

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОГО НАЗЕМНОГО ТРАНСПОРТНО-ТЕХНОЛОГИЧЕСКОГО СРЕДСТВА

**Соловьев В.В.**

*Санкт-Петербургский государственный архитектурно-строительный  
университет, Санкт-Петербург*

**Ключевые слова:** наземные транспортно-технологические средства, программное обеспечение, классификация программного обеспечения, сравнение программного обеспечения, специальное программное обеспечение, многоагентные системы.

**Аннотация.** Статья посвящена анализу программного обеспечения, как ключевого компонента автоматизированных систем управления наземных транспортно-технологических средств. Представлена классификация программного обеспечения. Приведены сравнительные таблицы классов программного обеспечения, а также рассмотрены перспективы и сложности с программным обеспечением, предназначенным для управления группами наземных транспортно-технологических средств в рамках многоагентных систем.

## SOFTWARE FOR AUTOMATED GROUND TRANSPORTATION AND TECHNOLOGICAL FACILITY

**Solovev V.V.**

*Saint-Petersburg State University of Architecture and Civil Engineering,  
Saint-Petersburg*

**Keywords:** ground transportation and technological facilities, software, software classification, software comparison, special software, multi-agent systems.

**Abstract.** The article is devoted to the analysis of software as a key component of automated control systems of ground transportation and technological facilities. The classification of the software is presented. Comparative tables of software classes are presented, as well as the prospects and difficulties with software designed to manage groups of ground transportation and technological facilities within multi-agent systems.

Программное обеспечение (ПО) является одним из частей автоматизированной системы управления (АСУ) наземным транспортно-технологическим средством (НТТС). Оно позволяет организовать выполнение широкого спектра задач: от управления движением до взаимодействия с внешней средой.

Место ПО в АСУ НТТС представлено на рисунке 1 и характеризуется многоуровневым принципом [1], что позволяет разделить ответственность между компонентами и повысить надежность системы. Рассмотрим классы ПО:

- операционные системы (ОС);
- общее программное обеспечение (ОПО);
- специальное программное обеспечение (СПО).

Наибольшее распространение для решения прикладных задач в разработке и производстве НТТС занимают unix-подобные ОС (рис. 2), т.к. они имеют ряд преимуществ, главным из которых является открытый исходный код, доступный для получения и использования (изменения в своих целях) каждому [2].

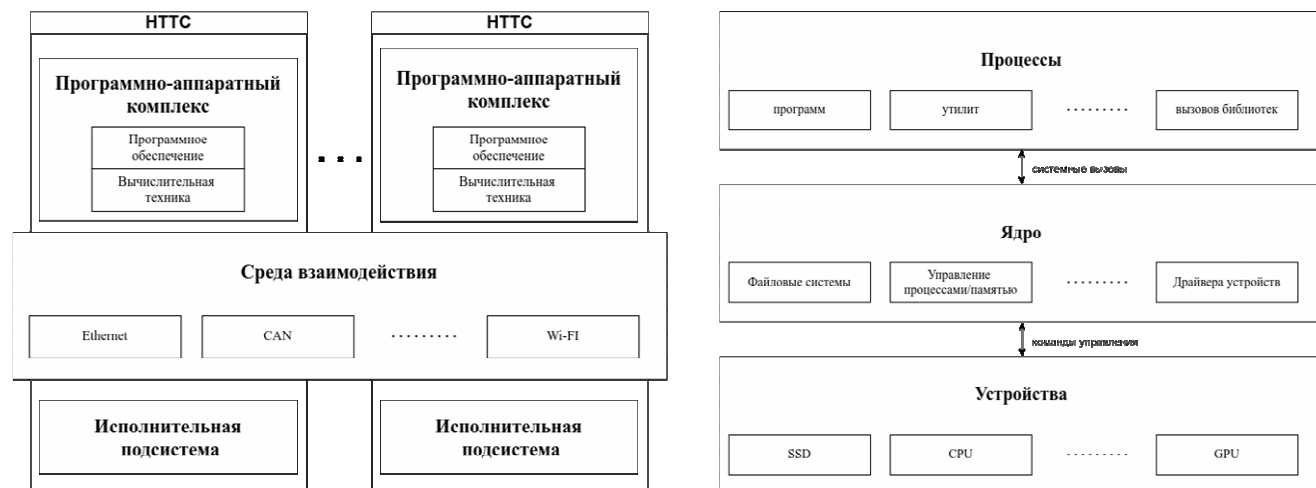


Рис. 1. Место ПО в АСУ НТТС

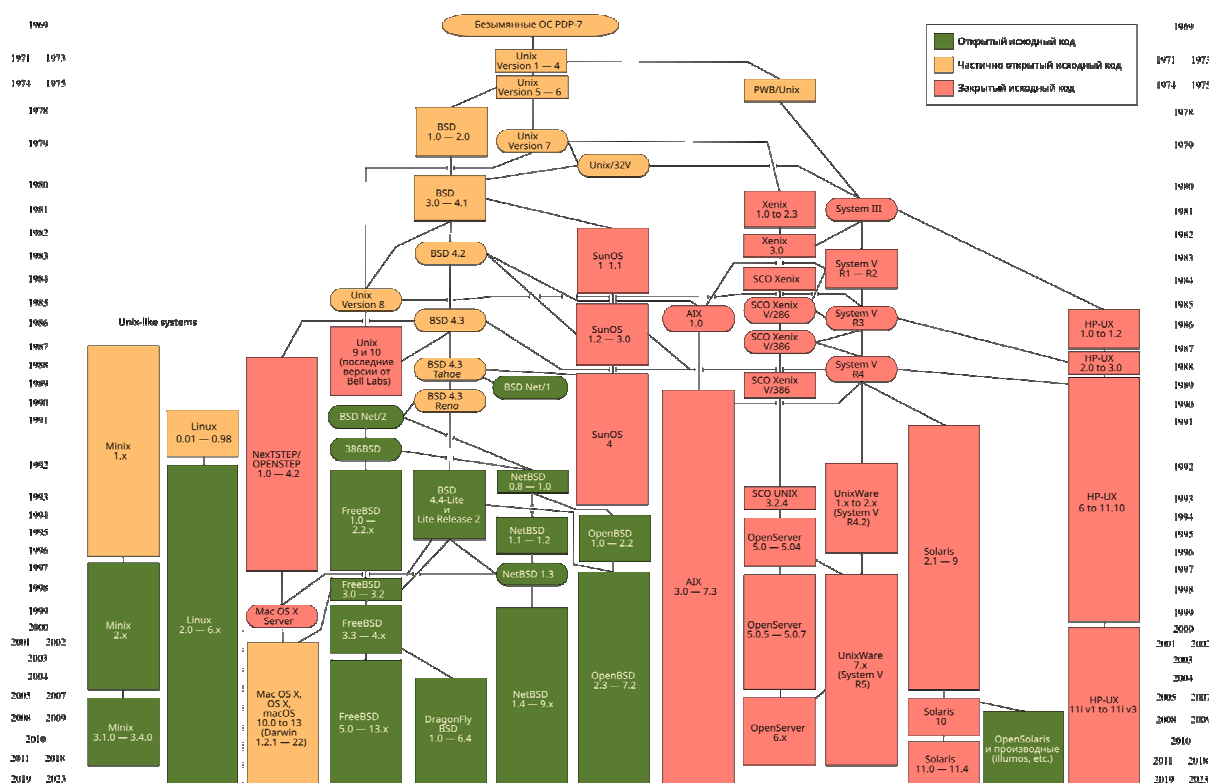


Рис. 2. Эволюция unix-подобных операционных систем

ОС для НТТС должны соответствовать жестким требованиям:

- *процессы*: изоляция задач, планирование с приоритетами;
- *ядро*: минимизация латентности, поддержка реального времени;
- *драйвера устройств*: прямое управление аппаратурой.

Примерами таких ОС, применяемых в разработке автоматизированных НТТС являются:

- Linux с дополнением PREEMPT\_RT (патч для ядра Linux, уменьшающий задержки планирования): открытость, поддержка широкого спектра устройств;
- QNX (микроядерная ОС с сертификацией для медицинских и промышленных систем.): детерминированное выполнение, отказоустойчивость;
- VxWorks (ОС реального времени, используемая в аэрокосмической отрасли): сертификация DO-178C для критических задач.

Сравнение примеров ОС для НТТС представлено в таблице 1.

Табл. 1. Сравнение ОС для НТТС

| ОС       | Тип ядра   | Макс. задержка | Сертификация | Поддержка    | Применение |
|----------|------------|----------------|--------------|--------------|------------|
| QNX      | Микроядро  | 10 мкс         | ISO 26262    | Коммерческая | Авто       |
| VxWorks  | Монолитное | 5 мкс          | DO-178C      | Коммерческая | Авиация    |
| FreeRTOS | Микроядро  | 50 мкс         | IEC 61508    | Открытая     | IoT        |

ОПО (рис. 3) включает базовые сервисы для управления ресурсами системы [3]:

- загружаемые модули ядра;
- системные библиотеки;
- утилиты мониторинга и др.

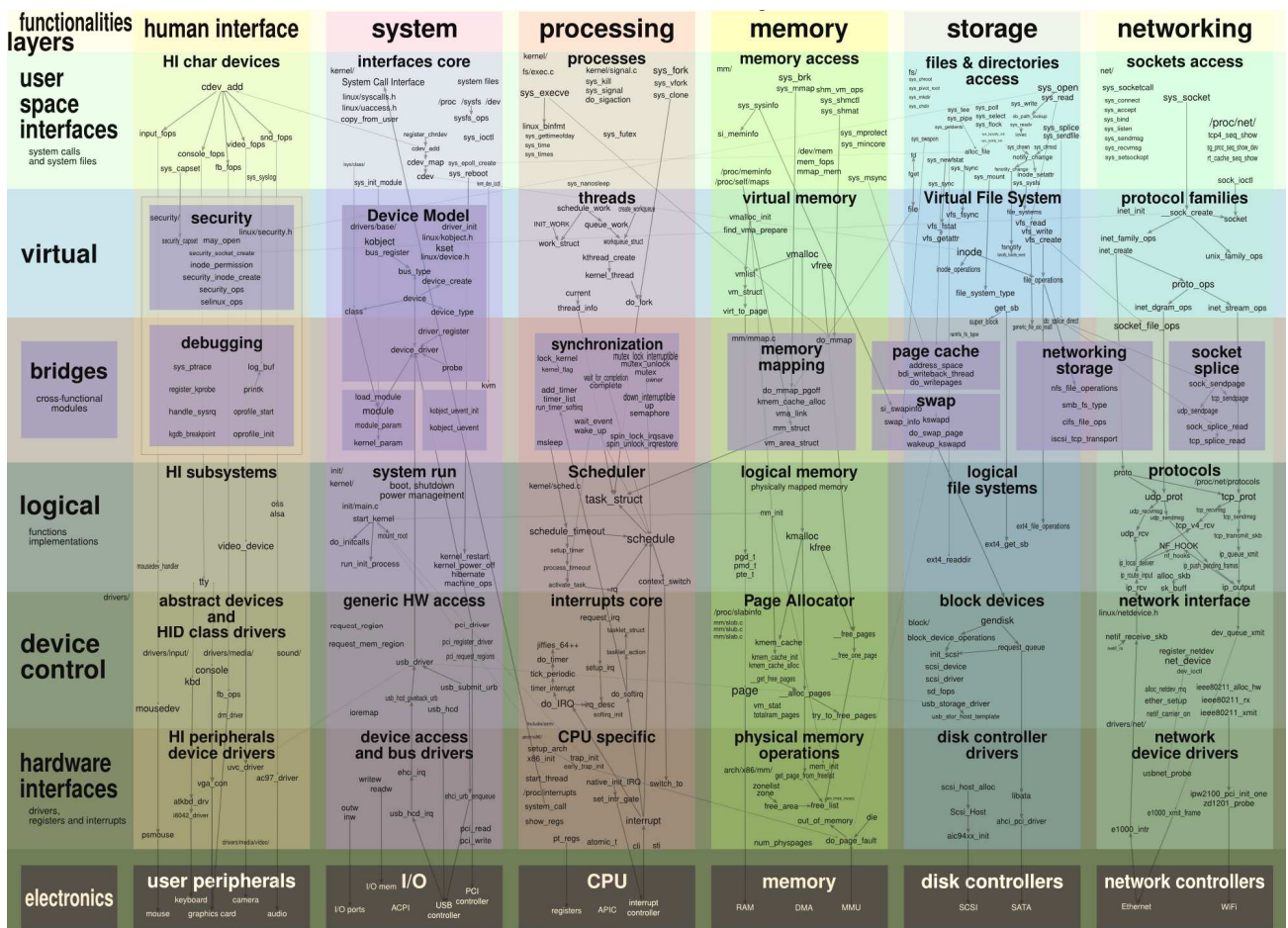


Рис. 3. Архитектура ОС и ОПО

СПО включает алгоритмы управления движением, обработки данных с датчиков и взаимодействия с исполнительными механизмами (рис. 4). Основные требования к нему:

- *детерминизм*: выполнение операций за строго заданное время;
- *отказоустойчивость*: автоматическое восстановление после сбоев;
- *интеграция с внешними системами*: предоставление интерфейса программного приложения (API) для доступа к ресурсам НТТС;
- *интеграция с ОС*: использование API ОС для доступа к ресурсам.



Рис. 4. Специализированное ПО АСУ НТТС

Примеры реализаций СПО для НТТС:

- ROS 2 (Robot Operating System 2 – фреймворк для разработки робототехнических систем с поддержкой реального времени): поддерживает распределенные вычисления и QoS;
  - AUTOSAR (стандарт для автомобильной электроники с разделением на прикладной и базовый уровни.): обеспечивает модульность и переносимость;
  - СПО на базе ИИ: нейросетевые модели для прогнозирования состояний.
- Сравнение примеров СПО для НТТС представлено в таблице 2.

Табл. 2. Сравнение СПО для НТТС

| Решение | Реальное время | Безопасность | Совместимость  | Применение          |
|---------|----------------|--------------|----------------|---------------------|
| ROS 2   | Да (с RTOS)    | SIL 2        | Linux, Windows | Робототехника       |
| AUTOSAR | Да             | ASIL D       | CAN, Ethernet  | Автомобили          |
| QNX     | Да             | IEC 61508    | ARM, x86       | Критические системы |

Рассмотренные классы ПО обеспечивают базовый функционал для отдельных компонентов системы, однако управление группой взаимодействующих автоматизированных НТТС требует принципиально иного подхода.

Координация множества автоматизированных НТТС в реальном времени представляет собой задачу:

- *масштабируемости*: синхронизация десятков, сотен или тысяч агентов;
- *ресурсных конфликтов*: распределение зон ответственности, энергии и маршрутов;
- *динамической адаптации*: перераспределение задач при отказах/вмешательствах.

Данная проблема решается в рамках рассмотрения НТТС как элемента *многоагентных систем* (МАС) [4], где каждый НТТС выступает как:

- автономный агент с локальными целями;
- участник подмножества МАС с глобальной задачей;
- активный потребитель и поставщик ресурсов.

Реализация различных сценариев применения НТТС в МАС требует специализированного СПО, включающего:

- математические модели распределенного принятия решений;
- алгоритмы оптимизации на графах зависимостей;
- механизмы балансировки централизованного и децентрализованного управления.

#### **Список литературы**

1. Таненбаум Э.С. Современные операционные системы. – СПб.: Питер, 2002. – 1040 с.
2. Кетов Д.В. Внутреннее устройство Linux. – СПб.: БХВ-Санкт-Петербург, 2017. – 320 с.
3. Уорд Б. Внутреннее устройство Linux. – СПб.: Питер, 2016. – 384 с.
4. Соловьев В.В., Евтюков С.А. Модель системы многозадачной координации автоматизированных транспортно-технологических средств // Грузовик. – 2023. – №7. – С. 36-39. – doi.org/10.36652/1684-1298-2023-7-36-39.

#### Сведения об авторе:

Соловьев Виктор Владимирович – аспирант.