

УДК: 004.89

Бакайкина Виктория Геннадиевна,

Ассистент кафедры Программной инженерии

ФГБОУ ВО «Поволжский государственный университет телекоммуникаций

и информатики»

Подопрелов Кирилл Игоревич,

Студент

ФГБОУ ВО «Поволжский государственный университет телекоммуникаций

и информатики»

**АРХИТЕКТУРА ОПЕРАЦИОННОЙ СИСТЕМЫ ДЛЯ РОЯ
НЕОДНОРОДНЫХ МИКРОРОБОТОВ (SMART DUST): ПРОБЛЕМЫ
УПРАВЛЕНИЯ, ПЛАНИРОВАНИЯ И САМООРГАНИЗАЦИИ**

***Аннотация:** В работе исследуется концепция операционной системы для роя неоднородных микророботов, известного как «Smart Dust». Классические парадигмы ОС, ориентированные на централизованное управление стабильными ресурсами, неприменимы в условиях высокой динамичности, ограниченности ресурсов каждого узла и необходимости коллективного поведения. Целью статьи является систематизация требований и предложение эталонной многоуровневой архитектуры SwartOS, которая разделяет абстракции на уровень индивидуального агента, кластера и роя. В работе проанализированы ключевые вызовы: децентрализованное планирование задач, миграция кода и состояния, энергоэффективная коммуникация. Проведено сравнение предлагаемой архитектуры с существующими подходами в области ОС для встраиваемых систем и роевого интеллекта. Результаты показывают, что переход от нода-центричной к рое-центричной модели ОС является необходимым условием для реализации потенциала распределённых микророботических систем.*

Ключевые слова: операционная система для роя, Smart Dust, микророботы, распределённые системы, самоорганизация, планирование задач, миграция состояния.

Bakaikina Victoria Gennadievna,

Assistant Professor of Software Engineering

Volga Region State University of Telecommunications and Informatics

Kirill Igorevich Podoprelov,

Student

Volga Region State University of Telecommunications and Informatics

**THE ARCHITECTURE OF THE OPERATING SYSTEM FOR A
SWARM OF HETEROGENEOUS MICROROBOTS (SMART DUST):
PROBLEMS OF MANAGEMENT, PLANNING AND SELF-
ORGANIZATION**

***Abstract:** The paper explores the concept of an operating system for a swarm of heterogeneous microrobots known as "Smart Dust". Classical operating system paradigms focused on centralized management of stable resources are not applicable in conditions of high dynamism, limited resources of each node and the need for collective behavior. The purpose of the article is to systematize the requirements and propose a reference multi-level architecture of SwarmOS, which divides abstractions into individual agent, cluster and swarm levels. The paper analyzes key challenges: decentralized task planning, code and status migration, and energy-efficient communication. The proposed architecture is compared with existing approaches in the field of OS for embedded systems and swarm intelligence. The results show that the transition from a node-centered to a swarm-centered OS model is a prerequisite for realizing the potential of distributed microrobotic systems.*

Keywords: *swarm operating system, Smart Dust, microrobots, distributed systems, self-organization, task planning, state migration.*

Введение

Современные тенденции миниатюризации и развития беспроводных сенсорных сетей открывают путь к созданию роев из сотен и тысяч микророботов, способных коллективно выполнять задачи в области мониторинга окружающей среды, микро-логистики и ремонта конструкций [1]. Однако отсутствие адекватного системного программного обеспечения является ключевым барьером для практического внедрения таких систем. Каждый микроробот в рое обладает крайне ограниченными вычислительными ресурсами, ёмкостью памяти, запасом энергии и нестабильным каналом связи [2]. Традиционные операционные системы реального времени, такие как FreeRTOS или Zephyr, решают задачи исключительно на уровне отдельного устройства, не предоставляя необходимых абстракций для управления роем как единым, целостным вычислительным ресурсом [3]. Этот концептуальный разрыв требует нового подхода. Целью данной статьи является систематизация фундаментальных требований к операционной системе для роя неоднородных микророботов, предложение эталонной многоуровневой архитектуры и глубокий анализ алгоритмических вызовов в области планирования и коммуникации. Научная новизна работы заключается в синтезе подходов из распределённых операционных систем, роевого интеллекта и энергоэффективных вычислений. Структура статьи включает анализ проблем и требований, описание архитектуры SwarmOS, детальное рассмотрение её ключевых модулей, сравнительный анализ с существующими решениями и заключение с перспективами дальнейших исследований.

Проблемы и требования к ОС для Smart Dust

Управление роем микророботов предъявляет уникальные требования, кардинально отличающиеся от парадигм классических операционных систем.

Первой фундаментальной проблемой является гетерогенность и динамичность среды. Рой состоит из узлов с различными сенсорами, актуаторами и постоянно изменяющимся уровнем энергии, а его топология непрерывно меняется из-за движения роботов и неизбежных отказов аппаратных компонентов [4]. Второй вызов проистекает из крайней ограниченности ресурсов индивидуального узла. Ни один микроробот не способен хранить полную карту роя или выполнять ресурсоёмкие глобальные вычисления, что делает любую централизованную логику управления неприемлемой, превращая её в точку отказа [5]. Третья критическая проблема - энергетические ограничения. Радиосвязь остаётся самым энергозатратным процессом, поэтому ОС должна быть спроектирована для минимизации объёмов передаваемых данных, делая акцент на локальные вычисления и агрегацию информации [6]. Четвёртый аспект связан с моделью программирования. Пользователь формулирует задачу для роя в целом, например, «составить температурную карту области» или «транспортировать объект», а не для отдельных роботов. Следовательно, ОС должна предоставлять механизмы для декомпозиции такой коллективной задачи и её распределения между агентами. Система должна обладать повышенной устойчивостью, сохраняя работоспособность при потере значительной части узлов и адаптивно перераспределяя задачи. Совокупность этих требований формирует необходимость в переходе к принципиально новой, рое-центричной парадигме операционной системы, где абстракция единого виртуального вычислительного ресурса является первичной [7].

Предлагаемая архитектура SwarmOS: многоуровневая модель

Для удовлетворения сформулированных требований предлагается архитектура SwarmOS, построенная на трёх логических уровнях, обеспечивающих чёткое разделение ответственности. Первый уровень агента, или Micro-Kernel Layer. Это базовая ОС, функционирующая на каждом отдельном микророботе которая содержит в себе задачу обеспечить изоляцию выполняемых задач, прямой доступ к сенсорам и актуаторам, управление

энергией и предоставление примитивов для связи. Ядро на этом уровне должно быть минимальным, по сути являясь микроядром или библиотечной ОС. Его ключевая функция это безопасное исполнение капсул кода, или мобильных агентов, которые получают от вышележащих уровней системы [8]. Второй уровень – это уровень кластера, или Emergent Middleware. Он формируется динамически группой географически близких роботов и реализует сервисы, невозможные на одном узле. К ним относятся распределённое хранение фрагментов состояния задачи, достижение локального консенсуса для принятия решений и агрегация данных от множества сенсоров. Для взаимодействия на этом уровне используются алгоритмы gossip-типа, позволяющие эффективно обмениваться метаданными без центрального координатора [9]. Третий, высший уровень - уровень роя, или Swarm Abstraction Layer. Это виртуальный слой, представляющий весь рой как единое целое. Он отвечает за приём высокоуровневых задач, описанных на специализированном языке или в виде графа целей, производит их декомпозицию и инициирует создание необходимых капсул кода. Важно отметить, что этот уровень не имеет фиксированного места исполнения; его логика может выполняться на любом достаточно мощном узле роя или на внешней базовой станции, что обеспечивает отказоустойчивость [10]. Взаимодействие между уровнями и поток данных в системе SwarmOS иллюстрирует Рисунок 1.



Рисунок 1. Архитектура операционной системы SwarmOS и потоки данных

Ключевые алгоритмические модули SwarmOS

Реализация предложенной архитектуры требует разработки специализированных алгоритмических модулей. Одним из центральных является модуль децентрализованного планирования на основе рынка ресурсов. В этой модели общая задача представляется как набор подзадач с чёткими требованиями к ресурсам, таким как тип сенсора или вычислительная мощность. Каждый узел роя непрерывно оценивает свою «цену» за выполнение той или иной подзадачи, основываясь на текущем уровне энергии, загруженности и географической близости к цели. Подзадачи в результате конкурентного процесса «покупаются» теми узлами, которые предлагают наилучшее соотношение цены и эффективности. Такой рыночный подход обеспечивает оптимальное распределение работы без единого центрального планировщика [11]. Другим критически важным модулем является механизм миграции состояния и капсул кода. Для балансировки нагрузки и обхода отказавших узлов капсула кода вместе с минимально необходимым контекстом состояния должна иметь возможность мигрировать с одного робота на другой.

Чтобы минимизировать накладные расходы на передачу данных, уровень кластера поддерживает распределённый журнал изменений состояния задачи, что позволяет пересылать лишь дифференциальные обновления и недостающие фрагменты кода [12]. Третий ключевой модуль отвечает за энергоэффективную коммуникацию через виртуальную роевую шину данных. Вместо неэффективного прямого обмена сообщениями между всеми узлами используется модель распространения информации на основе протоколов типа управляемого слуха или наводнения с контролем времени жизни. Данные распространяются в рое волнообразно, а применение нецентрализованных фильтров Блума на каждом узле позволяет эффективно избегать дублирования информации и снижать общий сетевой трафик [13].

Сравнительный анализ

Анализ существующих подходов демонстрирует, что ни одна из известных систем не удовлетворяет полностью требованиям управления высокодинамичным роём микророботов. Так, кластерные ОС, например Kubernetes, рассчитаны на стабильную сеть и централизованное управление, что делает их неприменимыми для распределённых систем с мобильными узлами [14]. Операционные системы для беспроводных сенсорных сетей, включая TinyOS и Contiki, оптимизированы под статические или малоподвижные сенсорные узлы и не поддерживают коллективную мобильность и распределённое выполнение сложных задач [15]. Роевые алгоритмы, основанные на поведении стаи, предоставляют полезные модели самоорганизации, но не решают системные задачи изоляции кода, управления ресурсами и формализации высокоуровневых задач [16]. Предложенная концепция SwarmOS объединяет принципы самоорганизации роевого интеллекта с системными абстракциями, формализованными через API операционной системы. В частности, ключевое отличие заключается в ориентации на рой как на единый виртуальный вычислительный ресурс, в отличие от классических RTOS, ориентированных на отдельный узел. В Таблице 1 представлено сравнение характеристик уровня агента SwarmOS с

классической RTOS, что наглядно демонстрирует изменения парадигмы управления и взаимодействия:

Таблица 1.

Сравнение уровня агента SwarmOS и классической RTOS

Характеристика	Классическая RTOS (FreeRTOS)	Micro-Kernel Layer (SwarmOS)
Единица планирования	Задача/поток	Капсула кода (мобильный агент)
Приоритет управления	Дедлайны отдельного устройства	Глобальная цель роя и уровень энергии узла
Межпроцессное взаимодействие	Очереди, семафоры (внутри устройства)	Асинхронные сообщения (роевая шина)
Целевая модель	Стабильное исполнение	Динамическая миграция и остановка по требованию системы

Таблица дает точно понять, что SwarmOS обеспечивает переход к роецентричной модели управления: акцент смещается с локальных задач отдельных узлов на глобальные цели роя, поддерживается миграция мобильных агентов и эффективная асинхронная коммуникация между узлами.

Заключение

В статье была обоснована необходимость создания и предложена архитектура специализированной операционной системы SwarmOS для управления роем неоднородных микророботов. Ключевым концептуальным отличием является переход от парадигмы управления отдельными устройствами к парадигме программирования роя как виртуального, динамически реконфигурируемого вычислительного субстрата. Основные

научно-технические результаты работы включают: формализацию требований к рое-центричной ОС, разработку многоуровневой модели абстракций, а также описание ключевых алгоритмических модулей для децентрализованного планирования и энергоэффективной коммуникации. Практическая реализация предложенных принципов открывает путь к созданию нового класса автономных, масштабируемых и устойчивых к сбоям микророботических систем для широкого спектра приложений.

Список использованных источников:

1. Warneke B., Last M., Liebowitz B., Pister K. S. J. Smart Dust: Communicating with a cubic-millimeter computer [Электронный ресурс] // Computer. – 2001. – Vol. 34, no. 1. – Режим доступа: <https://ieeexplore.ieee.org/document/895117> (свободный). Заглавие с экрана. – Дата обращения: 12.12.2025.
2. Rahimi M. et al. Mobile Robotic Sensors for Perimeter Detection and Tracking [Электронный ресурс] // Proceedings of the ISA/IEEE Sensors for Industry Conference. – 2003. – Режим доступа: <https://ieeexplore.ieee.org/document/1245086> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.
3. Levis P., Culler D. Maté: A Tiny Virtual Machine for Sensor Networks [Электронный ресурс] // Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X). – 2002. – Режим доступа: <https://dl.acm.org/doi/10.1145/605397.605424> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.
4. Kornienko S. et al. Towards an Autonomous Micro-Robot Using Chemical Propulsion [Электронный ресурс] // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). – 2005. – Режим

доступа: <https://ieeexplore.ieee.org/document/1570648>(свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

5. Mamei M., Zambonelli F. Programming Pervasive and Mobile Computing Applications: The TOTA Approach [Электронный ресурс] // ACM Transactions on Software Engineering and Methodology (TOSEM). – 2009. – Vol. 18, no. 4. – Режим доступа: <https://dl.acm.org/doi/10.1145/1538942.1538945> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

6. Karp V., Kung H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks [Электронный ресурс] // Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom). – 2000. – Режим доступа: <https://dl.acm.org/doi/10.1145/345910.345953> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

7. Nagpal R. Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Degenerate Nanotechnology [Электронный ресурс]. – PhD Thesis, MIT. – 2002. – Режим доступа: <https://dspace.mit.edu/handle/1721.1/39499> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

8. Dunkels A., Gronvall B., Voigt T. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors [Электронный ресурс] // Proceedings of the First IEEE Workshop on Embedded Networked Sensors (EmNets). – 2004. – Режим доступа: <https://ieeexplore.ieee.org/document/1372982> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

9. Beal J., Vachrach J. Infrastructure for Engineered Emergence in Sensor/Actuator Networks [Электронный ресурс] // IEEE Intelligent Systems. – 2006. – Vol. 21, no. 2. – Режим доступа: <https://ieeexplore.ieee.org/document/1614339> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

10. Brambilla M. et al. Swarm Robotics: A Review from the Swarm Engineering Perspective [Электронный ресурс] // Swarm Intelligence. – 2013. – Vol. 7, no. 1. – Режим доступа: <https://link.springer.com/article/10.1007/s11721-012-0075-2> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

11. Dias M. B., Stentz A. A Market Approach to Multirobot Coordination [Электронный ресурс] // Carnegie Mellon University, Technical Report CMU-RI-TR-01-26. – 2001. – Режим доступа: <https://www.ri.cmu.edu/publications/a-market-approach-to-multirobot-coordination/>(свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

12. Welsh M., Mainland G. Programming Sensor Networks Using Abstract Regions [Электронный ресурс] // Proceedings of the 1st USENIX Symposium on Networked Systems Design and Implementation (NSDI). – 2004. – Режим доступа: <https://www.usenix.org/conference/nsdi-04/programming-sensor-networks-using-abstract-regions> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

13. Coullouris G., Dollimore J., Kindberg T. Distributed Systems: Concepts and Design [Электронный ресурс]. – 5th ed. – Addison-Wesley, 2011. – Режим доступа: <https://www.cdk5.net/wp/>(свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

14. Bernstein D. Containers and Cloud: From LXC to Docker to Kubernetes [Электронный ресурс] // IEEE Cloud Computing. – 2014. – Vol. 1, no. 3. – Режим доступа: <https://ieeexplore.ieee.org/document/6884430> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

15. Hill J. et al. System Architecture Directions for Networked Sensors [Электронный ресурс] // Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX). – 2000. –

Режим доступа: <https://dl.acm.org/doi/10.1145/356989.356998> (свободный). –
Заглавие с экрана. – Дата обращения: 12.12.2025.

16. Rubenstein M., Cornejo A., Nagpal R. Programmable self-assembly in a thousand-robot swarm [Электронный ресурс] // Science. – 2014. – Vol. 345, no. 6198. – Режим доступа: <https://www.science.org/doi/10.1126/science.1254295> (свободный). –
Заглавие с экрана. – Дата обращения: 12.12.2025.