

Операционные системы для работы с интернетом вещей

Чистяков Алексей Сергеевич

Российский экономический университет имени Г. В. Плеханова

студент

Аннотация

В статье рассматриваются параметры, которым должна соответствовать операционная система интернета вещей, а также приведены технические детали работы подобных операционных систем, включая их категоризацию по использованию ресурсов устройства, а также применение в них определенных технологий разработки операционных систем.

Ключевые слова: операционная система, интернет вещей, архитектура ядра

Operating systems for devices of the Internet of things

Chistjakov Aleksej Sergeevich

Plekhanov Russian University of Economics

student

Abstract

The article describes the parameters that the operating system of the Internet of things should match, as well as the technical details of the operation of such operating systems, including their categorization based on the use of device resources, and the use of certain operating system development technologies.

Keywords: operating system, Internet of things, kernel design

Актуальность проблемы

В настоящее время наблюдается активное проникновение решений на базе интернета вещей в самые разные сферы жизни общества, даже в такие консервативные сферы, как банковский, военный и государственный сектора экономики. Стремительно растущее количество устройств интернета вещей и их повсеместное использование поднимает вопрос стандартизации требований к операционным системам для устройств интернета вещей как в части функционала, так и в части обеспечения безопасности работы устройств.

В рамках данной статьи будут рассмотрены особенности, которыми должна обладать операционная система (ОС), чтобы ее можно было назвать операционной системой для работы с интернетом вещей (ОС ИВ).

Нужна ли операционная система для устройств интернета вещей?

Прежде чем определять требования для подобной ОС, нужно разобраться в том, действительно ли есть необходимость во внедрении ОС ИВ в решения, которые ранее обходились без ОС ИВ?

Несомненно, использование ОС для работы устройства не является необходимостью, но, в то же время, устройства постоянно совершенствуются и усложняются: добавляются новые сенсоры и датчики, возрастает количество данных, которые устройство должно передавать, добавляются новые требования к поддержке способов передачи данных во внешнюю среду, и т.д. Даже обычные датчики все чаще получают графический пользовательский интерфейс, а в некоторых случаях и продвинутые технологии распознавания, например, лица, или голоса. Благодаря снижению цен на микроэлектронные комплектующие, появляется возможность создания 32-битных архитектур по ценам, сопоставимым с 16-битными или даже 8-битными архитектурами. В контексте происходящих технологических изменений, использование ОС (в особенности, ОС реального времени, ОСРВ), позволяет заметно облегчить работу разработчиков и интеграторов ПО, так как аппаратные изменения переносятся из зоны ответственности ПО в зону ответственности ОС [1].

В целом, считается, что использование ОС ИВ оправдано в любых устройствах, у которых имеется 16 или больше килобайт оперативной или постоянной памяти. В таких устройствах использование единого цикла событий может привести к сильной задержке в реагировании устройства на изменения во внешней и внутренней среде, в то время как использование ОСРВ нивелирует данный недостаток [1].

Какими параметрами должна обладать операционная система интернета вещей?

Определив, в каких случаях использование ОС ИВ оправданно, можно перейти к определению параметров, которыми должна обладать ОС, чтобы ее можно было назвать ОС ИВ:

- **Размер.** Устройства интернета вещей, как правило, располагают крайне небольшим количеством вычислительных ресурсов, поэтому ОС ИВ должна занимать как можно меньше места не только на постоянном запоминающем устройстве (ПЗУ), но и на оперативном запоминающем устройстве (ОЗУ).
- **Масштабируемость.** ОС ИВ должна быть масштабируемой для любого типа устройств.
- **Переносимость.** ОС ИВ должна абстрагировать аппаратную составляющую устройств, чтобы программная составляющая не зависела от специфик определенных аппаратных компонентов.
- **Модульность.** В ядро ОС ИВ должны включаться только необходимые для работы приложений компоненты. Все остальные расширения должны подключаться как модули при необходимости.

- **Поддержка различных протоколов связи.** ОС ИВ должна поддерживать популярные протоколы связи, такие как thernet, Wi-Fi, BLE, IEEE 802.15.4 и другие.
- **Безопасность.** ОС ИВ должна поддерживать технологию Secure Boot, иметь встроенную криптографическую библиотеку с поддержкой SSL и соответствующие низкоуровневые драйвера [2]
- **Надежность.** Этот параметр важен для устройств, которые используются в критически важных системах или в удаленных устройствах, которые должны работать годами без необходимости вмешательства оператора.

Конечно же, ОС ИВ необязательно должна обладать всеми вышеперечисленными качествами – в основном, требования к ней зависят от устройства и его программного обеспечения. Так, для потребительской микроэлектроники важен малый размер ОС ИВ, а надежность требуется в устройствах, используемых в промышленности [2].

Какие сертификации существуют для операционных систем интернета вещей?

Устройства, используемые в критически важных системах, требуют от ОС ИВ наличия определенной сертификации. Основные широко распространенные сертификации включают в себя [3]:

- DO-178B (для устройств, используемых в авиапромышленности);
- IEC 61508 (для промышленных систем управления);
- ISO 62304 (для медицинских устройств) ;
- SIL3/SIL4 IEC (для устройств, используемых в ядерной промышленности).

Существующие операционные системы интернета вещей с открытым исходным кодом

Наиболее популярные ОС ИВ с открытым исходным кодом:

- TinyOS (одна из наиболее старых ОС ИВ) ;
- RIOT (относительно новая система с активным сообществом разработчиков) [4] ;
- Contiki (одна из наиболее старых ОС ИВ);
- Mantis OS;
- Nano RK;
- LiteOS (разрабатывается и поддерживается компанией Huawei) [5];
- FreeRTOS (активно увеличивает долю на рынке) [4];
- Apache Mynewt [6];
- Zephyr OS (защищена от атак на этапе компиляции) [6];
- Ubuntu Core (модули и приложения подключаются как пакеты) [7];
- ARM mbed (имеет малый размер и поддержку большого количества протоколов связи) [5];

- Android Things (разрабатывается и поддерживается компанией Google);
- Yocto (является платформой для создания собственного дистрибутива ОС ИВ) [7];
- Raspbian.

Существующие операционные системы интернета вещей с закрытым исходным кодом

Наиболее популярные ОС ИВ с закрытым исходным кодом:

- Windows 10 IoT (имеет несколько редакций, в т.ч. бесплатную для использования, поддерживается интеграция с облачными службами) [8];
- WindRiver VxWorks;
- Micrium μ C/OS;
- Micro Digital SMX RTOS;
- MicroEJ OS;
- Express Logic ThreadX;
- TI RTOS (используется в соответствующих микроконтроллерах) ;
- Freescale MQX (используется в соответствующих микроконтроллерах) ;
- Mentor Graphics Nucleus RTOS;
- Green Hills Integrity;
- Particle (поддерживается интеграция с облачными службами).

В целом, для систем, где важна безопасность и надежность работы устройств (например, в аэрокосмической и автомобильной промышленности и медицине), рекомендуется использовать коммерческие варианты ОС ИВ [9].

Должна ли операционная система интернета вещей являться операционной системой реального времени?

Операционная система реального времени (ОСРВ) – это ОС, которая способна обеспечить требуемый уровень сервиса в определенный промежуток времени. Иными словами, такая ОС должна решать возложенные на нее задачи не более чем за установленный промежуток времени.

Использование ОСРВ оправдано только в устройствах, которые должны обрабатывать несколько потоков (например, параллельно получать данные с нескольких сенсоров) и которые имеют установленный лимит на длительность операции (например, распознавание информации с датчика). В других устройствах, где некритична длительность выполнения операций, а также стабильность коммуникации с внешней средой, разумнее будет использовать простую ОС [9]. Тем не менее, в последнее время именно поддержка ОС ИВ выполнения инструкций в реальном времени является ключевым фактором в выборе ОС для устройства [4].

В качестве промежуточного варианта используются гибридные устройства, где запущено одновременно две системы: обычная ОС ИВ для сбора информации с датчиков и сенсоров, работающая, как правило, на 8-битном микроконтроллере, и ОС РВ, которая обеспечивает остальную функциональность устройства, и работает, как правило, на 32-битном процессоре [3].

Типичные требования к памяти со стороны операционных систем интернета вещей

Как было отмечено ранее, одна из особенностей ОС ИВ – это крайне низкие требования к памяти устройства, как ПЗУ, так и ОЗУ. Их можно разделить на несколько категорий, в зависимости от памяти, которая требуется для их функционирования:

- Низкие требования. К таким системам относятся Contiki (2Кб ОЗУ и 40Кб ПЗУ), Mantis и Nano RK [10], а также Apache Mynewt (8Кб ОЗУ и 64Кб ПЗУ) ;
- Средние требования. SMX RTOS (40Кб ОЗУ и 70Кб ПЗУ) относится к этой категории;
- Высокие требования. К данной категории относятся Ubuntu Core, Windows 10 IoT, Android Things и MicroEJ (потребление ОЗУ и ПЗУ более 100Мб) [2].

Первые две категории, как правило, используются в сенсорах [2], в то время как третья категория применяется в более сложных устройствах.

Какие архитектуры применяются в операционных системах интернета вещей?

От архитектуры конкретной ОС ИВ сильно зависит ее функционал и требования к памяти. В сфере ОС ИВ стоит выделить следующие тренды разработки:

- TinyOS и ARM mbed используют монолитную архитектуру, в то время как разработчики RIOT и FreeRTOS сделали выбор в пользу микроядра.
- ARM mbed выполняется в единственном потоке и реализует событийно-ориентированную модель программирования, а RIOT, FreeRTOS и μ C/OS поддерживают мультипоточность [2].
- В статических системах (TinyOS, Nano RK) все ресурсы выделяются на этапе компиляции. В динамических системах ресурсы выделяются по мере необходимости, но из-за этого они имеют более высокие накладные расходы.
- Простейшие ОС ИВ не поддерживают файловые системы вообще, либо реализуют одноуровневую файловую систему.
- В большинство ОС ИВ встроены алгоритмы оптимизации потребления энергии, как для процессора, так и для периферийных устройств.

Выводы

В рамках данной статьи были рассмотрены причины, по которым использование ОС ИВ растет даже в сфере устройств, которые ранее работали вообще без наличия ОС, а также отмечены плюсы и минусы использования ОС ИВ в подобных устройствах. Кроме того, были выделены параметры, характерные для ОС ИВ, и сферы, в которых наличие определенных параметров является наиболее востребованным. Наконец, были приведены примеры ОС ИВ, представленных на рынке, как с открытым, так и с закрытым исходным кодом, а также описаны технические особенности таких ОС в рамках потребления памяти и выделены тренды, используемые в разработке ОС ИВ.

Библиографический список

1. The Benefits of RTOSes in the Embedded IoT // Eetimes. 2015. URL: https://www.eetimes.com/author.asp?section_id=36&doc_id=1327623
2. IoT Operating Systems // Arrow. 2016. URL: <https://www.arrow.com/en/research-and-events/articles/iot-operating-systems>
3. Part 5: The Internet of Things and the RTOS // Micrium. 2017. URL: <https://www.micrium.com/iot/iot-rtos/>
4. Embedded Markets Study // UBM. 2015. URL: <https://devzone.nordicsemi.com/attachment/a61052ff4978f8c42b4f6f4b11a1b0e0>
5. List of Operating Systems for IoT that You Should Know About // iotleague. 2016. URL: <http://www.iotleague.com/list-of-operating-systems-for-iot-that-you-should-know-about/>
6. 7 open source IoT operating systems That Are Democratizing The IoT Space // electronicsforu. 2017. URL: <http://electronicsforu.com/resources/7-open-source-iot-operating-systems-democratizing-iot-space>
7. Operating Systems for IoT Embedded Systems // webofthings. 2016. URL: <https://webofthings.org/2016/12/12/iot-os-embedded/>
8. Windows 10 IoT editions explained // perficient. 2016. URL: <https://blogs.perficient.com/microsoft/2016/01/windows-10-iot-editions-explained/>
9. 8 IoT Operating Systems Powering The Future // informationweek. 2016. URL: <https://www.informationweek.com/iot/8-iot-operating-systems-powering-the-future/d/d-id/1324464>
10. IoT Software Development Guide // postscapes. 2017. URL: <https://www.postscapes.com/internet-of-things-software-guide/#embedded>