

УДК 004.031.6

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОДУКТИВНОСТІ ПРОГРАМНОГО СТЕКУ В ЕМУЛЯТОРІ QEMU ТА НА РЕАЛЬНОМУ RASPBERRY PI

Галкін П.В., Ємельянов Д.Д.

e-mail: denys.iemelianov@nure.ua

Харківський національний університет радіоелектроніки, каф. ПЕЕА
м. Харків, Україна

Raspberry Pi is a popular platform for embedded systems and IoT solutions, widely used in development and education. However, testing software without a physical device can be difficult. For this, the QEMU emulator is often used, which allows you to run Raspberry Pi software in a virtual environment. Despite the advantages of such a technique, it is important to evaluate how accurately it reproduces real code execution and interaction with hardware components. This article analyzes the performance and accuracy of emulation for various software and hardware operations.

Raspberry Pi є популярною платформою для вбудованих систем[1] та IoT-рішень[2], що широко використовується у розробці[3] та навчанні[4]. Однак тестування програмного забезпечення без фізичного пристрою може бути складним[5]. Для цього часто використовують емулятор QEMU, який дозволяє запускати програмне забезпечення Raspberry Pi у віртуальному середовищі. Незважаючи на переваги такої методики, важливо оцінити, наскільки точним є відтворення реального виконання коду та взаємодії з апаратними компонентами. У цій статті проведено аналіз продуктивності та точності емуляції для різних програмних та апаратних операцій.

У цій статті розглянуто питання порівняльного аналізу продуктивності програмного стеку при роботі у віртуальному середовищі QEMU та на фізичному пристрої Raspberry Pi. Проведено вимірювання продуктивності виконання обчислювальних задач, операцій введення/виведення, а також оцінку точності емуляції апаратних ресурсів, таких як GPIO, UART та I2C. Проаналізовано переваги та обмеження використання QEMU для тестування програмного забезпечення для Raspberry Pi.

На рисунку 1 приведений графік порівняння швидкості роботи периферійних пристроїв. Як видно, реальний пристрій забезпечує значно нижчі затримки, особливо для GPIO та I2C, що є критичним для вбудованих систем. Графік демонструє порівняння затримок (у мілісекундах) для трьох основних периферійних інтерфейсів: GPIO, UART, I2C, працюючих у середовищі QEMU та на реальному Raspberry Pi. Очікувано, емуляція у QEMU має значно більші затримки порівняно з фізичним пристроєм через накладні витрати емуляції та відсутність апаратної оптимізації. Емуляція периферійних пристроїв у QEMU значно поступається реальному Raspberry Pi через відсутність повної апаратної підтримки та високу затримку обробки сигналів. GPIO, UART та I2C у QEMU працюють повільніше через

програмну обробку замість апаратної оптимізації, що особливо критично для систем реального часу. Хоча QEMU є зручним для загального тестування програмного коду, для точного вимірювання продуктивності периферії необхідне тестування безпосередньо на фізичному пристрої.

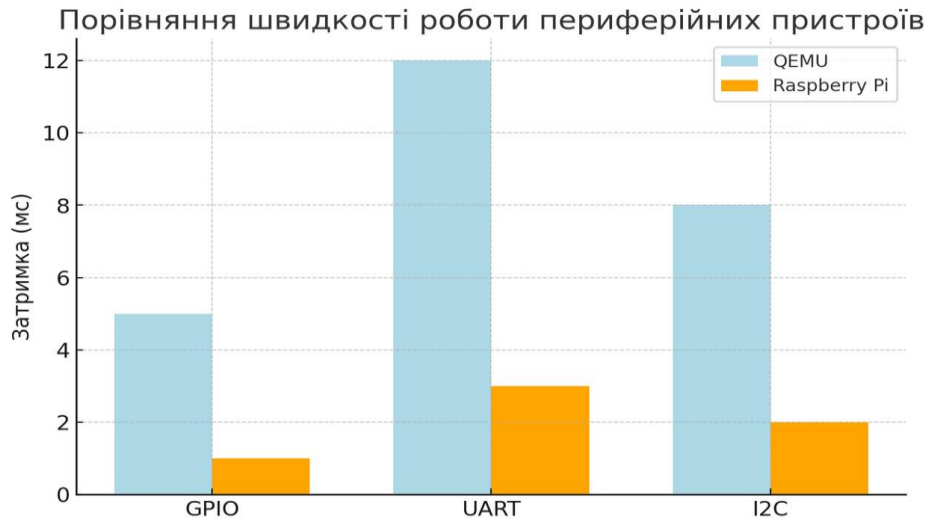


Рисунок 1 – Графік порівняння швидкості роботи периферійних пристроїв

На рисунку 2 приведений графік порівняння продуктивності обчислень на Python та C.

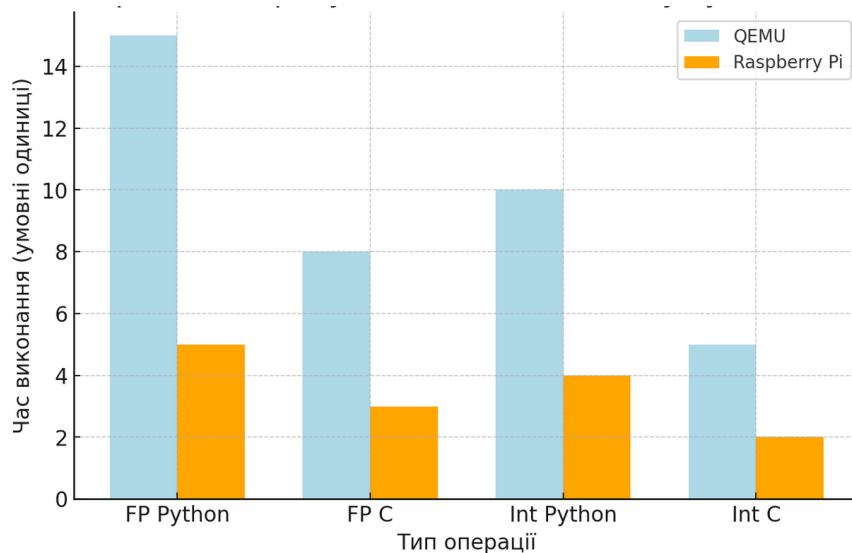


Рисунок 2 – Графік порівняння продуктивності обчислень на Python та C.

Як видно з рисунку 2, графік порівнює продуктивність операцій з плаваючою комою (FP) та цілочисельних обчислень (Int) у Python та C на QEMU та реальному Raspberry Pi. Як видно, реальний пристрій працює значно швидше завдяки оптимізованому виконанню коду та апаратному прискоренню. Реальний Raspberry Pi значно перевершує QEMU у виконанні обчислювальних операцій, особливо для операцій з плаваючою комою,

завдяки апаратній оптимізації. Виконання коду на C є швидшим за Python на обох платформах, але різниця у продуктивності між QEMU та Raspberry Pi залишається суттєвою через накладні витрати емуляції. QEMU може бути корисним для загального тестування алгоритмів, але для точного аналізу продуктивності необхідне виконання на фізичному пристрої.

Порівняльний аналіз показав, що QEMU є корисним інструментом для тестування загальної логіки програмного забезпечення для Raspberry Pi, проте має значні обмеження у точності емуляції апаратних ресурсів.

Обчислювальна продуктивність у QEMU поступається реальному Raspberry Pi через накладні витрати на емуляцію. Операції введення/виведення демонструють непередбачувану продуктивність у QEMU, що може впливати на тестування файлових операцій. Емуляція GPIO, UART та I2C у QEMU є неповною, що ускладнює тестування програм, що взаємодіють з апаратним забезпеченням. У майбутньому можливе використання комбінованого підходу: первинне тестування програмного коду у QEMU з подальшим доопрацюванням та валідацією на фізичному Raspberry Pi. Це дозволить оптимізувати процес розробки та мінімізувати необхідність у постійному використанні фізичного пристрою.

Список використаних джерел:

1. Данильченко М. А. Лабораторний стенд для проектування та управління хмарними ІТ-технологіями. – 2018. 101 с.
2. Балабан В. В. Навчальний комплекс для розроблення IoT пристроїв на базі мікроконтролерів. – 2021.
3. Конорєв О. Ю. Удосконалення захищеності ядра Linux на платформі ARM. – 2018.
4. Галкін П. В. Розробка лабораторного комплексу по вивченню вбудованих систем управління і промислової автоматизації // Матеріали 21-го Міжнародного молодіжного форуму "Радіоелектроніка та молодь у XXI столітті", 25-27 квітня 2017 р. [Текст] : [збірник] / П.В. Галкін // Т. 2 : Конференція "Автоматизовані системи та комп'ютеризовані технології радіоелектронного приладобудування".- Т. 2.- Харків: ХНУРЕ.- С.94-95.
5. Galkin P. Development of Testbenches Base on STM32 and CC253X Microcontrollers / P. Galkin // Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA'2022) : proceedings of the IV International Scientific and Practical Conference. - Kharkiv, 2022. - P. 32-34.