Микроядерная архитектура как основа надёжности и безопасности ОСРВ «Нейтрино»

Владимир Махилёв
ООО «СВД ВС»
Руководитель группы разработки
Отдел операционных систем

СВД
Встраиваемые Системы

О нас

«СВД ВС» - разработчик ПО для встраиваемых систем реального времени

- Компания основана в 2002 году для работы на российском рынке
- Офис в Санкт-Петербурге
- >50 человек, бóльшая часть - разработчики
- Лицензии ФСБ, ФСТЭК, МО, МПТ, ПО входит в реестр Минцифры и имеет сертификаты ФСТЭК и Минобороны

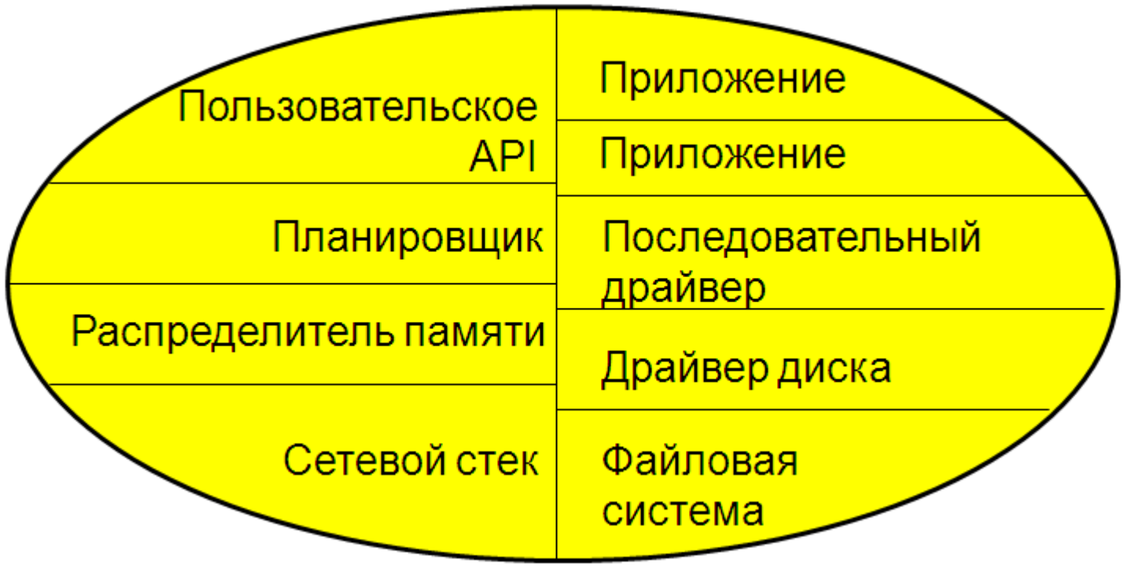
СВД
Встраиваемые Системы

Продукты:

- операционная система «Нейтрино»
- СДКУ «Фокус»
- Картографический пакет
- ИИ «Синаптика»

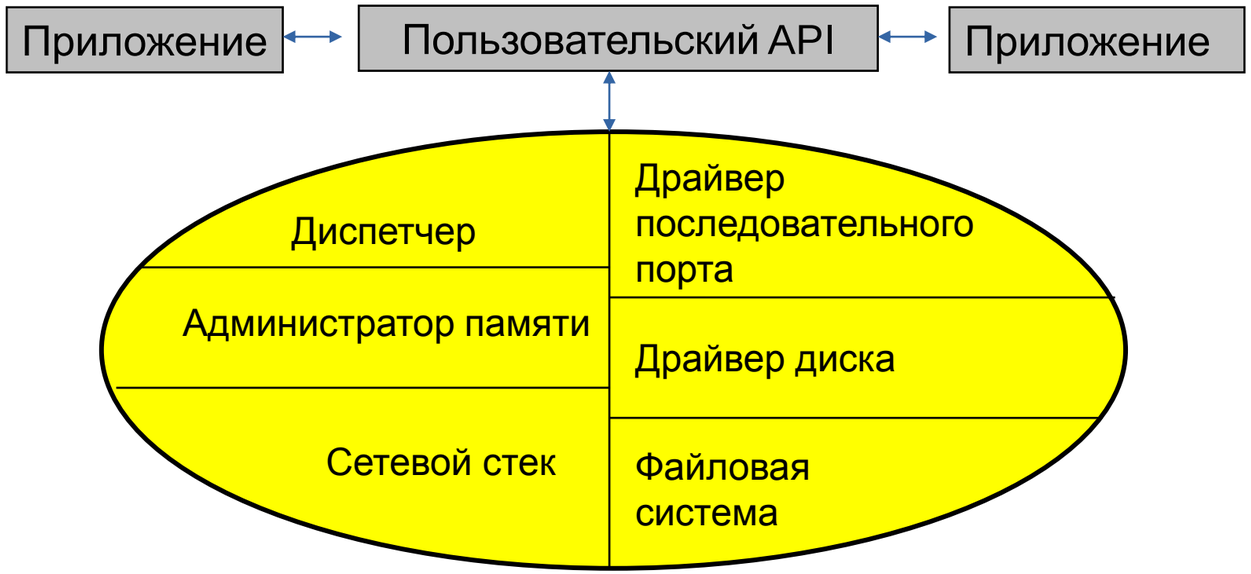
Архитектура: исполнительный модуль

- Низкие требования к системным ресурсам и памяти, возможность работать на микроконтроллерах без MMU
- Относительная простота
- Все компоненты ОС используют единое адресное пространство и, по сути, являются одной большой программой
- Сбой в любом компоненте может привести к отказу системы



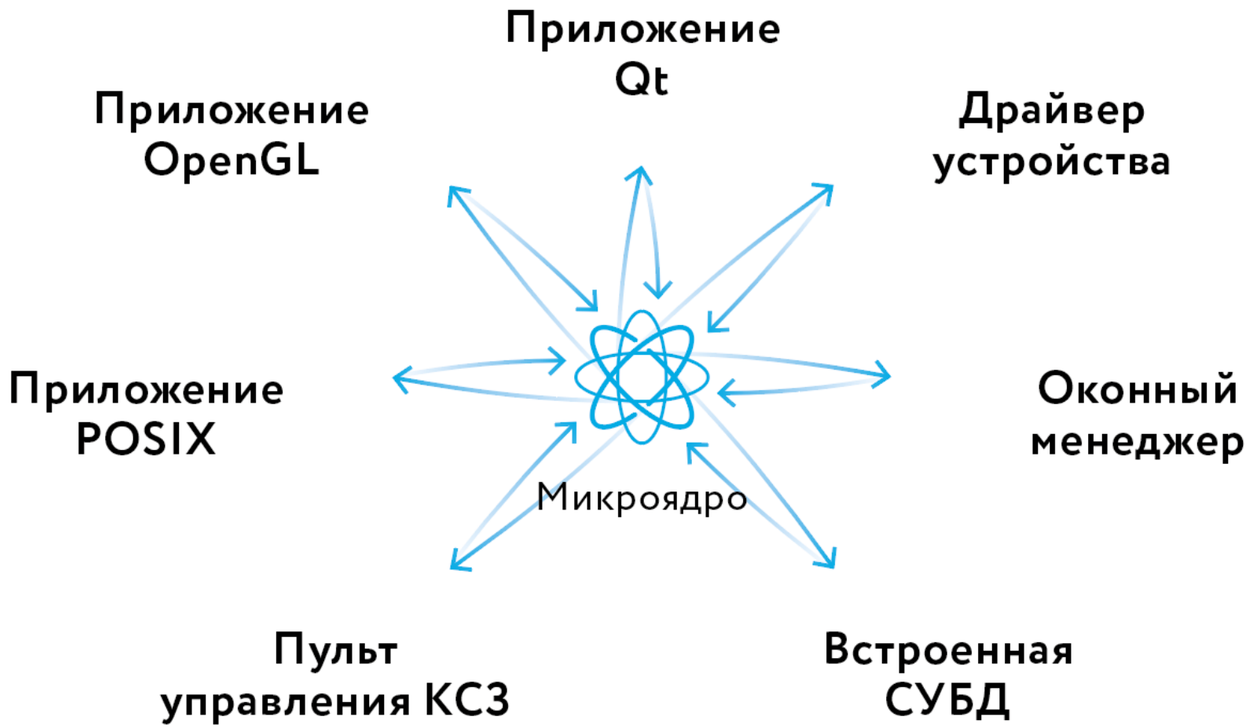
Архитектура: монолитное ядро

- Все функциональные компоненты ОС, в том числе драйверы, входят в состав ядра и работают в общем адресном пространстве
  - Приложения — это процессы в обособленной памяти
  - Ядро защищено от приложений, а приложения — друг от друга
  - Сбой в одном компоненте ядра может привести к отказу системы



Архитектура: микроядро

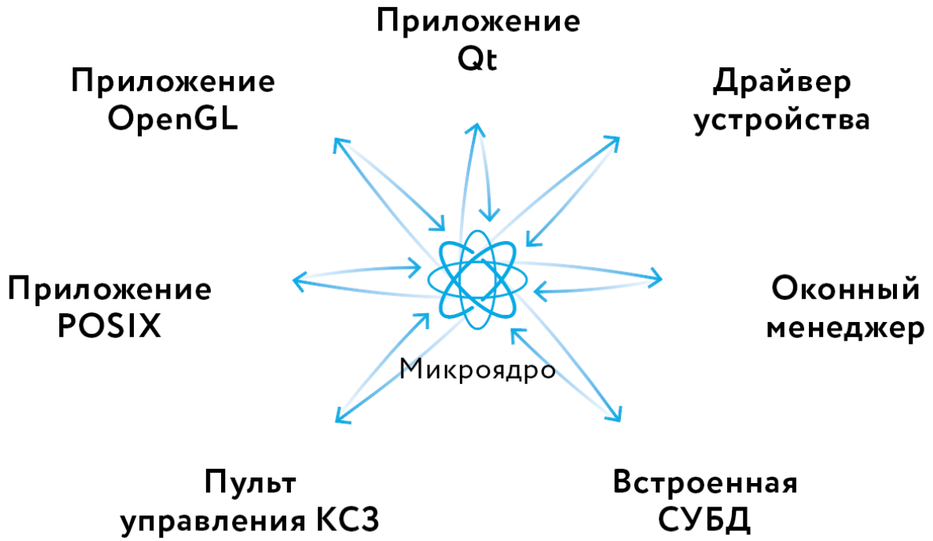
- ОС состоит из микроядра и множества взаимодействующих процессов
- Микроядро отвечает за ограниченный набор функций и передачу сообщений
  - Драйверы и пользовательские процессы отделены от микроядра и работают в изолированных адресных пространствах



Микроядерная архитектура

Преимущества:

- Ядро, драйверы и пользовательские процессы работают в изолированных адресных пространствах
- Сбой в процессе не влияет на микроядро и процесс может быть перезапущен, включая драйверы
- Масштабируемость и простота настройки — драйверы запускаются по необходимости
- Предсказуемость поведения

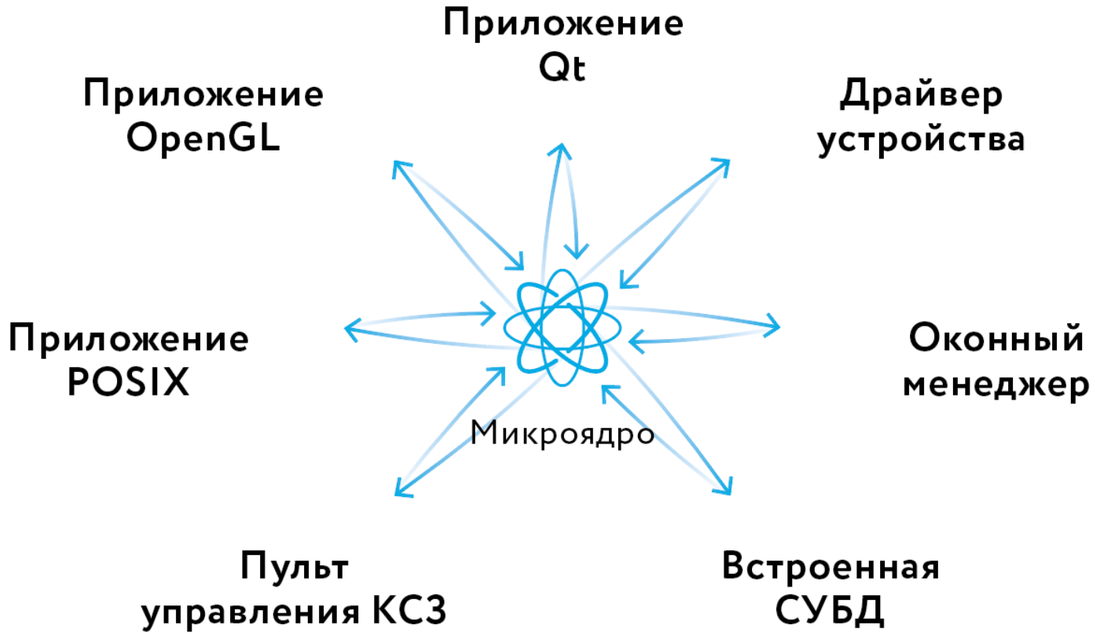


Недостатки:

- Накладные расходы
  - Переключения контекста
  - Копирование данных между процессами и ядром

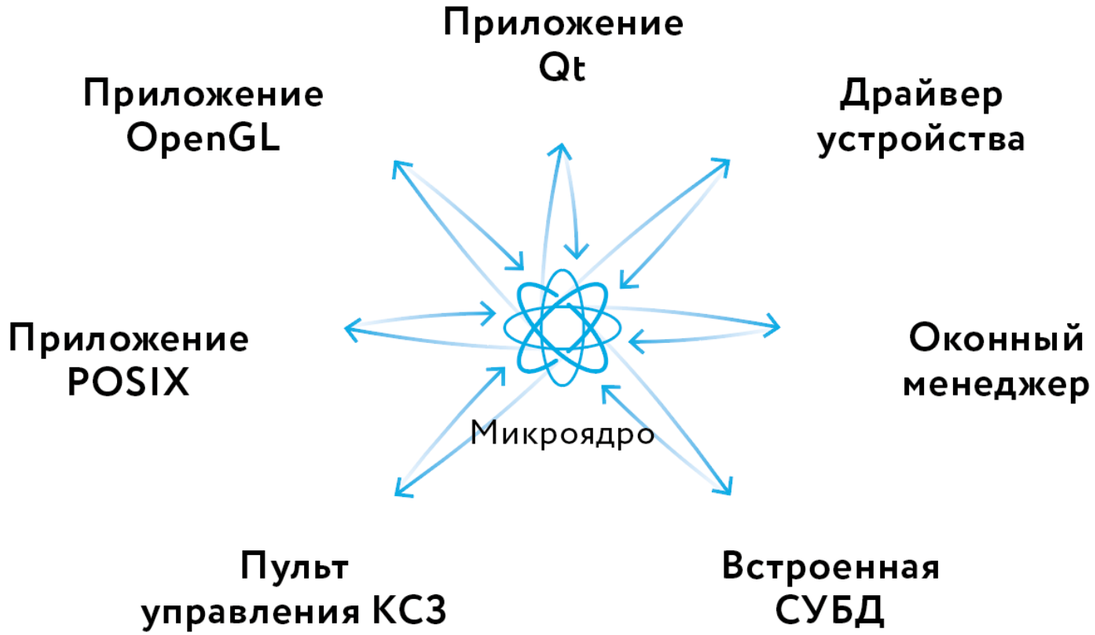
Микроядро «Нейтрино»

- Связующий элемент системы, который объединяет подсистемы в единое целое
- Программы взаимодействуют с ядром с помощью системных функций (вызовов), которые исполняют код в ядре
- Большинство других подсистем, в том числе пользовательские приложения, взаимодействуют между собой с помощью методов межпроцессного взаимодействия, предоставляемых ядром



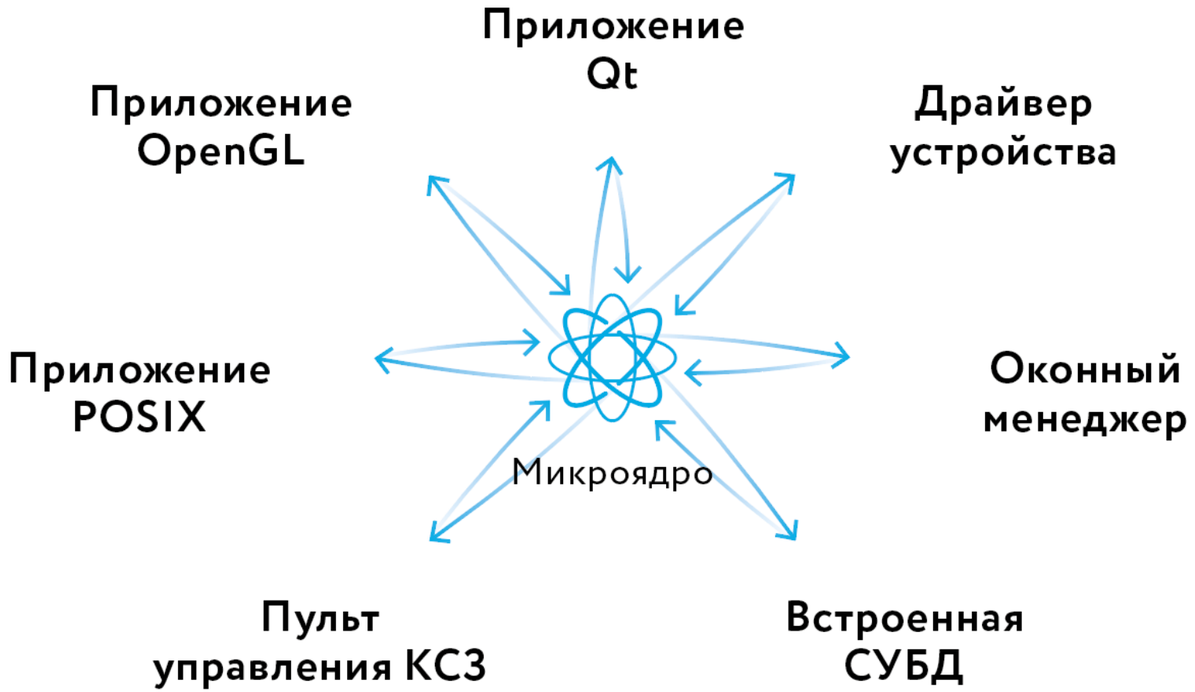
Микроядро «Нейтрино»

- Микроядро не подлежит плановому исполнению, можно рассматривать его как библиотеку
- Выполняется только в ответ на следующие события:
  - системный вызов
  - прерывание
  - исключение процессора (недопустимая инструкция, некорректный адрес)



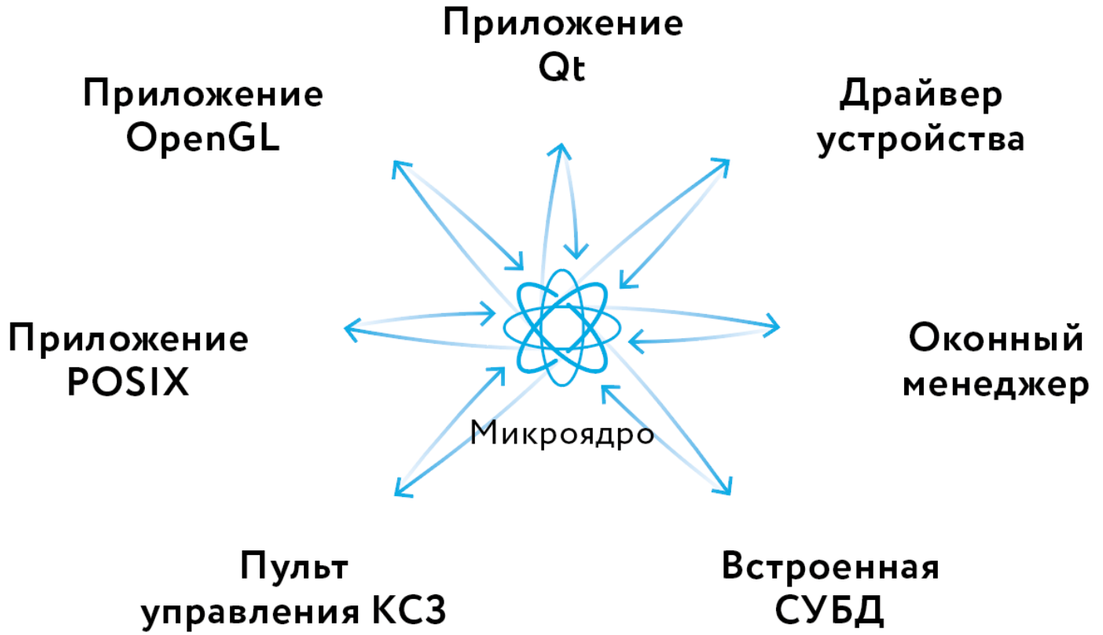
Микроядро «Нейтрино»: основные функции

- Диспетчеризация
  - Приоретиты потоков 0 — 255
- Синхронизация
  - Мьютексы, условные переменные, семафоры
- Управление потоками
  - Создание/завершение
  - Ожидание завершения
  - Изменение атрибутов
- Межпроцессное взаимодействие
  - Сообщения (обмен данными)
  - Импульсы (уведомления о событиях)
  - Сигналы (POSIX, прерывание другого процесса)
- Работа с прерываниями



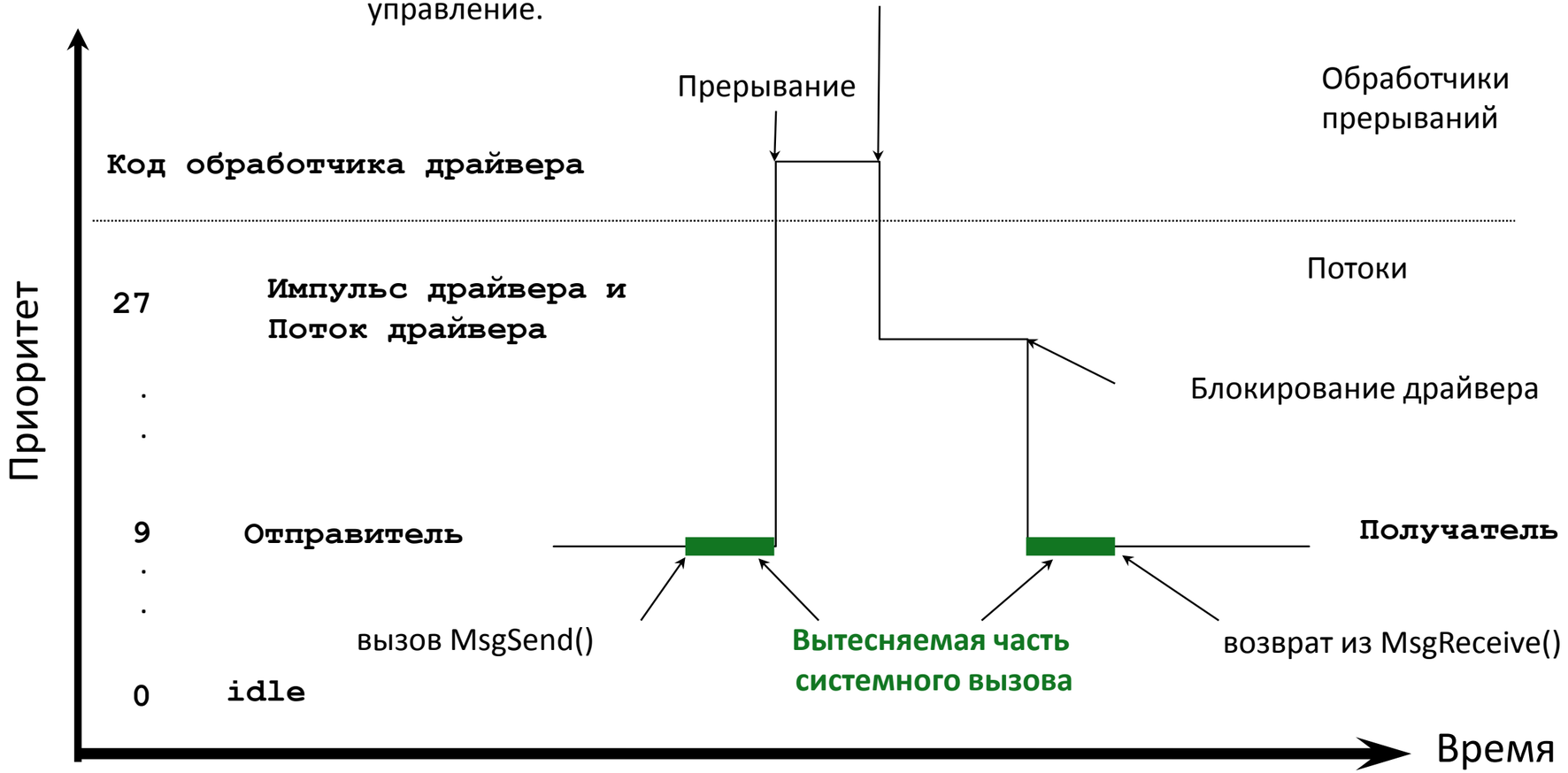
Микроядро «Нейтрино»: вытесняемость ядра

- Операции, требующие значительных ресурсов (загрузка программы в память), вынесены во внешние процессы (менеджеры ресурсов)
- Микроядро вытесняемо
  - Длительные операции (копирование данных при передаче сообщения) сохраняют и восстанавливают состояние
  - Короткие операции (проверка адресов) выполняются повторно
  - Самые короткие операции, требующие атомарности операций (изменение состояния потока, захват мьютекса), запрет прерываний и вытеснений
    - Промежуток максимально короткий — порядка ~десятков наносекунд

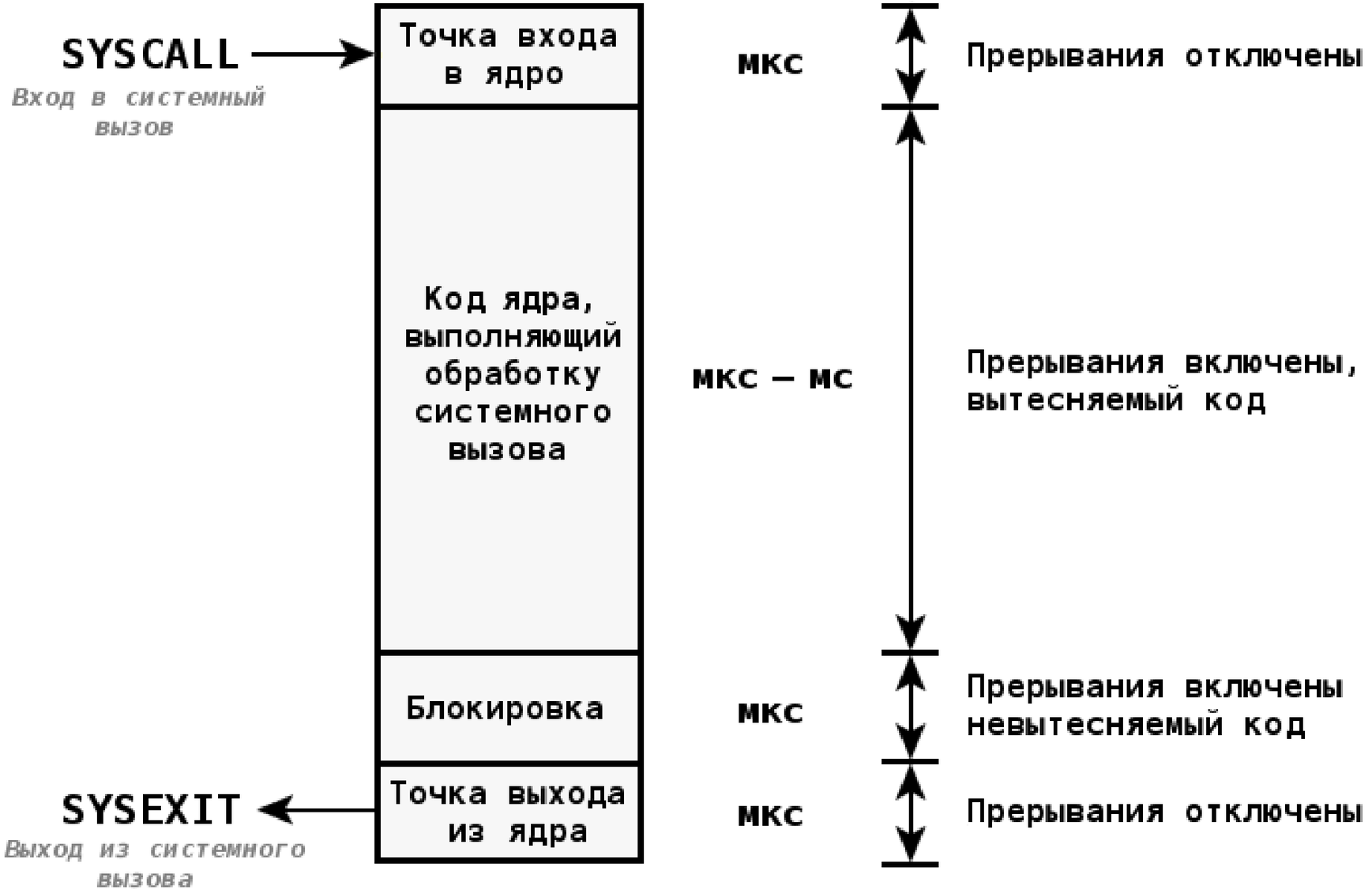


Микроядро «Нейтрино»: вытесняемость системного вызова

Обработчик прерывания возвращает в ядро импульс с приоритетом 27, затем ядро переводит поток драйвера в состояние READY и передаёт ему управление.



Микроядро «Нейтрино»: вытесняемость системного вызова



Вытесняемость системных вызовов

Преимущества:

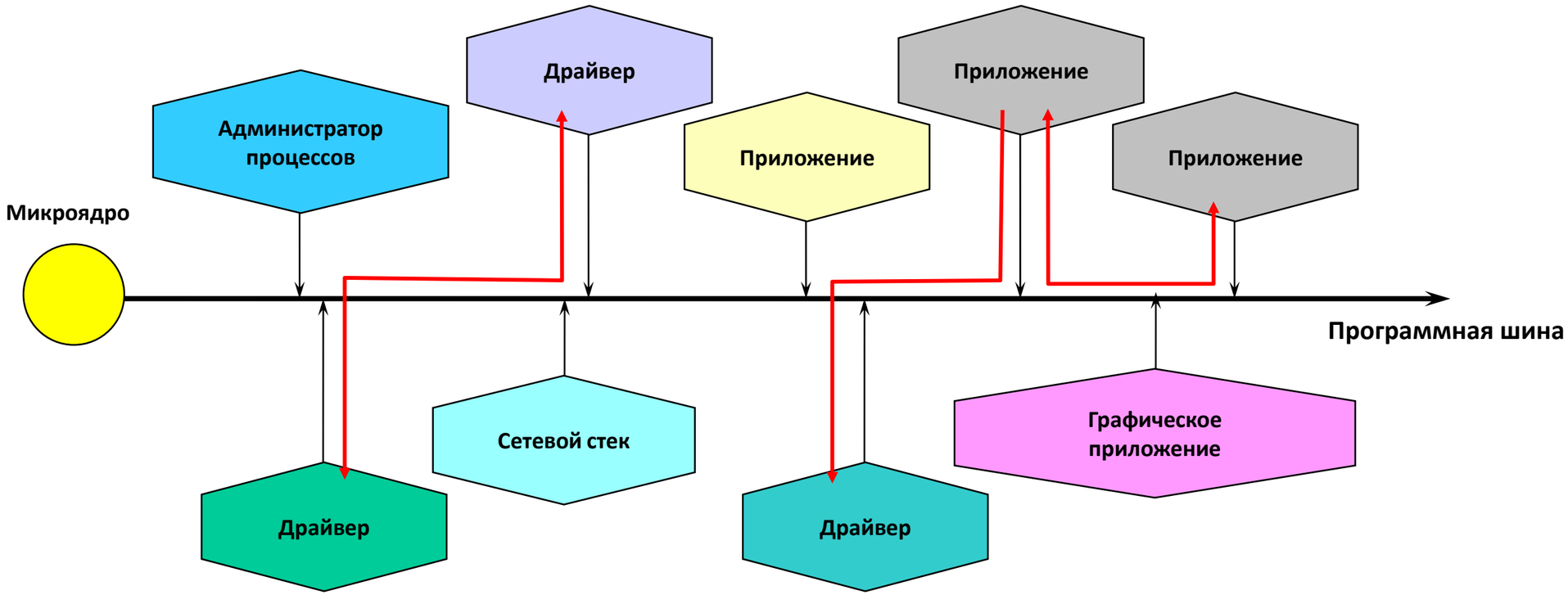
- сокращение задержек
- быстрая реакция на новые события
- уменьшение задержки прерывания и переключения

Недостатки:

- накладные расходы
- требуется больше времени для перезапуска прерванного системного вызова
- требуется больше времени для сохранения текущего состояния и перезапуска вытесненной задачи

Микроядро: взаимодействие процессов

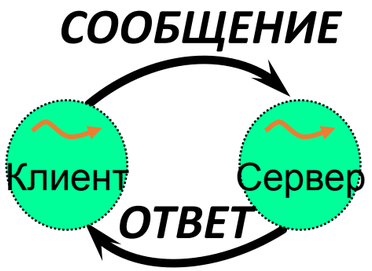
- Пользовательские и системные процессы взаимодействуют друг с другом с помощью единых методов межпроцессного взаимодействия и передачи сообщений



Межпроцессное взаимодействие

Методы межпроцессного взаимодействия, предоставляемые ядром:

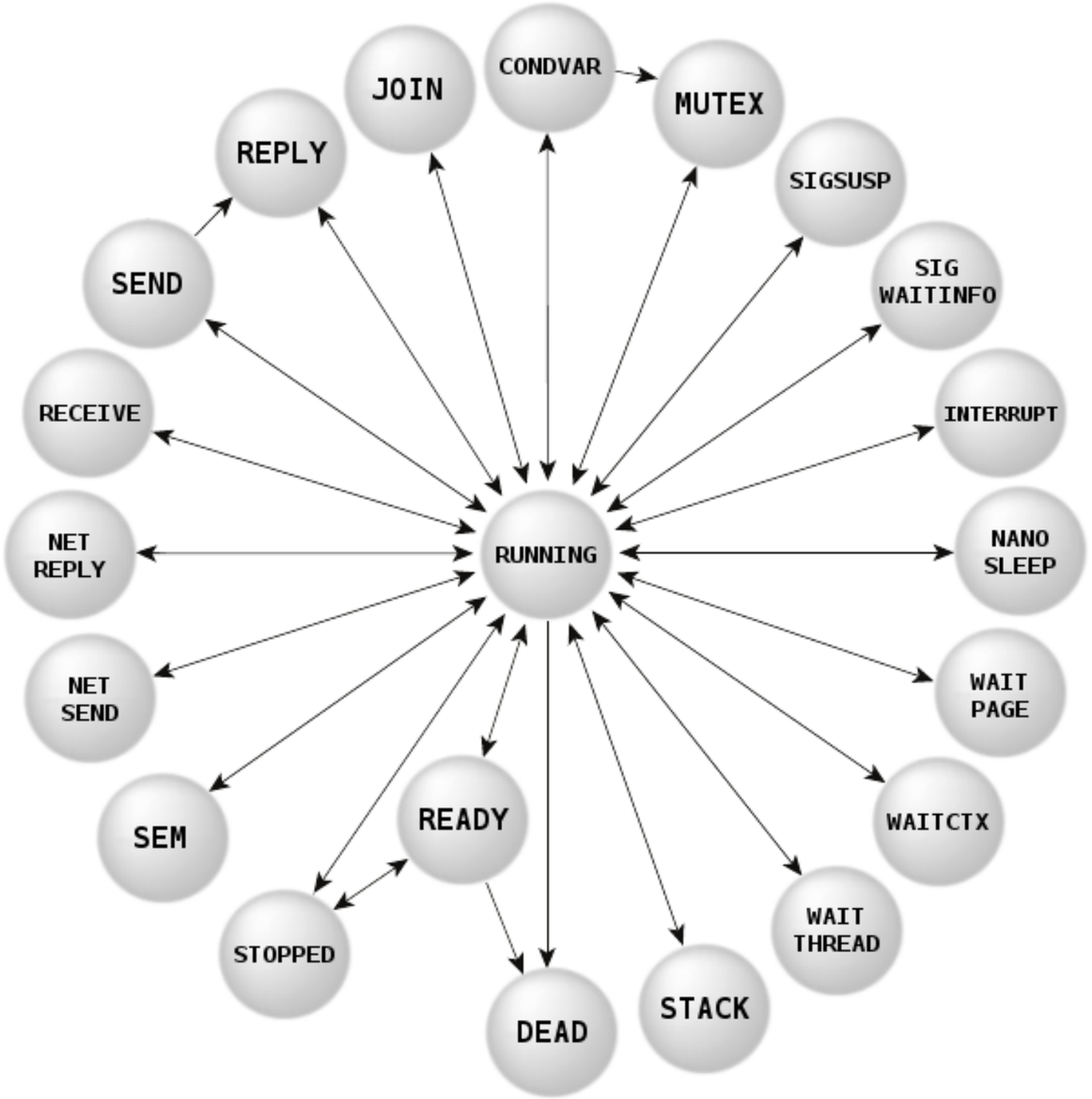
- Сообщения
  - обмен данными между процессами
- Импульсы
  - доставка уведомления процессу
- Сигналы
  - прерывание процесса и его переключение на другую задачу (обычно завершение)



Управление потоками

Функции управления потоками:

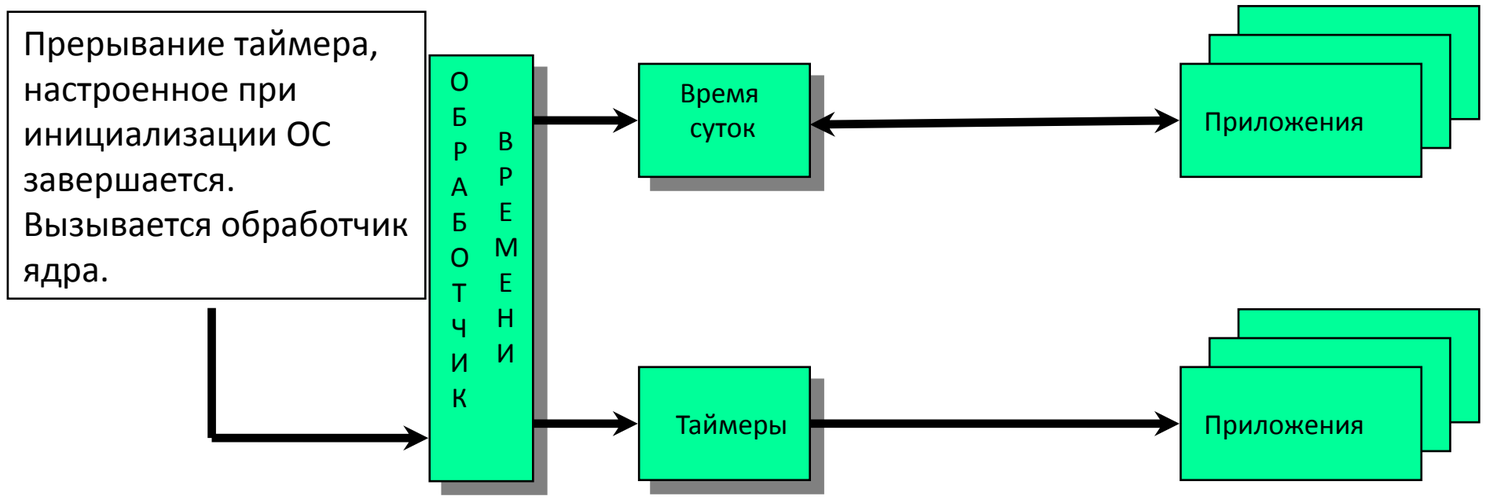
- Создание/завершение
- Ожидание завершения
- Изменение атрибутов



Подход к работе с временем в «Нейтрино»

Ядро:

- ведёт хронометраж, позволяет считывать и устанавливать время
- позволяет создавать и использовать:
  - периодические и однократные таймеры
  - таймауты для блокирующих вызовов

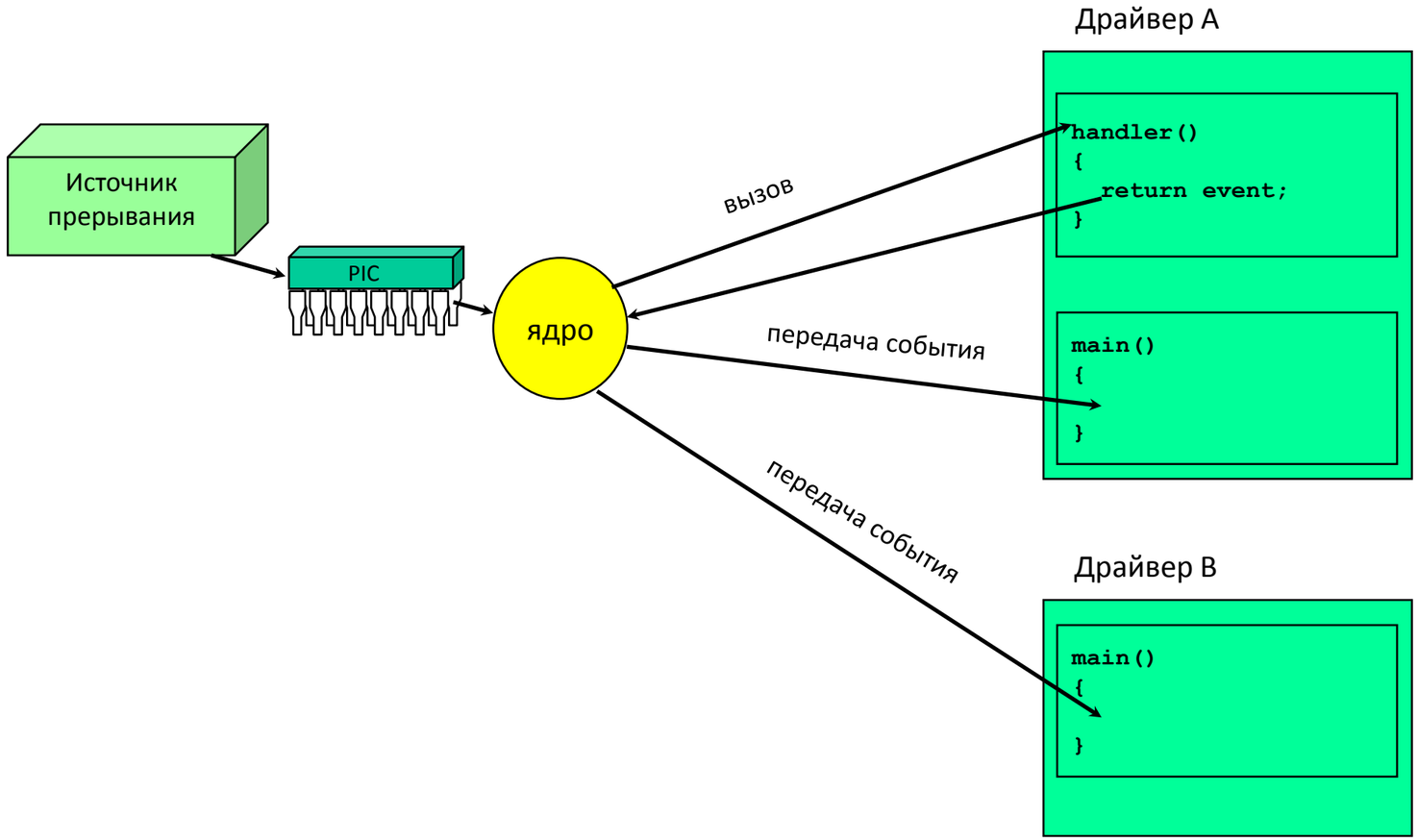


Обработка прерываний

Ядро обслуживает все аппаратные прерывания.

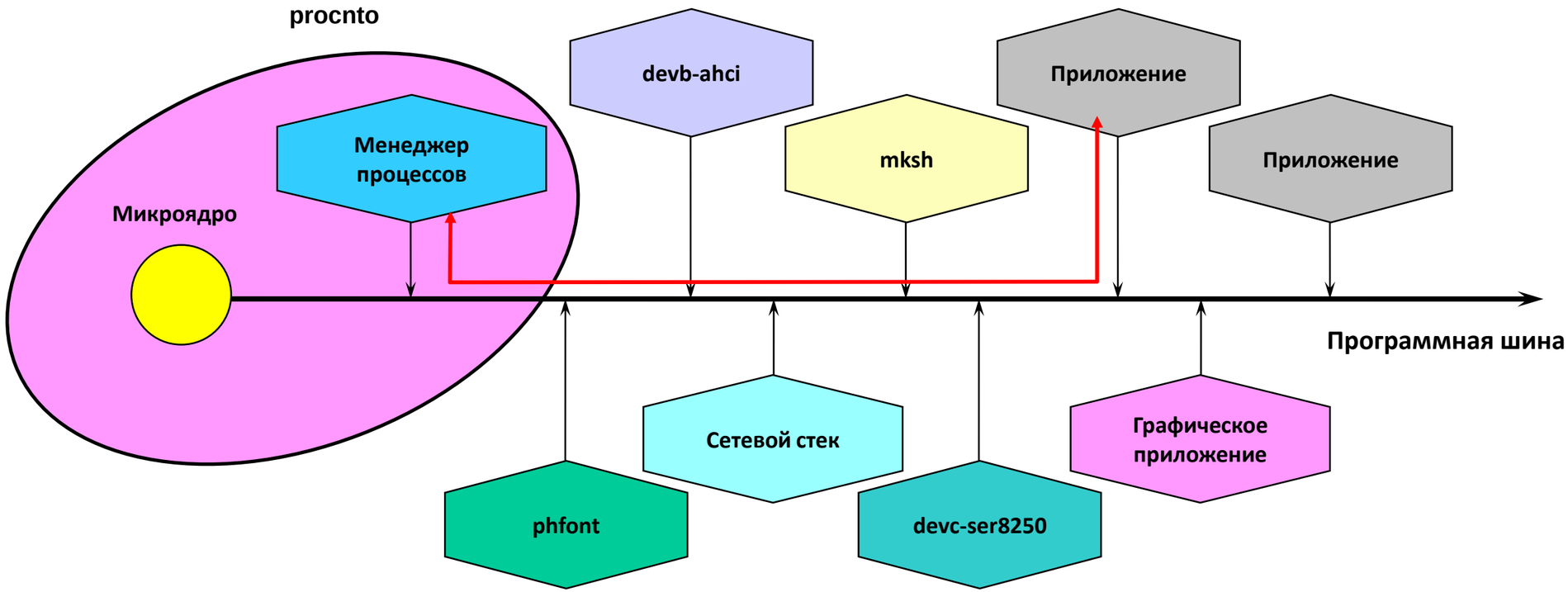
Процесс может:

- Зарегистрировать функцию-обработчик, которую ядро будет вызывать при прерывании
  - Выполняется в контексте ядра
  - Ограниченный функционал
  - Обработчик должен быть максимально коротким
- Получать уведомление о возникновении прерывания
  - Выполняется в контексте процесса
  - Полный функционал



Менеджер процессов

- **Модуль ядра «Нейтрино» procnto** состоит из двух компонентов:
  - менеджер процессов (process manager)
  - микроядро
- оба компонента имеют общее адресное пространство, но являются отдельными модулями
- администратору процессов можно отправлять сообщения



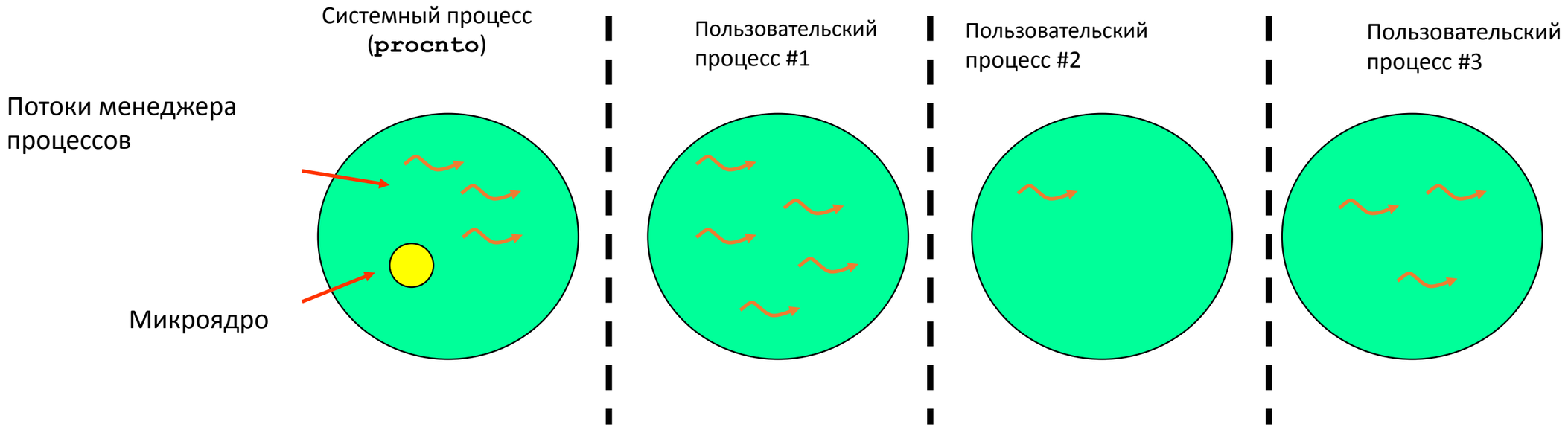
Менеджер процессов

- объединяет группы потоков в процессы
- защищает память и управляет ей (в том числе общей памятью для межпроцессного взаимодействия)
- управляет путевыми именами (/, /dev, /proc)
  - Позволяет менеджерам ресурсов регистрировать свои префиксы (/my/prefix)
- создает и завершает процессы:
  - `spawn*()` / `exec*()` / `fork()`
  - загружает исполняемые ELF-файлы
- поток ожидания (`idle`) — работает, когда в системе нет других готовых к выполнению потоков

Менеджер процессов: управление памятью

Используется виртуальная адресация:

- каждый процесс выполняется в своем защищённом адресном пространстве
- указатели, с которыми ведётся работа, содержат виртуальные, а не физические адреса
- используется одно физическое адресное пространство



Диспетчеризация

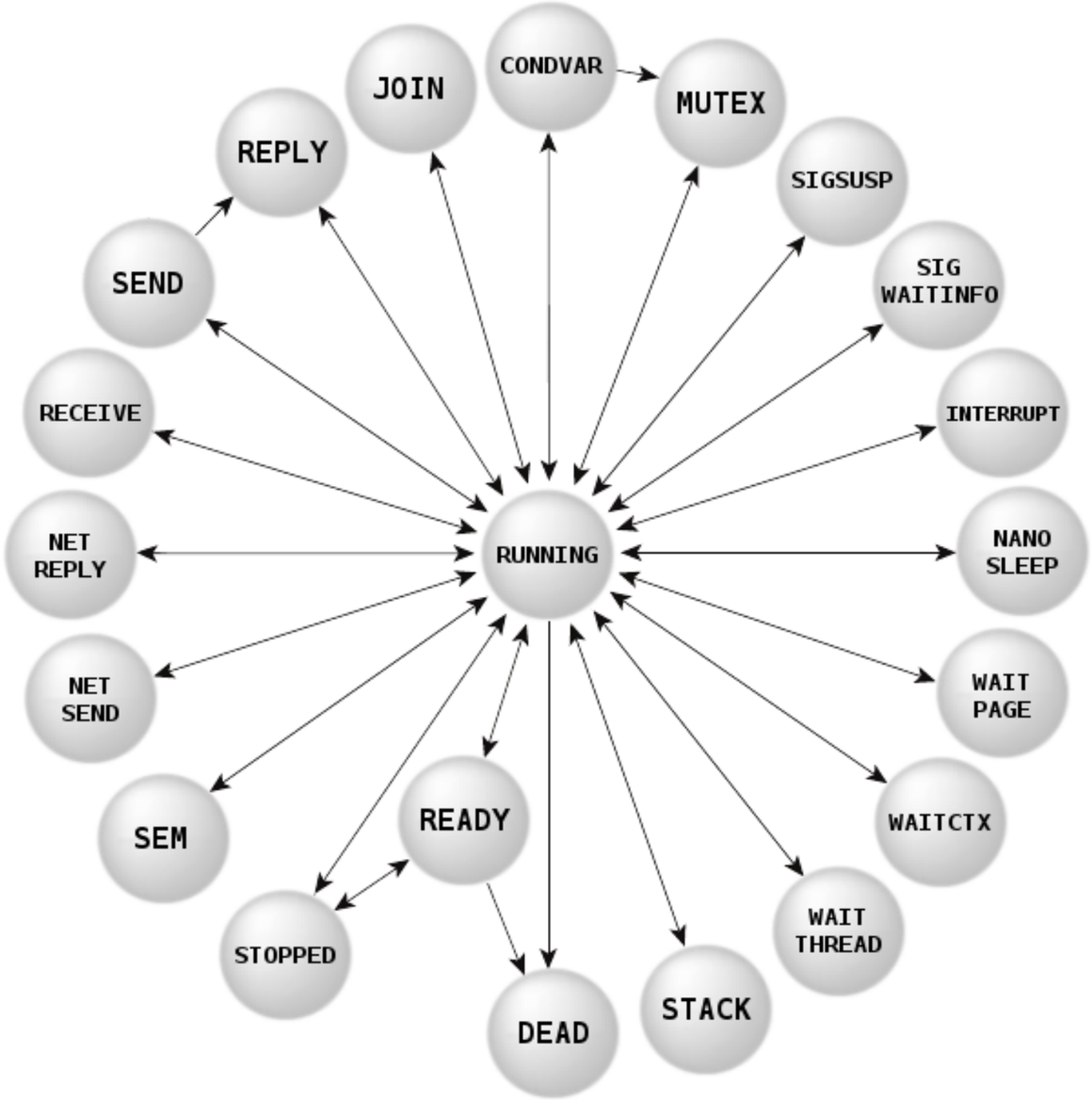
Потоки находятся в двух основных состояниях:

Блокированные:

- в ожидании какого-либо события
  - **REPLY** — ожидание ответа от другого потока
  - **MUTEX** — ожидание освобождения мьютекса
  - **RECEIVE** — ожидание поступления сообщения
  - …

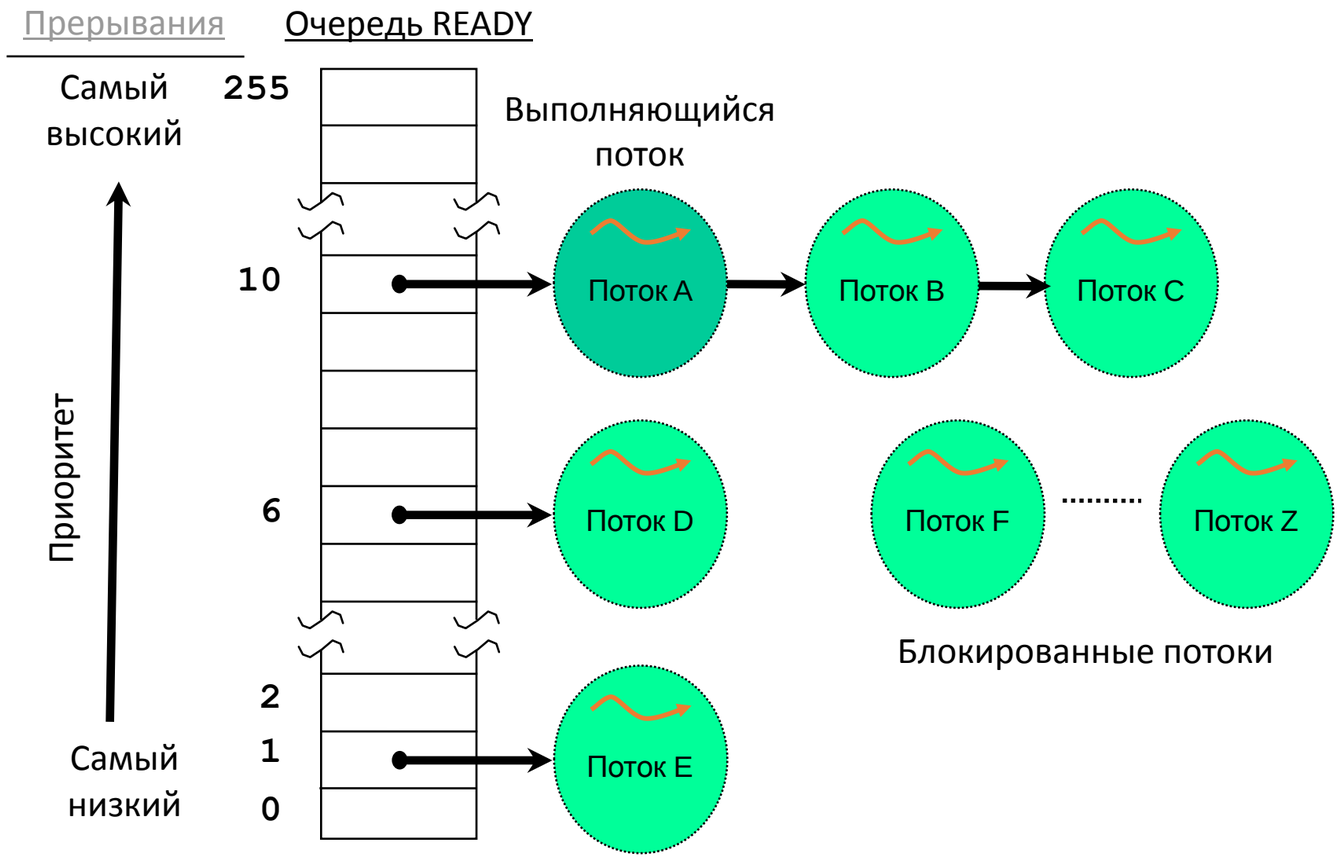
Неблокированные:

- способны использовать процессор
  - **RUNNING** — выполнение на процессоре
  - **READY** — ожидание доступности процессора

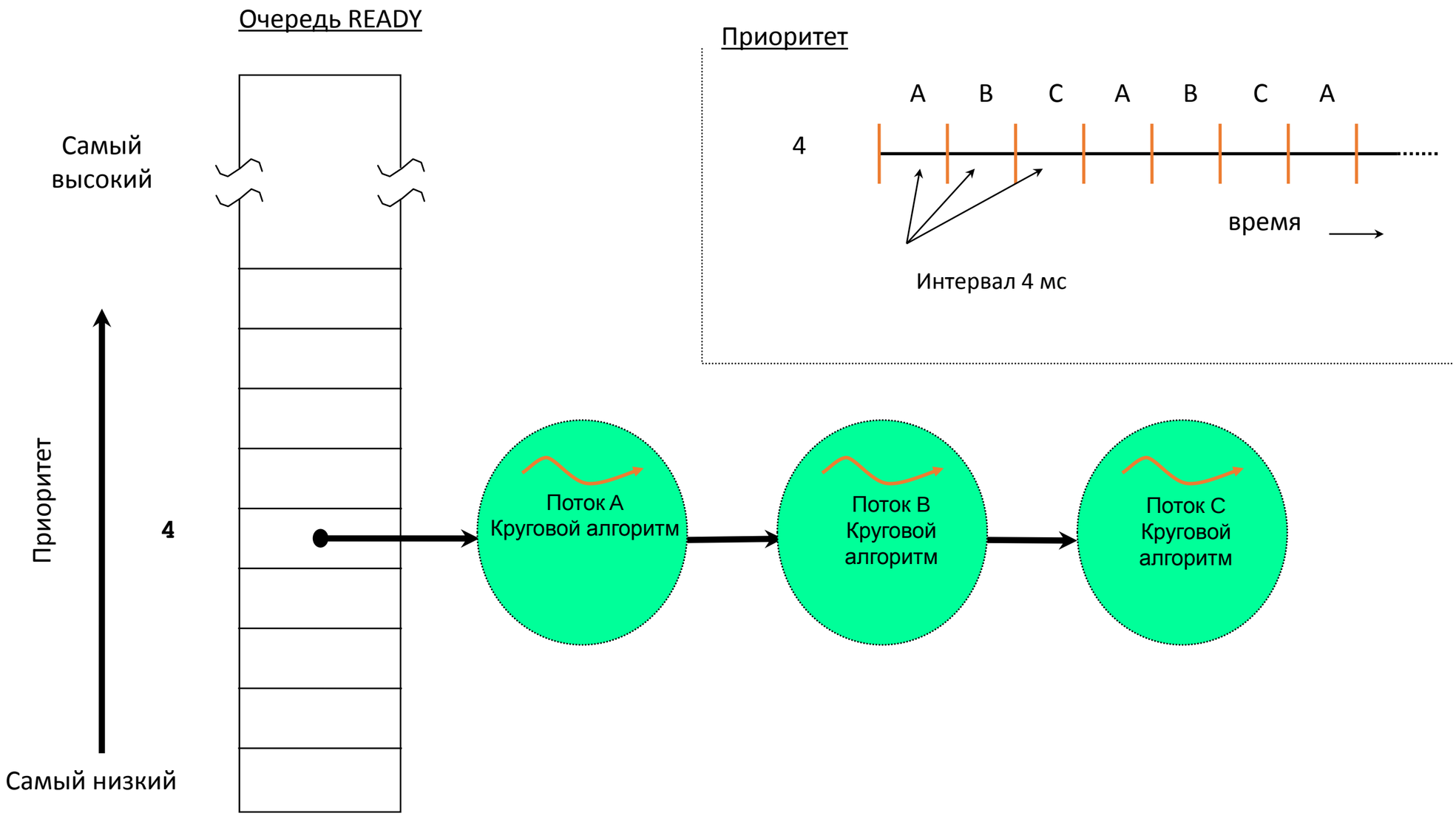


Диспетчеризация: приоритеты потоков

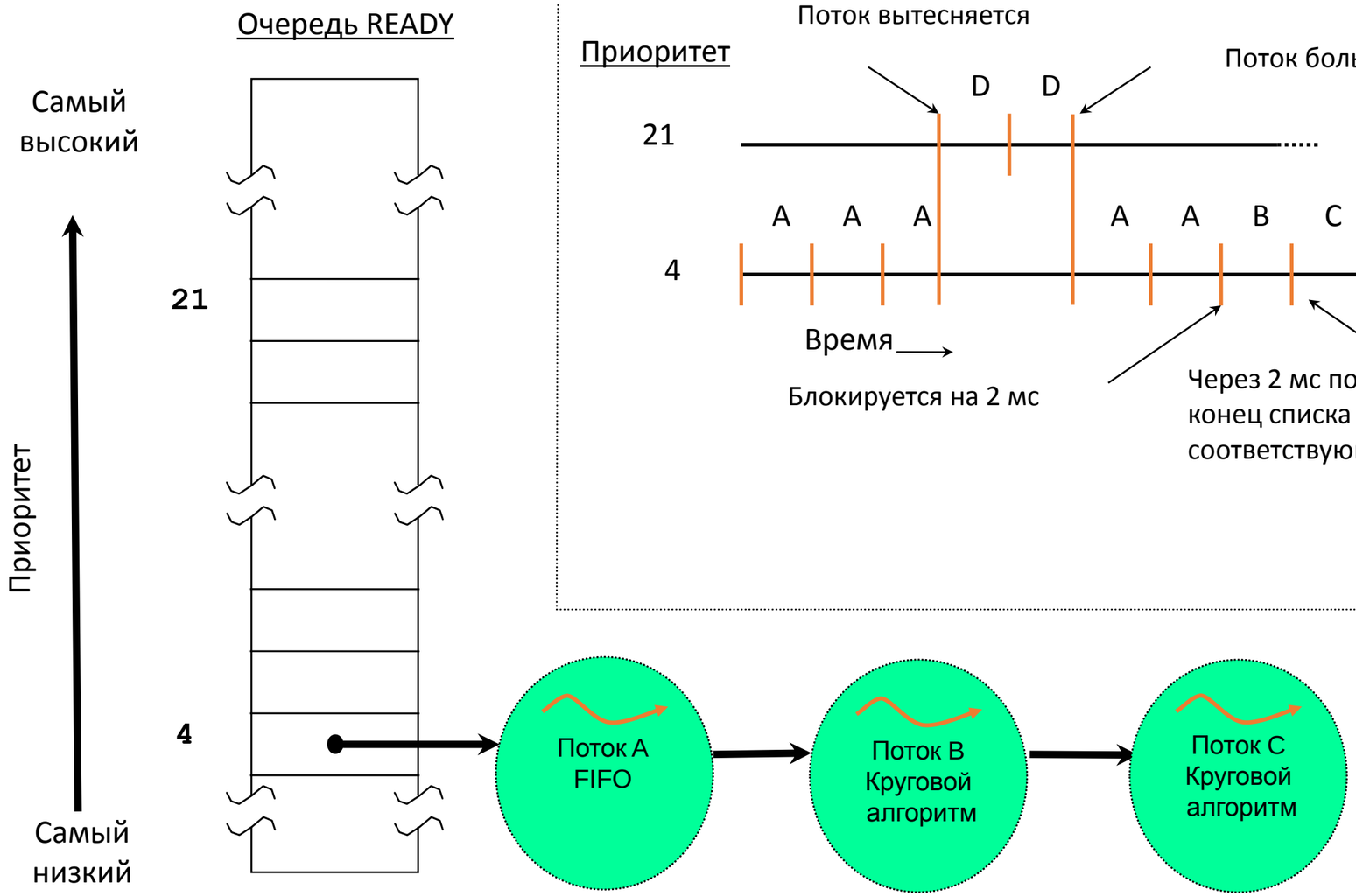
- Приоритеты: от 0 (самый низкий) до 255 (самый высокий)
- приоритет имеет значение только для потоков **READY**
- ядро всегда передает управление потоку **READY** с наивысшим приоритетом
  - поток переходит в состояние **RUNNING**
  - блокированные потоки игнорируются
- обычно потоки находятся в состоянии блокировки большую часть времени
  - так ресурс процессора распределяется между потоками



Диспетчеризация: круговой/карусельный алгоритм планирования



Диспетчеризация: алгоритм планирования FIFO



Защищенная операционная система реального времени «Нейтрино»

- **Архитектура на основе микроядра**
  - Небольшое микроядро отвечает только за минимальный функционал
  - Большинство системных служб реализованы в виде внешних процессов, включая драйверы оборудования
  - Приоритетная диспетчеризация с вытеснением
  - Отказоустойчивость, простота, надёжность, масштабируемость
- **Жёсткое реальное время**
  - Максимальное время реакции на прерывание: ~1.7 мкс (Baikal-T1), ~6.1 мкс (Эльбрус 4С)
- **Процессорные архитектуры**
  - x86, ARMv8 (AArch64), ARMv7, MIPS BE/LE, PowerPC BE/SPE, Эльбрус, RISC-V (в процессе)
- **Российские требования**
  - ФСТЭК - ОС типа А и В второго класса защиты
  - сертификат соответствия Минобороны
  - входит в реестр ПО Минцифры

Документация и полезные материалы

- Русскоязычная документация
  - Системная архитектура
  - Руководство пользователя
  - Описание базовых компонентов целевой системы
  - Руководство разработчика

- Публичный git репозиторий с примерами исходных кодов драйверов и утилит для «Нейтрино»
  - https://git.kpda.ru/explore/repos

Спасибо за внимание

Владимир Махилёв

Руководитель группы разработки

Отдел операционных систем

ул. Кузнецовская, д. 19,
г. Санкт-Петербург

+7 (812) 346-89-56

www.kpda.ru

support@kpda.ru

СВД
Встраиваемые Системы