



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
НА ТЕМУ:**

«Сетевые операционные системы»

Студент ИУ7-53Б

_____ Сапожков А. М.

Руководитель

_____ Строганов Ю. В.

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-7
(Индекс)

_____ И. В. Рудаков
(И.О.Фамилия)

«16» сентября 2022 г.

ЗАДАНИЕ
на выполнение научно-исследовательской работы

по теме

«Сетевые операционные системы»

Студент группы **ИУ7-53Б**

Сапожков Андрей Максимович

Направленность НИР

учебная

Источник тематики

НИР кафедры

График выполнения НИР: 25% к 6 нед., 50% к 9 нед., 75% к 12 нед., 100% к 15 нед.

Техническое задание

Провести обзор существующих операционных систем для устройств интернета вещей. Провести анализ предметной области интернета вещей, сформулировать критерии сравнения и оценки рассмотренных операционных систем. Классифицировать существующие решения по выделенным критериям.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на **12-25** листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т. п.)

Презентация на **6-10** слайдах.

Дата выдачи задания «16» сентября 2022 г.

Руководитель НИР

Студент

(Подпись, дата)

(Подпись, дата)

Ю. В. Строганов
(И.О.Фамилия)

А. М. Сапожков
(И.О.Фамилия)

Содержание

Введение	4
1 Анализ предметной области	5
1.1 Основные определения	5
1.2 Архитектура интернета вещей	6
1.3 Проблема выбора операционной системы для интернета вещей	8
2 Описание существующих решений	10
2.1 Операционные системы реального времени	10
2.1.1 Azure RTOS	10
2.1.2 (Amazon) FreeRTOS	10
2.1.3 Zephyr	10
2.1.4 OCPB МАКС	10
2.2 Операционные системы разделения времени	10
2.2.1 Windows 10 IoT	10
2.2.2 Azure Sphere	10
2.2.3 Azure IoT Edge	10
2.2.4 Azure Sphere	10
2.2.5 Contiki	11
2.2.6 Mbed OS	11
2.2.7 Android Things	11
2.2.8 Huawei LightOS	11
2.2.9 TinyOS	11
2.2.10 Ubuntu Core	11
2.2.11 Raspbian	11
2.2.12 RIOT	11
2.2.13 Fuchsia OS	11
3 Классификация существующих решений	12
3.1 Критерии сравнения операционных систем	12
3.2 Сравнительный анализ операционных систем	12
Заключение	13
Список использованных источников	14

Введение

В настоящий момент все наблюдаемые формы коммуникаций сводятся к схемам человек-человек человек-устройство. Но интернет вещей (Internet of Things, IoT) предлагает нам колоссальное интернет-будущее, в котором появятся коммуникации типа машина-машина (Machine-to-Machine, M2M). Это дает возможность для объединения всех коммуникаций в общую инфраструктуру, позволяя не только управлять всем, что находится вокруг нас, но и предоставляя информацию о состоянии этих вещей. [1]

Традиционная модель интернета вещей заключается в том, что устройство локально собирает данные, а затем отправляет их в облачный сервис. Там эти данные обрабатываются с помощью искусственного интеллекта или машинного обучения для получения бизнес-аналитики. Главная задача интернета вещей [2] — создание среды, в которой вещи имеют способность слушаться управления, а данные о них могут быть получены и обработаны для выполнения желаемой задачи посредством обучения устройств. Эта концепция позволяет не только объединять предметы материального мира посредством Интернета для обмена информацией между ними, но и развивать возможности по накоплению, структурированию и анализу различной информации о поведении людей в городском пространстве, дома и на работе.

В основе большинства решений в области интернета вещей лежит устройство, которое подключается к облачным сервисам для обмена данными. Обработка данных может выполняться либо на самом устройстве, либо на удалённом сервере. Возможности для разработки таких решений обеспечивает операционная система. При выборе подходящей операционной системы для устройства интернета вещей разработчик должен убедиться, что она совместима с необходимым оборудованием, приложениями и требованиями к подключению. Большое разнообразие операционных систем позволяет выбрать подходящий вариант в зависимости от особенностей конкретной задачи.

Целью данной работы является изучение существующих операционных систем для устройств интернета вещей. Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) провести анализ предметной области интернета вещей;
- 2) рассмотреть существующие операционные системы для устройств интернета вещей;
- 3) сформулировать критерии сравнения и оценки рассмотренных операционных систем;
- 4) провести сравнительный анализ существующих решений по выделенным критериям.

1 Анализ предметной области

В данном разделе будут представлены основные определения, а также будет описана архитектура интернета вещей и поставлена проблема выбора операционной системы для устройств интернета вещей.

1.1 Основные определения

Интернет вещей [3] — это система взаимосвязанных вычислительных устройств, которые могут собирать и передавать данные по беспроводной сети без участия человека. В более широком смысле интернет вещей можно определить как концепцию, описывающую сеть физических объектов ("вещей"), в которые встроены датчики, программное обеспечение и другие технологии для обмена данными с другими устройствами и системами через интернет. [4]

Операционная система компьютера [5] — комплекс взаимосвязанных программ, который действует как интерфейс между приложениями и пользователями, с одной стороны, и аппаратурой компьютера, с другой стороны.

Сетевая операционная система [5] позволяет пользователю работать со своим компьютером как с автономным и добавляет к этому возможность доступа к информационным и аппаратным ресурсам других компьютеров сети. В идеальном случае сетевая ОС должна представить пользователю сетевые ресурсы не в виде сети, а в виде ресурсов единой централизованной виртуальной машины. Для такой операционной системы используют специальное название — **распределённая ОС**, или **истинно распределённая ОС**.

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (ОС ЕС);
- системы разделения времени (ОС семейств Linux, Windows, MacOS);
- системы реального времени (QNX, FreeRTOS).

Современные операционные системы, проектируемые для устройств интернета вещей, являются либо системами реального времени, либо системами разделения времени. **Операционная система реального времени** [5] — это система, предназначенная для управления физическими объектами (процессами), которая способна *обеспечить предсказуемое время реакции* в ответ на изменение состояния управляемого объекта (процесса). **Система разделения времени** [5] — такая форма организации вычислительного процесса, при которой сразу несколько пользователей одновременно работают на компьютере, причём каждому из них кажется, что он получил компьютер в полное своё распоряжение. Главной целью и критерием эффективно-

сти систем разделения времени является *обеспечение удобства и эффективности работы пользователей*.

Ввиду большого многообразия устройств, используемых , при выборе ОС для устройств интернета вещей большую роль играет переносимость. ОС называют **переносимой** (portable) [5], или **мобильной**, если её код может быть сравнительно легко перенесён с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на аппаратную платформу другого типа. Мобильность - не бинарное состояние, а понятие степени.

Наиболее общим подходом к структуризации операционной системы является разделение всех её модулей на две группы: модули ядра, выполняющие основные функции ОС, и вспомогательные модули ОС. Ядро ОС [5] представляет собой сложный многофункциональный комплекс, имеющий многослойную структуру. В развитии современных операционных систем наблюдается тенденция в сторону переноса кода в верхние слои и удалении при этом всего, что только возможно, из режима ядра, оставляя минимальное *микроядро* [6]. Обычно это осуществляется переключением выполнения большинства задач операционной системы на средства пользовательских процессов. Такой подход способствует переносимости, расширяемости, повышению надёжности системы и создаёт хорошие предпосылки для поддержки распределённых приложений. Модель с микроядром хорошо подходит для поддержки **распределённых вычислений**, так как использует механизмы, аналогичные сетевым: взаимодействие клиентов и серверов путём обмена сообщениями. Серверы микроядерной ОС могут работать как ан одном, так и на разных компьютерах. В этом случае при получении сообщения от приложения микроядро может обработать его самостоятельно и передать локальному серверу или же переслать по сети микроядру, работающему на другом компьютере. Переход к распределённой обработке требует минимальных изменений в работе операционной системы — просто локальный транспорт заменяется сетевым. В контексте интернета вещей эта концепция приобретает очень большое значение, находя своё применение в разработке распределённых вычислительных систем. [5]

1.2 Архитектура интернета вещей

Определение архитектуры интернета вещей является предметом серьезных дискуссий. Однако все согласны с тем, что для того, чтобы концепция интернета вещей работала, она должна состоять из сети, датчиков и коммуникаций. Одной из наиболее подробных архитектур является семиуровневая модель, предложенная компанией Cisco [7]. Ниже, в соответствии с рисунком 1.1, описаны её уровни. [8] [9]

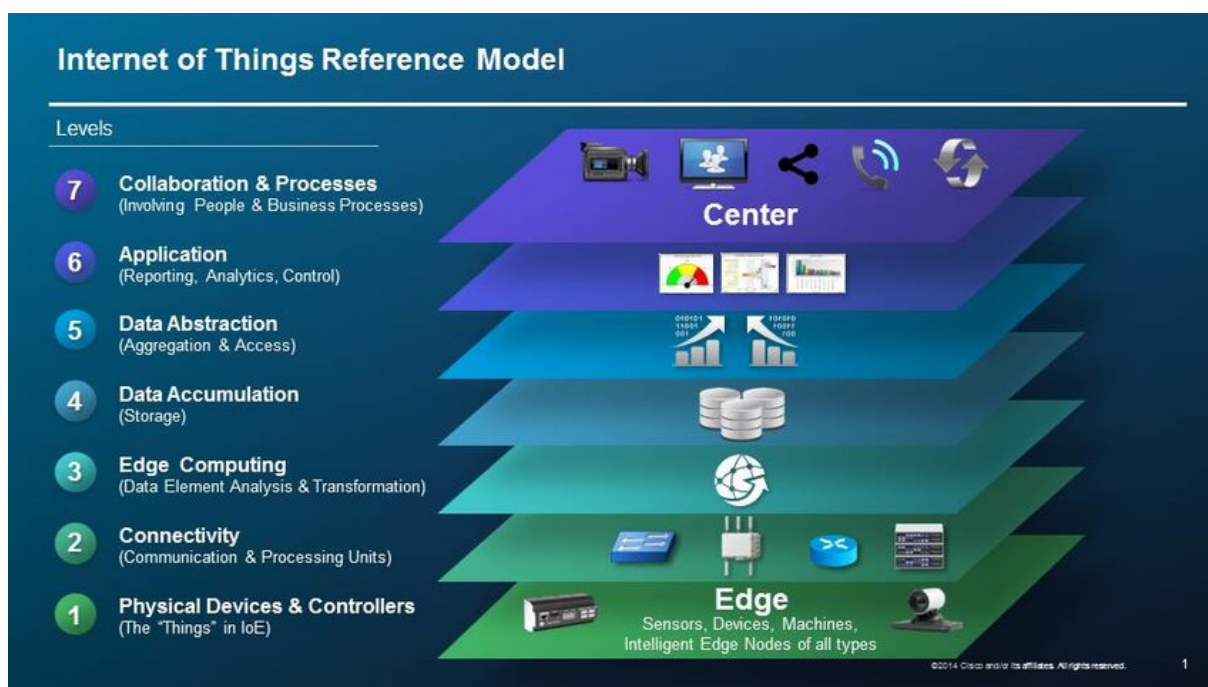


Рисунок 1.1 — Архитектура интернета вещей, предложенная Cisco.

1. **Физические устройства и контроллеры** — это уровень, содержащий вещи в IoT. Сюда входит широкий спектр конечных устройств, которые могут отправлять или получать информацию (например, датчики и считыватели радиочастотной идентификации (RFID)).

2. **Соединение** — это уровень, содержащий все компоненты, способные передавать информацию. Передача может осуществляться между устройствами на первом уровне, между компонентами на этом уровне или между первым и третьим уровнем.

3. **Граничные (туманные) вычисления** — это первый уровень, на котором происходит обработка данных. Здесь могут собираться и обрабатываться большие объемы информации, что снижает нагрузку на более высокие уровни. Данный уровень также позволяет форматировать и декодировать данные до того, как они будут обработаны.

4. **Накопление данных** — это уровень, на котором данные сохраняются, чтобы приложения могли получить к ним доступ в случае необходимости. Как правило, необходимая обработка информации не может быть выполнена на сетевых скоростях, поэтому вычислительная система нуждается в промежуточном хранилище данных. Сохранённые данные также могут быть преобразованы и рекомбинированы, чтобы быть готовыми к использованию на более высоких уровнях. В результате на этом уровне данные в движении преобразуются в данные в состоянии покоя.

5. **Абстракция данных** — это уровень, позволяющий хранить данные более эффективным образом для повышения производительности более высоких уровней. На данном уровне над данными могут выполняться операции нормализации, индекси-

рования, форматирования, проверки, консолидации, а также обеспечивается доступ к нескольким хранилищам данных.

6. Приложения — это уровень, на котором информация, накопленная ранее, интерпретируется приложениями. Именно на этом уровне располагается бизнес-логика приложений.

7. Взаимодействие и процессы — это уровень, объединяющий всё вместе. Система бесполезна, если информация, предоставляемая на этом уровне, не является полезной. Данные из интернета вещей должны использоваться для принятия обоснованных решений.

Операционные системы для интернета вещей существенно упрощают разработку программного обеспечения для систем с многоуровневой архитектурой, описанной выше, так как позволяют разработчикам абстрагироваться от особенностей аппаратуры конкретных устройств и предоставляют шаблоны для создания приложений с той или иной архитектурой.

P.S.: Только попробуйте спросить про единицу измерения простоты! Это не количественная оценка, а КАЧЕСТВЕННАЯ! Здесь может стоять только одно из трёх слов: упрощает, усложняет и не меняет. А упрощает СУЩЕСТВЕННО, потому что ПОЛНОСТЬЮ меняет процесс разработки, а значит, не может влиять в малой степени (тут оценка уже бинарная: меняет либо в большей, либо в меньшей степени).

1.3 Проблема выбора операционной системы для интернета вещей

Операционные системы, предназначенные для интернета вещей, обладают различной функциональностью, которая определяет преимущества и недостатки каждого конкретного решения. Выбранная операционная система должна полностью соответствовать требованиям и ограничениям, предъявляемым к устройствам. Можно выделить четыре основных аспекта и перечень главных вопросов, которые разработчики устройств учитывают при выборе ОС для периферийных устройств IoT. [10]

1. Какой уровень надежности и долгосрочной поддержки необходим?

Под надёжностью в данном случае понимается соответствие операционной системы определенным стандартам и сертификациям. Необходимо, чтобы выбранная операционная система обеспечивала варианты, подходящие для конкретных отраслевых стандартов и требований.

2. Какие требования предъявляются к производительности?

Выбранная операционная система должна соответствовать требованиям устройства к вычислительной мощности и производительности в реальном времени. Также при ответе на данный вопрос стоит учитывать другие аспекты, связанные с производительностью и оказывающие влияние на выбор ОС. К таким можно отнести энер-

гопотребление и объём памяти устройства, а также требования к времени отклика системы на внешние события.

3. Обеспечивает ли операционная система безопасность устройства?

Безопасность — один из основных факторов, учитываемых при разработке устройств интернета вещей. Это касается и используемой ОС, так как от неё зависит целостность данных, собираемых устройствами. Если система не обеспечивает необходимый уровень безопасности, то это может привести к порче или краже данных, нарушению запланированных процессов.

4. Является ли выбранная операционная система масштабируемой?

В связи с тем, что операционные системы, как и любое программное обеспечение, меняются вместе с требованиями к функциям, которые они должны предоставлять. Разработка IoT-устройства с масштабируемой ОС позволяет в будущем адаптироваться к другим задачам без внесения значительных изменений. Масштабируемые ОС могут обрабатывать дополнительные ресурсы без изменения скорости вывода, охватывать несколько устройств и географических регионов.

2 Описание существующих решений

В данном разделе будет проведён анализ существующих операционных систем для устройств интернета вещей. Рассматриваемые операционные системы будут относиться к одному из двух типов: ОС реального времени и ОС разделения времени.

Стараться указывать тип ядра ОС, а также информацию по остальным критериям сравнения.

2.1 Операционные системы реального времени

2.1.1 Azure RTOS

[10]

2.1.2 (Amazon) FreeRTOS

2.1.3 Zephyr

2.1.4 OCPB МАКС

2.2 Операционные системы разделения времени

2.2.1 Windows 10 IoT

3 версии: Windows 10 IoT Core, Windows 10 IoT Enterprise и Windows IoT Server. [10]

2.2.2 Azure Sphere

[10]

2.2.3 Azure IoT Edge

[10]

2.2.4 Azure Sphere

[10]

2.2.5 Contiki

2.2.6 Mbed OS

2.2.7 Android Things

2.2.8 Huawei LightOS

2.2.9 TinyOS

2.2.10 Ubuntu Core

Snappy - OS with Ubuntu Core?

2.2.11 Raspbian

Вроде бы сделана только для Raspberry Py, то есть переносимая.

2.2.12 RIOT

2.2.13 Fuchsia OS

Если этого мало, то здесь (<https://www.intuz.com/top-iot-operating-systems-for-iot-devices>) есть ещё больше.

3 Классификация существующих решений

В данном разделе будут описаны критерии сравнения операционных систем для устройств интернета вещей

3.1 Критерии сравнения операционных систем

Описание (и желательно обоснование) критериев сравнения.

Критерии:

- Тип ядра: монолитное или микроядро;
- Тип лицензии;
- Поддержка POSIX;
- Тип многозадачности: вытесняющая, кооперативная, отсутствует;
- Кроссплатформенность или переносимость (бинарный признак);
- ????? Открытый/закрытый исходный код;
- ????? Поддержка SMP;
- ????? Популярность (востребованность) на рынке (мб взять статистику популярности ОС и сравнивать по объёму доли на рынке в %);
- ????? Промышленное применение (бинарный признак, типа может быть только промышленное и частное применение).

Наличие сетевых функций не оценивается, так как все ОС, рассматриваемые в данной работе, являются сетевыми.

3.2 Сравнительный анализ операционных систем

Сводная таблица сравнения операционных систем.

Если по всем критериям одна из систем оказывается лучшей из описываемых, то задать себе закономерный вопрос, почему остальные системы ещё используются. Предложить ещё один критерий сравнения.

Заключение

Эх, вот если бы я был embedded-разработчиком, то в таких-то проектах ставил бы на свои устройства такую ОС, а в других — другую, потому что потому.

Список использованных источников

1. Довгаль В. А. Довгаль Д. В. Интернет Вещей: концепция, приложения и задачи // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. — 2018. — С. 129–135.
2. В. Маркеева А. Интернет вещей (IoT): возможности и угрозы для современных организаций // Общество: социология, психология, педагогика. — 2016.
3. Что такое интернет вещей? Определение и описание [Электронный ресурс]. — Режим доступа: <https://www.kaspersky.ru/resource-center/definitions/what-is-iot> (дата обращения: 09.11.2022).
4. What is IoT? [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/internet-of-things/what-is-iot/> (дата обращения: 09.11.2022).
5. Олифер В. Г. Олифер Н. А. Сетевые операционные системы. — Питер, 2009. — ISBN: 9785272001207.
6. С. Таненбаум Э. Операционные системы: разработка и реализация. — Питер, 2006. — ISBN: 5-469-00148-2.
7. Cisco - Networking, Cloud, and Cybersecurity Solutions [Электронный ресурс]. — Режим доступа: <https://www.cisco.com/> (дата обращения: 09.11.2022).
8. Battery draining attacks against edge computing nodes in IoT networks / Smith Ryan, Palin Daniel, Ioulianou Philokypros, Vassilakis Vassilios, and Shahandashti Siamak // Cyber-Physical Systems. — 2020. — 01. — Vol. 6. — P. 1–21.
9. Masoodhu Banu N. M. Sujatha C. IoT Architecture a Comparative Study // International Journal of Pure and Applied Mathematics. — 2017. — Vol. 117. — P. 45–49.
10. Руководство по выбору операционной системы для пограничного устройства Интернета вещей [Электронный ресурс]. — Режим доступа: <https://www.quarta-embedded.ru/about/statiyi/rukovodstvo-po-vyboru-operacionnoj-sistemy-dlya-pogranichnogo-ustrojstva-interneta-veshchej/> (дата обращения: 10.11.2022).