

Разработка планировщика ОСРВ с совмещением кооперативной и вытесняющей многозадачности

Дипломная работа

Автор:

студент 544 группы
Логинова В.В.

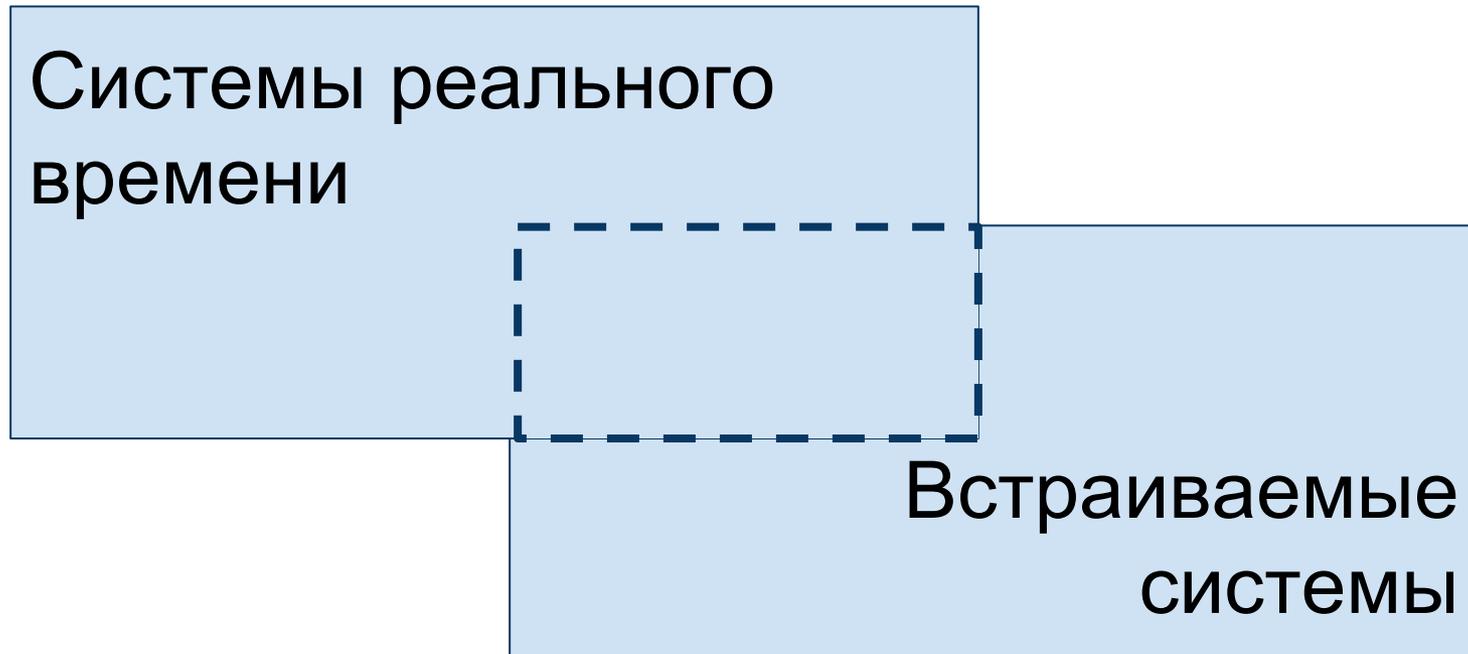
Научный руководитель:

асп. каф. системного программирования
Козлов А.П.

Рецензент:

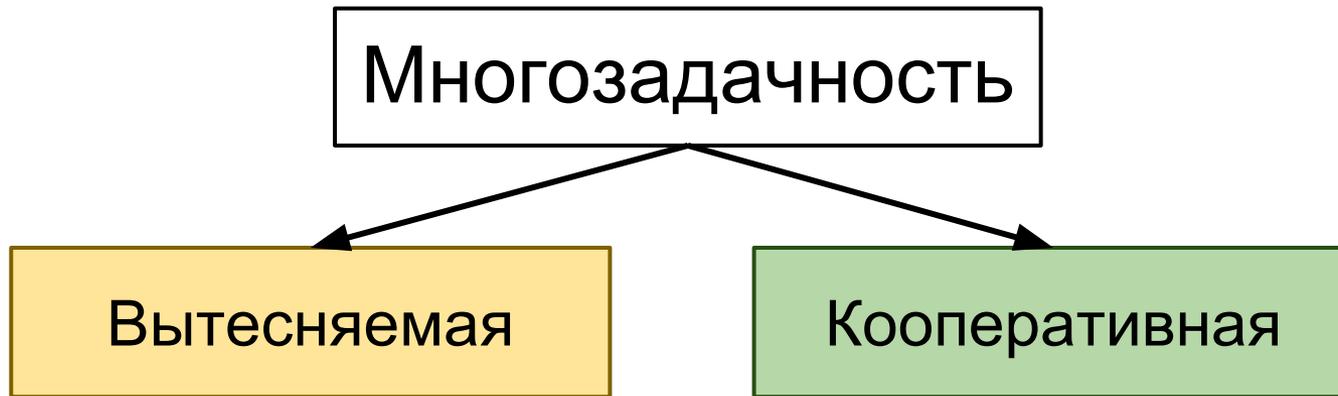
зав. каф. системного программирования, д.ф.-м.н., проф.
Терехов А.Н.

Предметная область



- Встраиваемые системы и системы реального времени
- Требования:
 - производительность
 - ограниченность ресурсов
 - предсказуемое время

Предметная область



Типы многозадачности:

- вытесняемая
- кооперативная
 - в том числе событийно-ориентированный подход

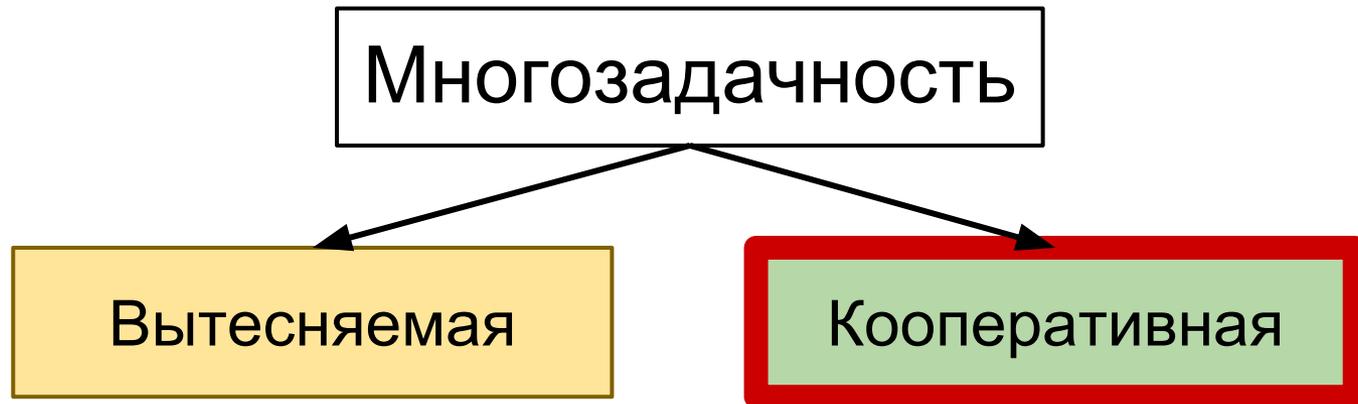
Проблемы многозадачности в существующих ОС:

- или поддерживаются не все типы
- или поддерживаются все, но в изолированных планировщиках и разных очередях

Постановка задачи

- Разработать архитектуру планировщика, который совмещает разные типы многозадачности и использует общую очередь для различных типов единиц планирования
- Реализовать разработанный планировщик в ядре ОСРВ Embox
- Произвести оценку полученной реализации

Обзор: TinyOS и Contiki



Свойства:

- событийно-ориентированный подход
- потоки не имеют стека

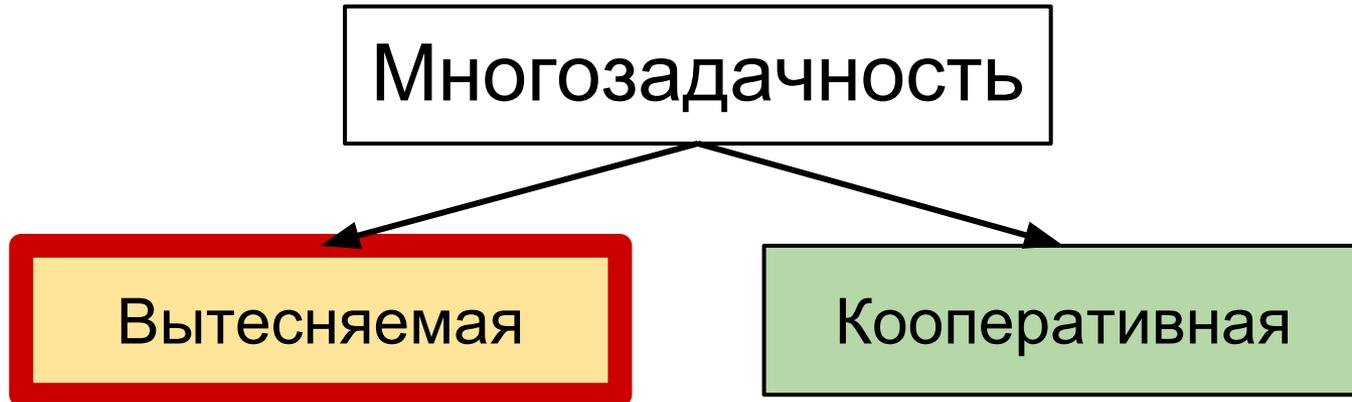
Преимущества:

- высокая производительность
- экономия памяти

Недостатки:

- для узкого класса устройств

Обзор: QNX



Свойства:

- обработчики прерываний как обычные потоки

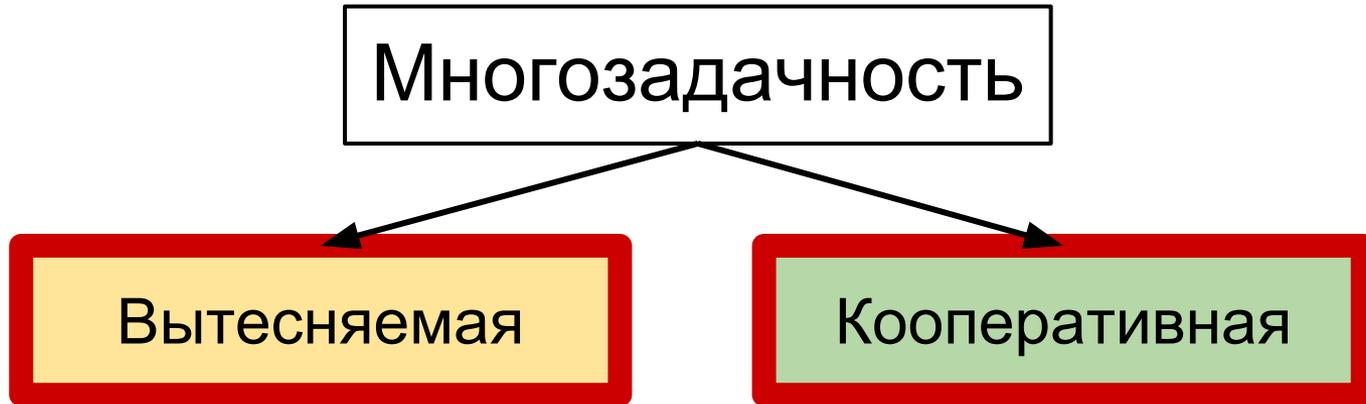
Преимущества:

- единый планировщик

Недостатки:

- ресурсозатратно: не для маленьких устройств

Обзор: FreeRTOS



Свойства:

- оба типа многозадачности: задачи - вытесняемая, сопрограммы - кооперативная
- сопрограммы не имеют стека

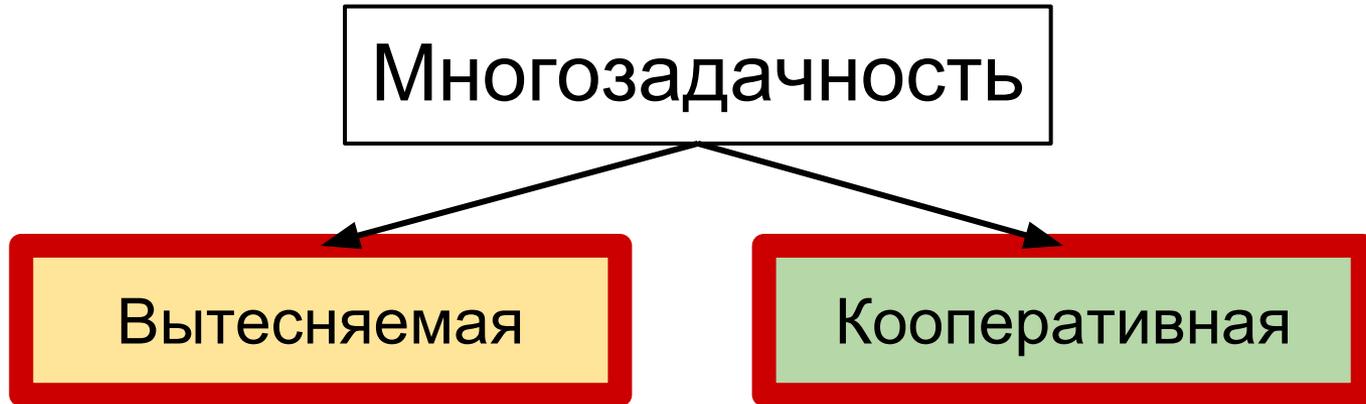
Преимущества:

- конфигурация либо с одним из типов, либо с обоими

Недостатки:

- планировщики изолированы
- нет синхронизации между задачами и сопрограммами

Обзор: Linux



Softirq

- обработчики прерываний задаются статически

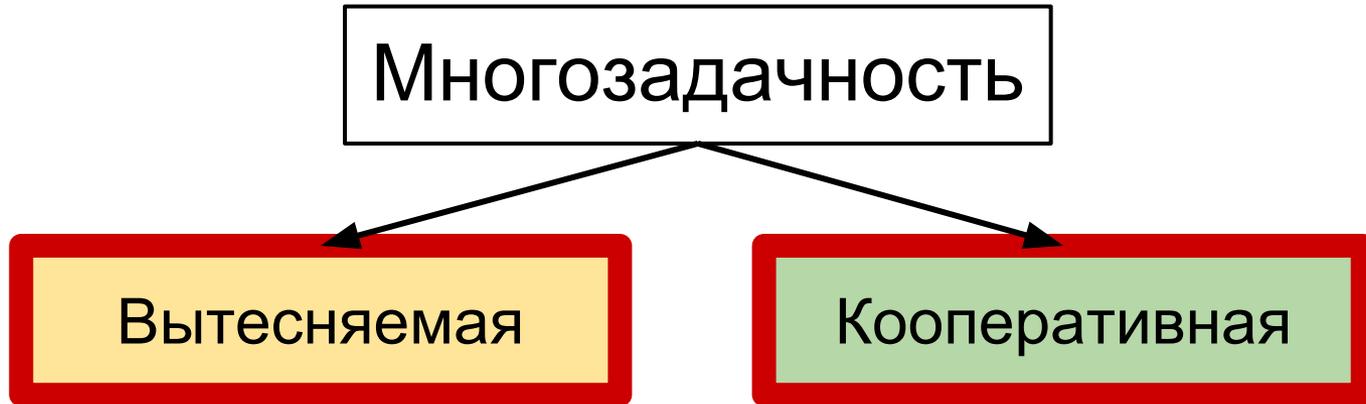
Tasklet

- кооперативная многозадачность
- планировщики изолированы, без синхронизации

Workqueue

- другой планировщик, но встроен в основной
- ресурсозатратно

Embox

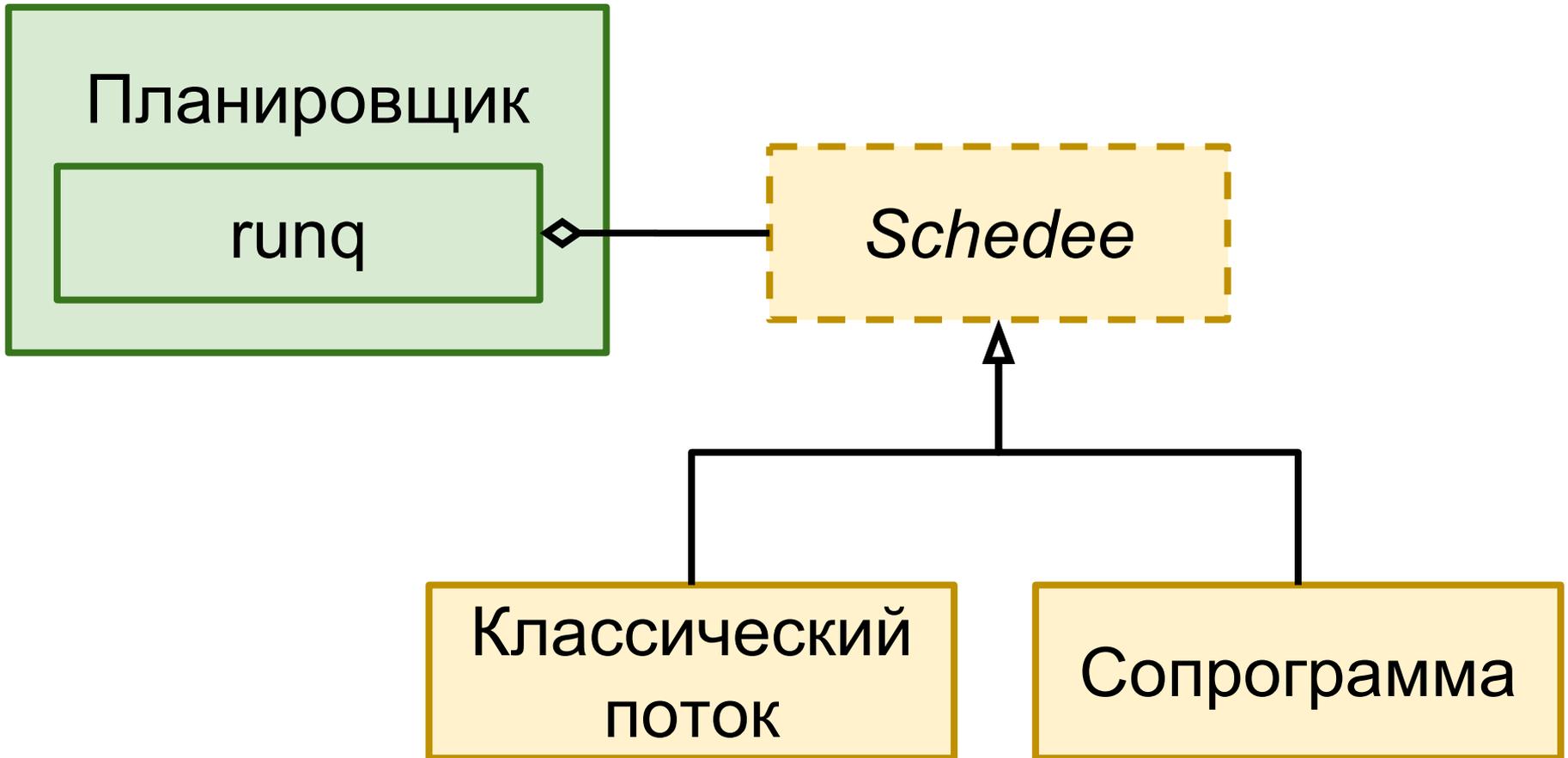


- Обработка прерываний в softirq как в Linux
- POSIX-совместимые потоки

Проблемы:

- **Планировщики изолированы**
- **Нельзя синхронизоваться в прерываниях**

Архитектура: абстрактная единица планирования



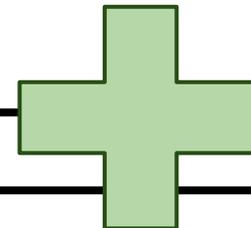
Архитектура: совмещенный алгоритм планирования

1. Планирование потоков со стеком

```
schedule() {  
    prev = get_prev_schedee();  
    if (!is_sleeping(prev))  
        runq.insert(prev);  
    next = runq.extract();  
    switch_context(prev, next);  
}
```

2. Планирование потоков без стека

```
schedule() {  
    while (true)  
        run(runq.extract());  
}
```

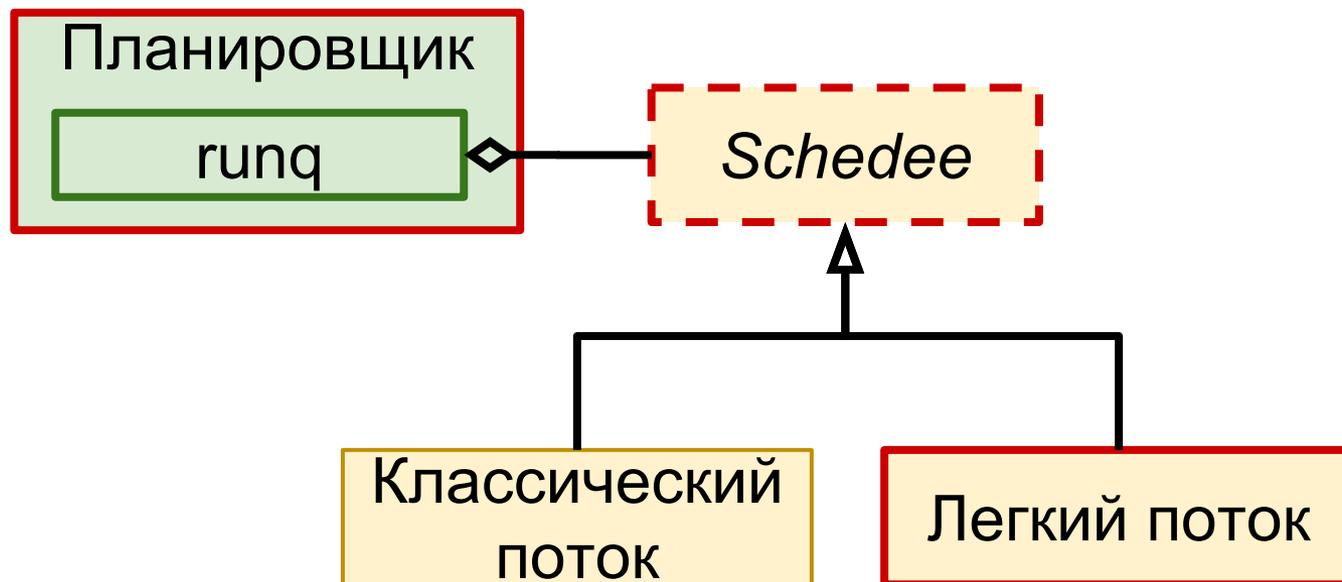


Архитектура: совмещенный алгоритм планирования

```
schedule() {  
    prev = get_prev_schedee();  
    if (!is_sleeping(prev))  
        runq.insert(prev);  
    do {  
        next = runq.extract();  
        next.process();  
    } while (next.has_stack());  
}
```

Реализация архитектуры в Embox: планировщик

- Реализация schedee и его API
- Переработка инфраструктуры планировщика
- Легкие потоки для поддержки кооперативной многозадачности.



Реализация архитектуры в Embbox: легкие потоки

Свойства легких потоков:

- Не имеют стека
- Кооперативные
- Сопрограммы
 - работа с метками на основе расширения GCC
Labels as Values
- Поддержка классической синхронизации

Применение обобщенного планировщика

- Один планировщик и одна очередь на все типы потоков:
 - обработка отложенных прерываний в легких потоках
 - гибкая настройка приоритетов
- Низкая связность компонент
 - конфигурируемость
 - использование ОС на устройствах с ограниченными ресурсами

Сравнение производительности легких и классических потоков

- Процессор: Intel Core i7 3517U 1900 МГц

	Число потоков	Полноценные потоки	Легкие потоки	Разница
Среднее время исполнения пустого потока	3	5414 (0.3%)	418 (0.5%)	92.3%
	10	5307 (0.4%)	274 (0.2%)	94.8%
	30	5360 (2.4%)	235 (0.2%)	95.6%
Среднее время заблокированных прерываний	3	405 (3.4%)	169 (1.0%)	58.3%
	10	432 (4.8%)	128 (0.4%)	70.4%
	30	436 (1.6%)	115 (0.9%)	73.6%

*Данные указаны в тактах процессора, в скобках приведено стандартное отклонение

Сравнение образов Embox с разной конфигурацией планировщика

- STM32VLDISCOVERY: 8кб RAM и 128кб flash
- Демо-приложение: 2 параллельных потока (легких или полноценных)

		Оба типа	Только легкие	Разница
Flash	text Os	30152	28724	1428 (4,7%)
RAM	data	436	436	3040 (37%)
	bss	3102	3168	
	Основной стек + стек потоков	1078 + 1782*2	1536 + 0	

*Данные указаны в байтах

Результаты

- Разработана архитектура планировщика:
 - выделена обобщенная единица планирования
 - разработан алгоритм с поддержкой различных типов единиц планирования в общей очереди
- Предложенная архитектура реализована в Embbox:
 - переработан существующий планировщик
 - разработаны кооперативные единицы планирования и добавлен программный интерфейс для управления ими
- Произведена оценка полученной реализации:
 - существенная экономия RAM при использовании легких потоков
 - жизненный цикл легких потоков короче на ~90%
 - время блокировки аппаратных прерываний сокращается на ~60-70% при переключении легких потоков