезпечення цілісності даних у багатозадачному середовищі в основу архітектури закладено використання спеціалізованих примітивів синхронізації. Пріоритетним методом міжпроцесної взаємодії обрано черги повідомлень, що працюють за принципом передачі даних шляхом копіювання. Це дозволяє ізолювати контексти задач та мінімізувати ризик виникнення помилок при маніпуляціях зі спільною пам'яттю. Окрім стандартних механізмів семафорів та м'ютексів, проєктне рішення передбачає інтеграцію протоколу успадкування пріоритетів. Такий підхід спрямований на вирішення класичної для комп'ютерних наук проблеми інверсії пріоритетів, що є критично важливим для забезпечення надійності систем реального часу.

Запропонована архітектурна концепція ОСРЧ для платформи з набором інструкцій RISC-V орієнтована на досягнення високої прогнозованості виконання задач при мінімальному споживанні ресурсів мікроконтролера. Використання багаторівневої моделі з виокремленим шаром апаратних абстракцій (HAL) забезпечує потенційну масштабованість системи та полегшує можливу модифікацію системи під інші пристрої на основі RISC-V. Подальший розвиток передбачає розробку та теоретичне обґрунтування алгоритмів динамічного розподілу пам'яті, які б враховували специфічні обмеження вбудованих обчислювальних систем.

#### Список використаних джерел

1. Arpaci-Dusseau R. H., Arpaci-Dusseau A. C. Operating Systems: Three Easy Pieces. Version 1.10. Arpaci-Dusseau Books, 2023. 714 p.
2. Chakraborty P. Operating Systems: Evolutionary Concepts and Modern Design Principles. Boca Raton: CRC Press, 2024. 618 p.
3. Lacamera D. Embedded Systems Architecture. 2nd ed. Birmingham: Packt Publishing, 2023. 342 p.
4. Lindgren P., Dzialo P., Lunnikivi H., Ericsson J. ENEST – Efficient Interrupt Nesting for RISC-V based CPUs. 2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON). IEEE, 2023. P. 1-6.
5. Yoo T., Choi B. W. Real-Time Performance Benchmarking of RISC-V Architecture: Implementation and Verification on an EtherCAT-Based Robotic Control System. Electronics. 2024. Vol. 13, No 4. P. 733.