



QNX SOFTWARE SYSTEMS



QNX Software Systems

Чарльз Иган (Charles Eagan)
Вице-президент по разработкам
ceagan@qnx.com



**QNX — выбор лидеров рынка
телекоммуникационных систем**

Теле-
коммуникации

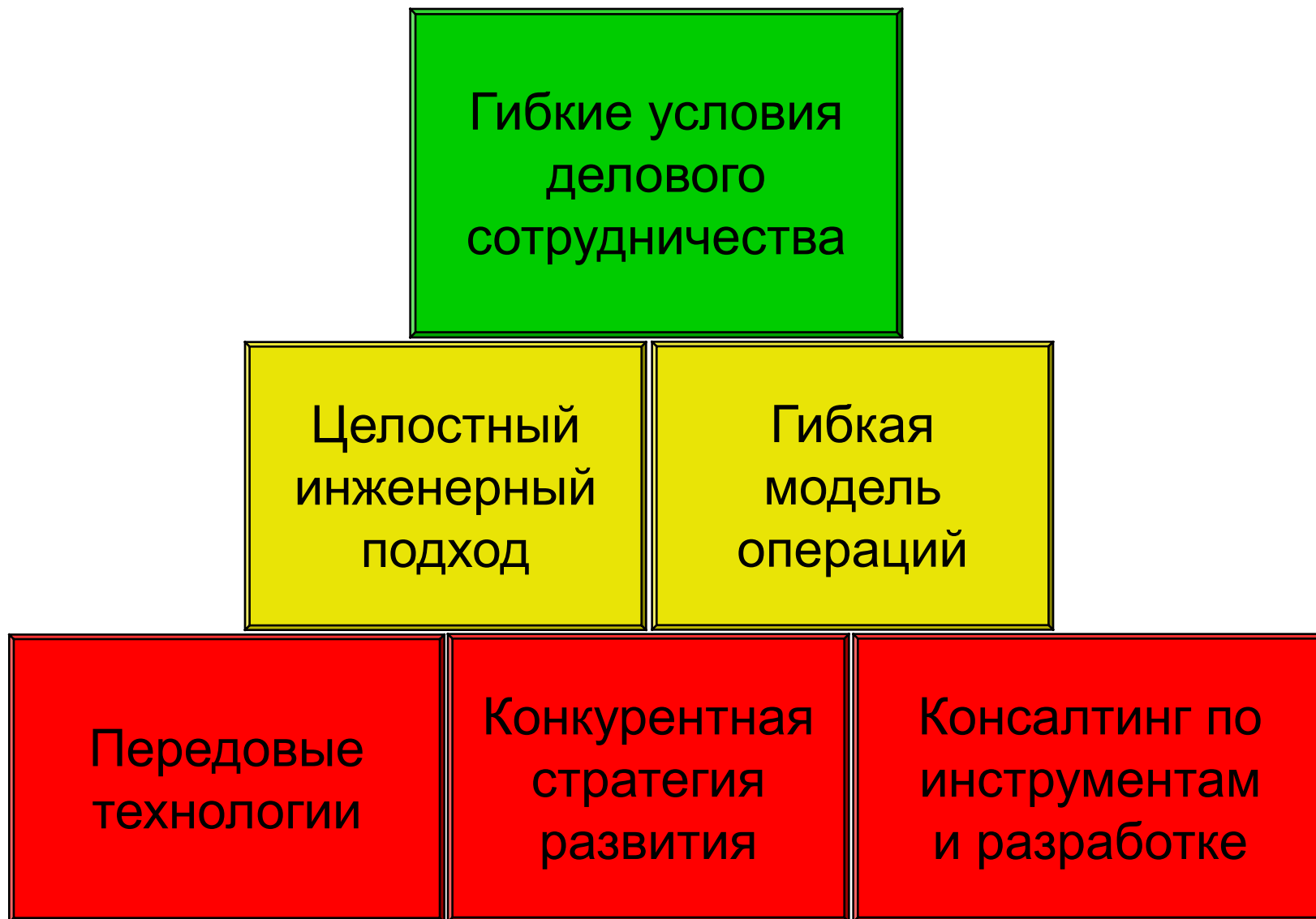


QNX — отличный партнер для
компаний в индустрии
телекоммуникаций

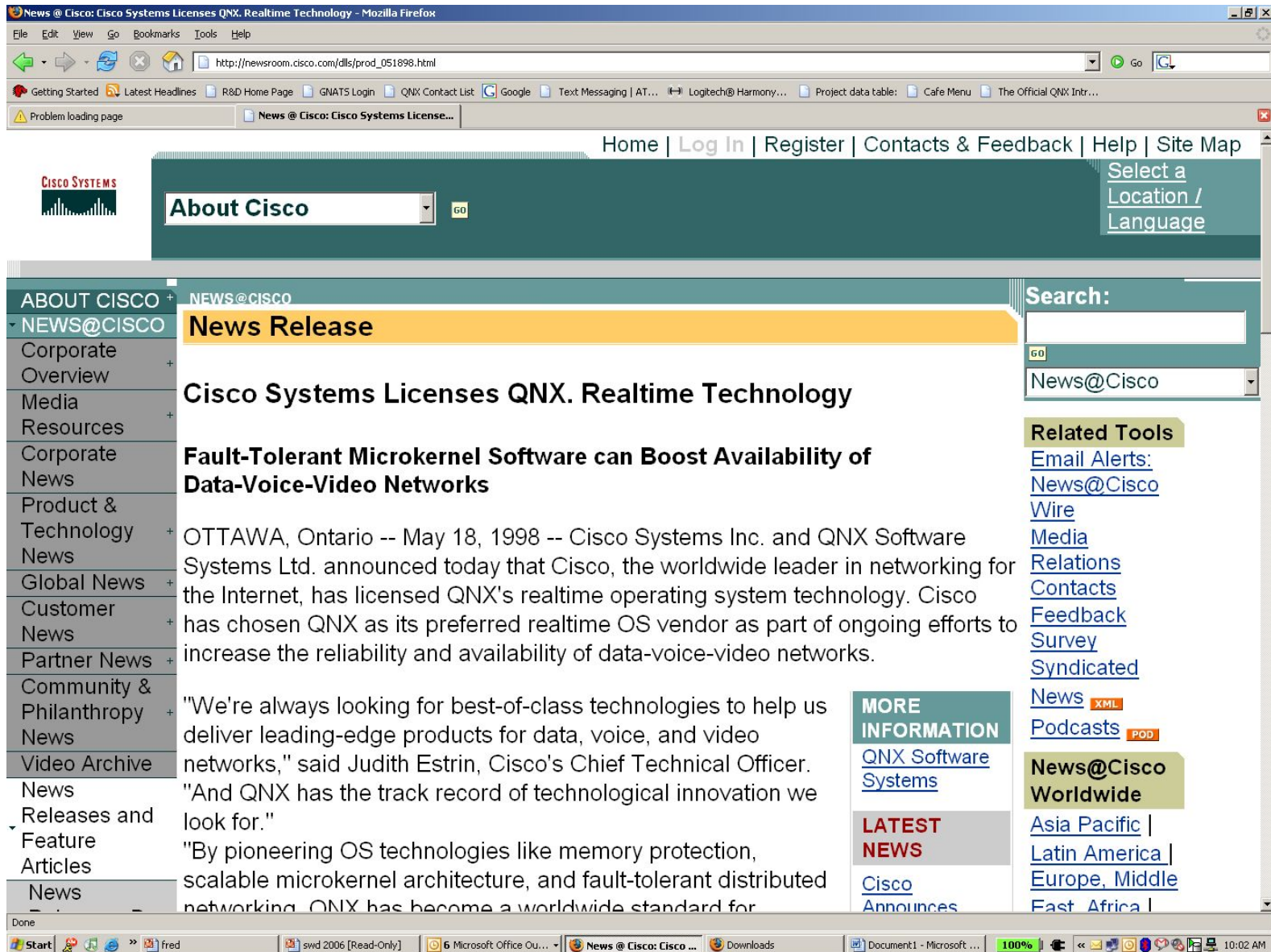
□ Как сетевая компания определяет свою стратегию в отношении операционной системы?

- ▶ Многие руководители не знают о возможности повышения производительности за счет применения коммерческой операционной системы и инструментов разработки.
- ▶ Те инженеры, которые знают об этих возможностях, зачастую не могут влиять на принятие решений.
- ▶ Период времени для качественного перехода с одной операционной системы на другую весьма ограничен.
- ▶ Для осуществления перехода требуется решительность, а также грамотное и эффективное руководство.





- **Cisco выбирает QNX в качестве стратегического партнера в период 1996/1997 гг.**
- **Интересно отметить, что в тот момент QNX-технологии были основаны преимущественно на процессорах Intel, а системы Cisco – на процессорах MIPS.**
- **Квалификация и опыт команды QNX в сочетании с эффективными инженерными решениями и согласованными рабочими подходами и планированием привели к новаторскому сотрудничеству.**
- **Применение технологий QNX привело к созданию более 10 различных рабочих групп в компании Cisco.**



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://newsroom.cisco.com/dlls/prod_051898.html`. The page content includes a navigation menu with links for Home, Log In, Register, Contacts & Feedback, Help, and Site Map. A search bar is present with the text "News@Cisco" entered. The main content area features a news release titled "Cisco Systems Licenses QNX. Realtime Technology" with a sub-headline "Fault-Tolerant Microkernel Software can Boost Availability of Data-Voice-Video Networks". The text of the release mentions that Cisco Systems Inc. and QNX Software Systems Ltd. announced the licensing on May 18, 1998. A sidebar on the left contains a navigation menu with categories like "ABOUT CISCO", "NEWS@CISCO", "Corporate Overview", "Media Resources", "Corporate News", "Product & Technology News", "Global News", "Customer News", "Partner News", "Community & Philanthropy News", "Video Archive", "News Releases and Feature Articles", and "News". A "Related Tools" section on the right lists links for "Email Alerts: News@Cisco Wire Media Relations Contacts Feedback Survey Syndicated News XML Podcasts POP". A "MORE INFORMATION" section links to "QNX Software Systems" and a "LATEST NEWS" section links to "Cisco Announces". The browser's taskbar at the bottom shows several open applications, including "swd 2006 [Read-Only]", "Microsoft Office", and "News @ Cisco: Cisco ...". The system tray shows the time as 10:02 AM.

Коммутатор
Catalyst 6k Ethernet



CRS-12000



Семейство
адаптеров
сетевых портов
Cisco

**Базовая
технология QNX
и
набор
инструментов**

CRS-1



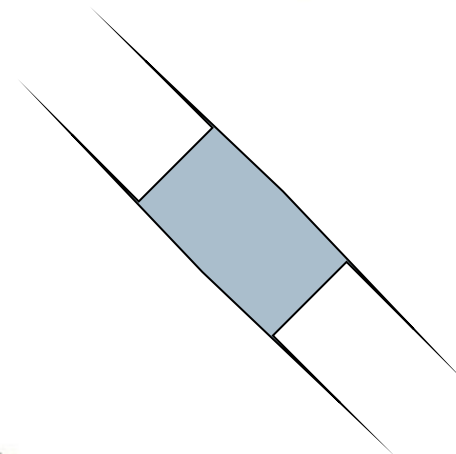
Многие другие
аппаратные
платформы



- **Высокопроизводительная архитектура высокой готовности.**
- **Гибкая, масштабируемая архитектура, полностью распределенная или монолитная.**
- **Безопасность.**
- **Применение следующих стандартов по технологиям и разработке:**
 - ▶ IEEE
 - ▶ POSIX, Unix
 - ▶ Java, C++, C, gcc
- **Гибкость**
 - ▶ Независимость от порядка следования разрядов двоичного слова (Endian abstraction)
 - ▶ Поддержка различных семейств процессоров: MIPS, PPC, Intel
- **Инструменты повышения производительности.**

Масштабируемые решения для Cisco

- QNX Neutrino для широкого круга приложений: от интерфейсов фреймеров до системы маршрутизации CRS-1
- Поддержка от одного процессора до тысячи сетевых процессоров, функционирующих как единый вычислительный ресурс

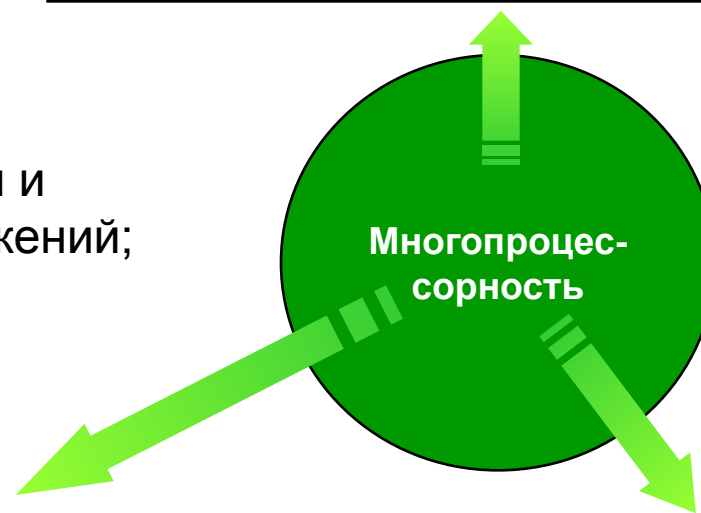


Решение QNX обеспечивает программный переход на многоядерные процессоры:

- ✓ надежная поддержка любых многоядерных архитектур;
- ✓ полный набор инструментов разработчика для профилирования и оптимизации многоядерных приложений;
- ✓ профессиональный консалтинг и обучение;
- ✓ широкий набор BSP-пакетов для многоядерных плат.

Симметричная многопроцессорность

- оптимизированные многоядерные приложения;
- разделяемое использование ресурсов на уровне ОС;
- прозрачное масштабирование свыше двух ядер.



Асимметричная многопроцессорность

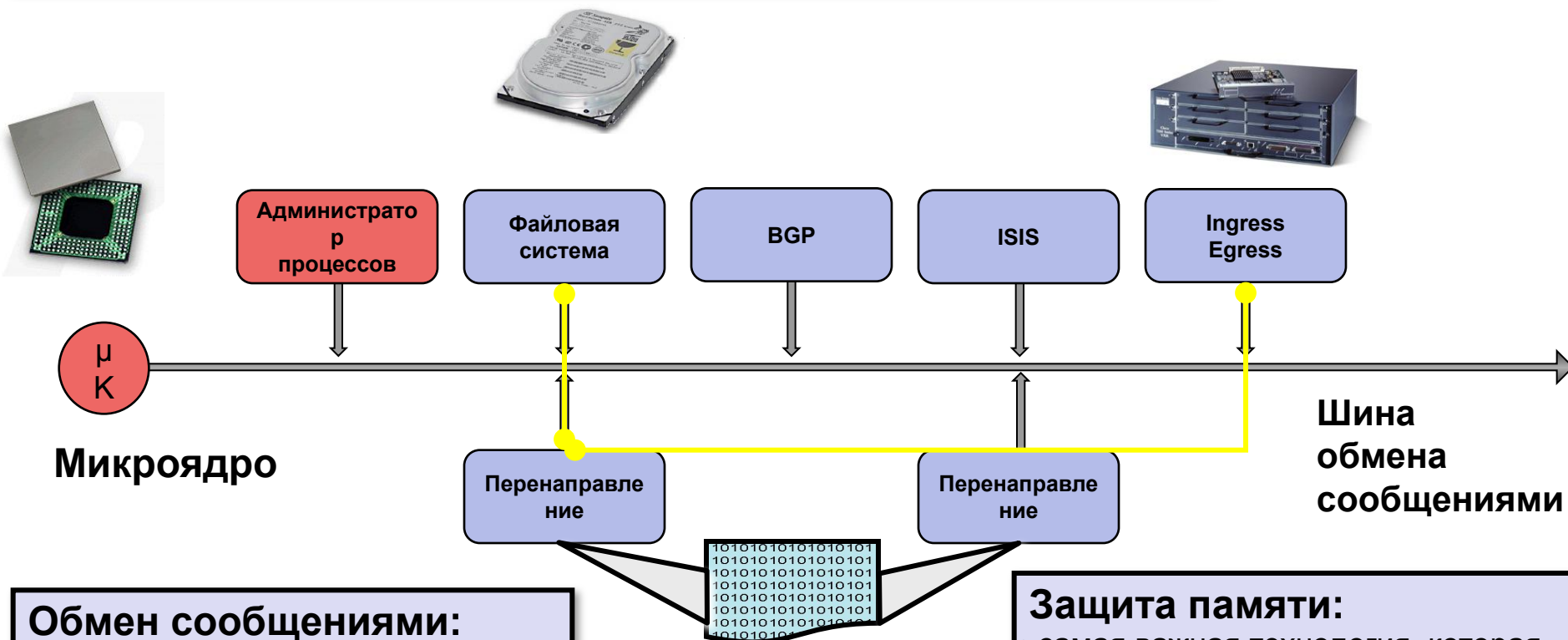
- поддержка существующей программной базы; неоптимизированные однопроцессорные приложения;
- гетерогенная прикладная база требует AMP
- нетривиальное разделяемое использование ресурсов; масштабирование свыше двух ядер достаточно сложно.

Исключительная многопроцессорность

- возможность переноса существующей программной базы;
- совместное использование существующих приложений с оптимизированными многоядерными приложениями;
- прозрачное масштабирование свыше двух ядер;
- передовая разработка компании QNX.

Высокая готовность

Самая эффективная в мире архитектура высокой готовности. Процессы взаимодействуют посредством обмена сообщениями.



Обмен сообщениями:

- прозрачное взаимодействие автономных процессов;
- POSIX-вызовы основаны на сообщениях.

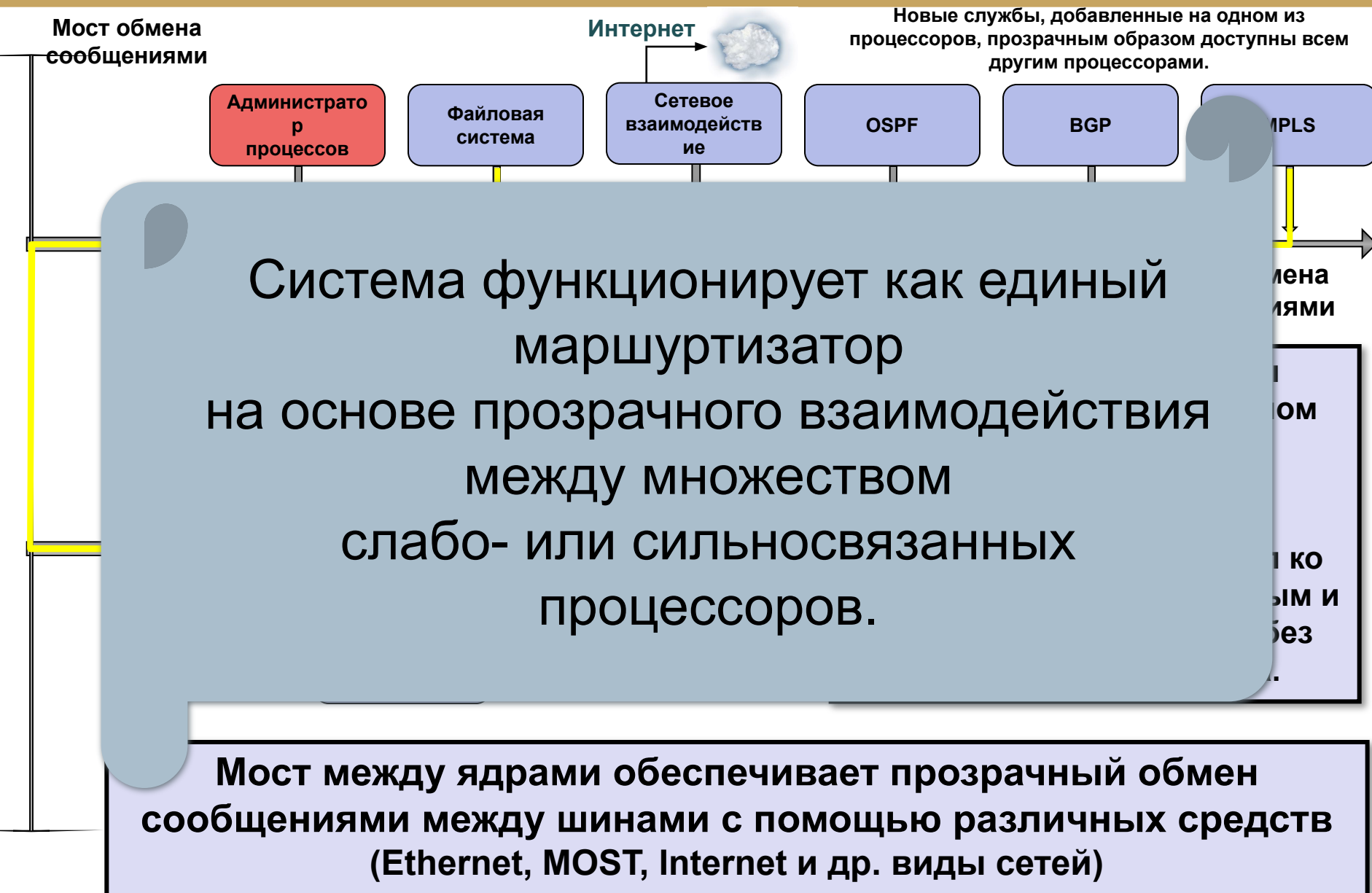
Разделяемая память:

доступ к большим массивам данных и оборудованию.

Защита памяти:

- самая важная технология, которая обязательна для настоящей системы высокой готовности;
- 90% всех системных сбоев происходят из-за конфликтов обращения к памяти.

Гибкая архитектура – полностью распределенная или монолитная



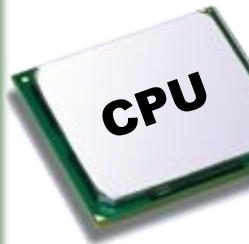
Распределенная вычислительная архитектура

Контроллер маршрутизации

Распределенный маршрутизатор

Процессор маршрутизации

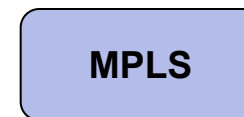
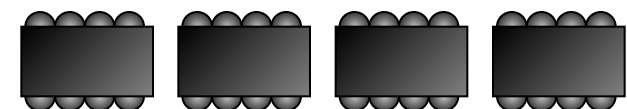
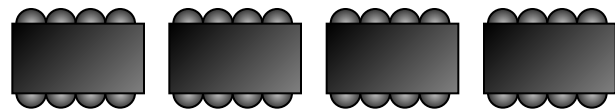
Line Card/
Forwarded Plane



SMP
BMP
AMP

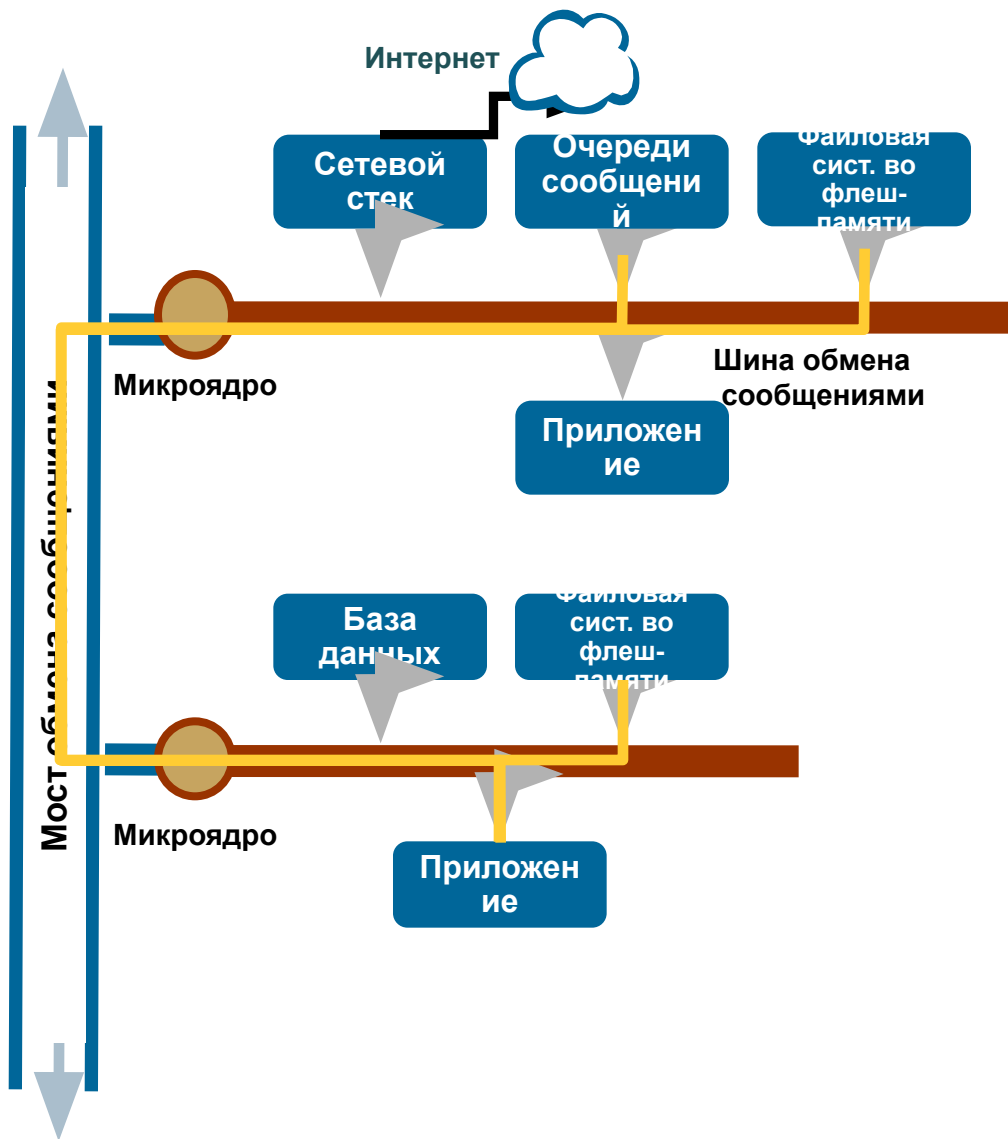
DMP

Протокол
прозрачного
безопасного
распределения



Приложения могут прозрачным образом взаимодействовать со всеми интерфейсами (line card/forwarded plane).

Автообнаружение и выравнивание нагрузки



Прозрачные распределенные вычисления:

- ▶ распределенная POSIX-модель;
- ▶ платформа для динамического взаимодействия между оборудованием и программным обеспечением на удаленных узлах;
- ▶ служба глобальных имен для обнаружения нового оборудования и приложений;
- ▶ Остановка приложений на одном узле и перезапуск на другом:
 - без необходимости перезагрузки;
 - прозрачное управление всем соединениями;
- ▶ широкое применение в CRS-1

- **Разделение привилегий:**
 - ▶ разные уровни привилегий для разных приложений;
 - ▶ приложению присваивается низший необходимый уровень привилегий.
- **Полный контроль безопасности:**
 - ▶ тотальная проверка на доступ.
- **Установки по умолчанию для отказоустойчивости:**
 - ▶ низший необходимый уровень привилегий/доступа по умолчанию.
- **Разработка:**
 - ▶ принципы “объектно-ориентированной” разработки;
 - абстрагирование, модульность, инкапсуляция, изоляция.
 - ▶ важно, если эти принципы поддерживаются на уровне ОС
- **Защита ресурсов на прикладном уровне:**
 - ▶ память, процессорные циклы, регистры, периферия и т.д.
- **Архитектура ОС может значительно влиять на работу этих принципов и даже возможность их применения.**

Высокая готовность: защита от сбоев и восстановление

- Способность восстановления характеризуется уровнем “готовности”:
 - ▶ Вероятность того, что система или подсистема выполнит предназначенную функцию в заданный период времени.

$$\text{Готовность} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

- ▶ **MTBF** – среднее время наработки на отказ
 - ▶ **MTTR** – среднее время до восстановления
- Готовность на уровне 99,999% (“пять девяток”) = меньше 5,25 минут простоя (запланированного и незапланированного) в год.
- Сетевые компании и аналитики ИТ-индустрии обращают особое внимание на эти показатели.

□ Повышение MTBF:

- ▶ многократное (!) тестирование и отладка;
- ▶ большинство ОС имеют множество инструментов для повышения MTBF.

□ Также сокращение MTTR:

- ▶ выявление, ограничение и исправление ошибок;
- ▶ готовность приближается к 100% при MTTR, стремящемся к 0.

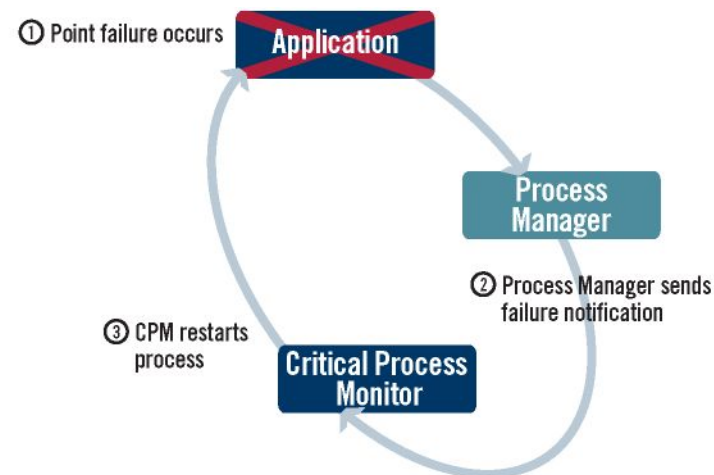
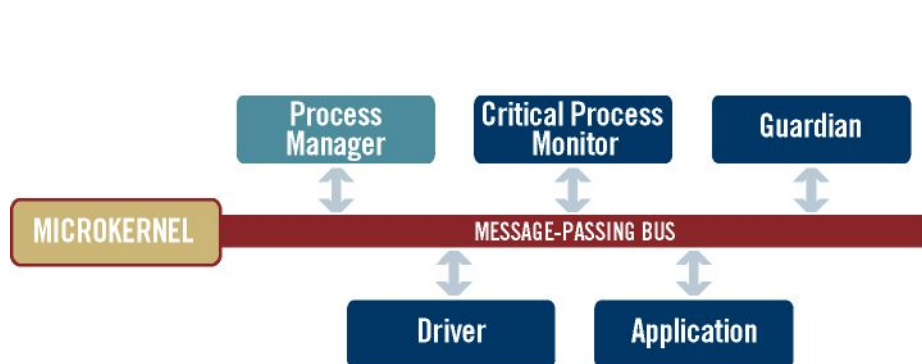
□ Сценарии восстановления:

- ▶ системная перезагрузка (исполняемый модуль реального времени, монолитное ядро);
 - восстановление за период от неск. секунд до неск. минут;
- ▶ службы перезапуска (микроядро, монолитное приложение);
 - восстановление за миллисекунды (<< 1 сек.).

□ Сочетание микроядра с системой восстановления:

- ▶ упрощает достижение высокой готовности на уровне “пяти девяток”.

Система высокой готовности – Монитор ключевых процессов



- Разработана в сотрудничестве с Cisco (ключевым клиентом компании QNX).
- Система высокой готовности:
 - ▶ пользовательские сценарии восстановления;
 - ▶ автоматическое прозрачное восстановление системы за короткое время – обеспечение минимального периода простоя.

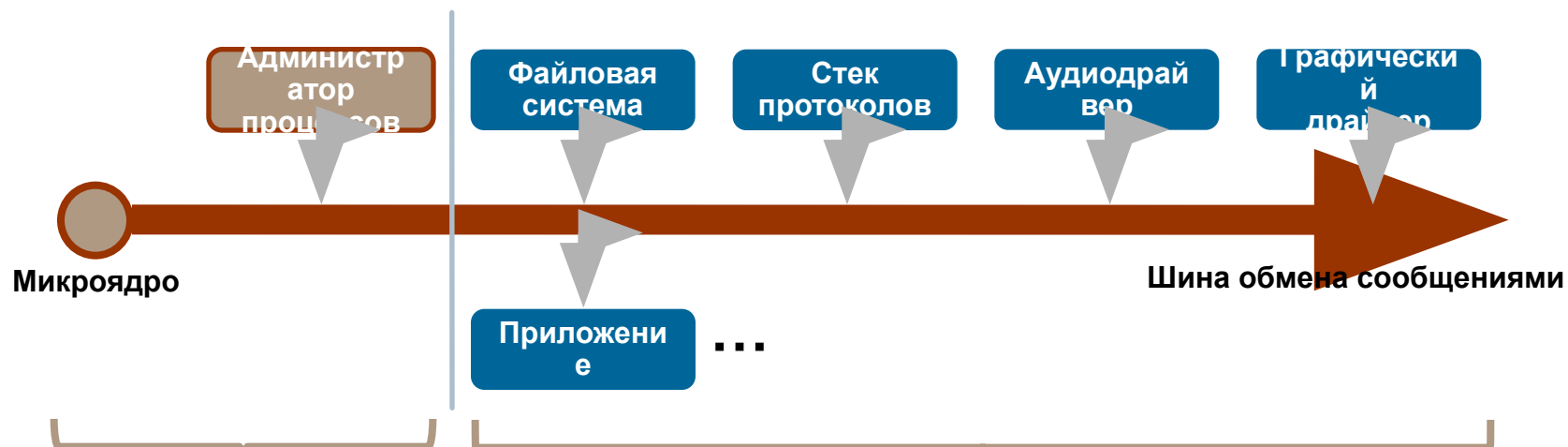
Система высокой готовности – Монитор ключевых процессов (МКП)



- Система высокой готовности и восстановления (монитор ключевых процессов) отслеживает состояние компонентов и производит восстановление при их сбоях.
- Дублирующий процесс обеспечивает замену процесса высокой готовности при его сбое и защищает от возникновения единой точки сбоя (SPoF).
- Клиентская библиотека позволяет быстро и прозрачно переподключать компоненты:
 - ▶ пользователь может ввести информацию о состоянии и настроить процедуру восстановления.
- Службы генерации квитанций работоспособности выявляют зависание компонентов, что обеспечивает самомониторинг системы.



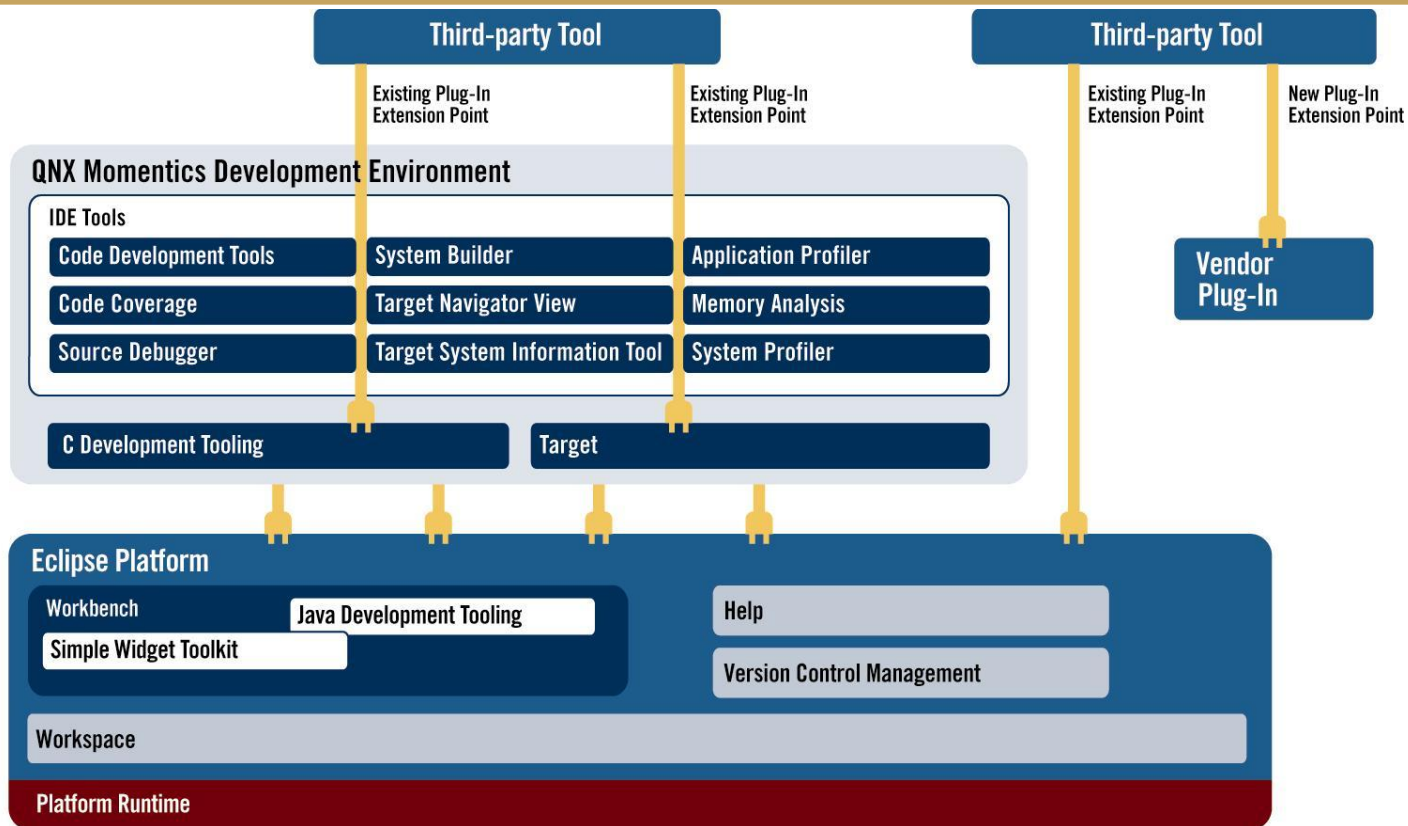
1. Сбой драйвера из-за недопустимого обращения к памяти вне защищенной области.
2. Ядро извещает монитор ключевых процессов о сбое.
3. Сбор отладочной информации о неисправном процессе.
4. Завершение драйвера и возвращение всех ресурсов системе.
Удаление канала обмена сообщениями.
5. Монитор ключевых процессов перезапускает новый драйвер.
6. Повторное создание каналов обмена сообщениями посредством клиентской библиотеки монитора ключевых процессов.
7. Драйвер запрашивает у монитора информацию о последнем сохраненном состоянии. Работа восстановлена.



Микроядро –
единственный
доверяемый
компонент

- Приложения, файловые системы и драйверы:
 - ▶ существуют как процессы на шине обмена сообщениями;
 - ▶ находятся в защищаемом адресном пространстве;
 - ▶ **запуск, остановка, добавление, удаление, перемещение и обновление компонентов без перезагрузки;**
 - ▶ невозможность нарушить работу других компонентов.

Momentics: лидер и основатель Eclipse



Масштабируемость, надежность и высокая производительность

Интегрированная поддержка:

- различных типов инструментальных, целевых систем, языков программирования и BSP-комплектов;
- оптимизирующих компиляторов;
- совместимость со всеми плагинами третьих сторон для Eclipse.



QNX: новейшая технология адаптивной декомпозиции.

Инновационное решение для телекоммуникационных
приложений

□ Что такое адаптивная декомпозиция?

- ▶ Новая технология QNX – расширение OCPV Neutrino.
- ▶ Позволяет создавать безопасные группы (объединения) из нескольких приложений или потоков.
- ▶ Гарантированное выделение процессорного времени приложениям, входящим в группу. Управление гарантиями происходит на основе бюджетов.

□ Почему адаптивная?

- ▶ Запатентованная технология обеспечивает эффективное (без потерь) выделение всех имеющихся процессорных циклов именно тем группам, которым они требуются.
- ▶ Повышение производительности за счет оптимального использования процессорных ресурсов и перераспределения нагрузки.

□ Легкий старт:

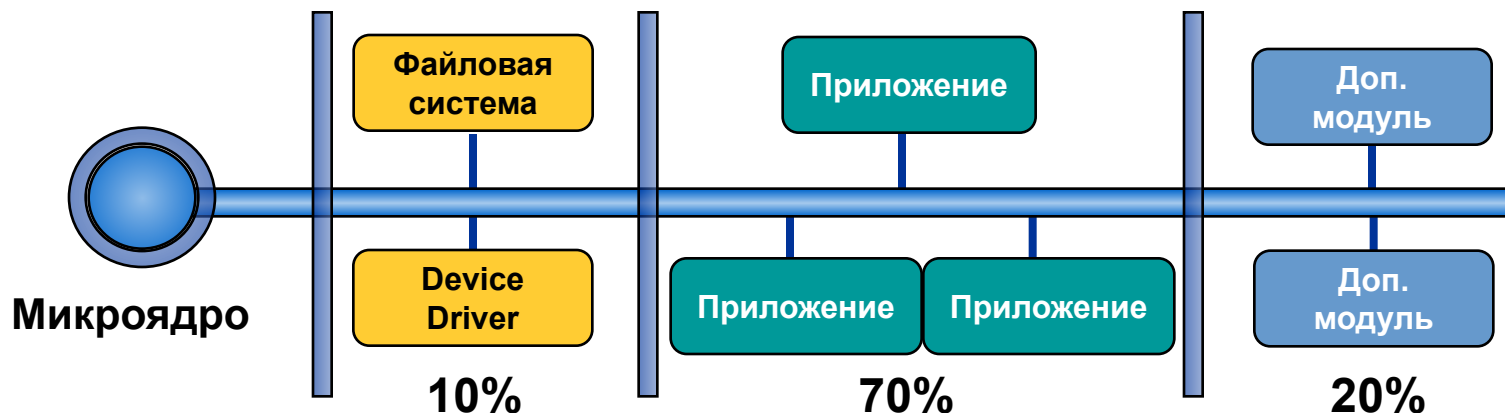
- ▶ разработчикам не нужно переучиваться:
 - программная модель POSIX сохраняется для знакомых методов разработки, программирования и отладки.
- ▶ для реализации группы не требуется вносить изменения в код.

Микроядро QNX
Neutrino



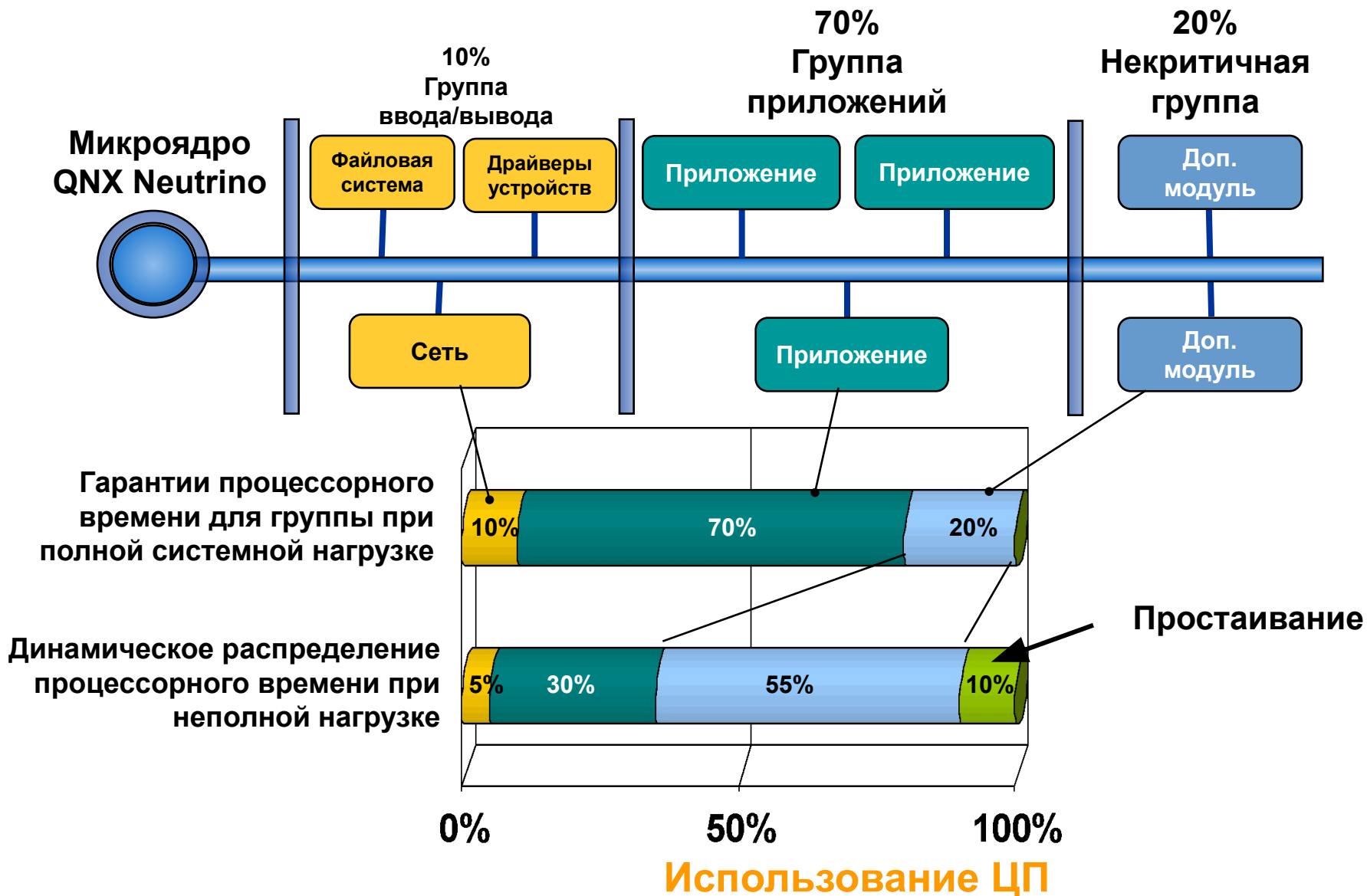
Приложения и драйверы:

- > являются процессами, которые соединяются с шиной обмена сообщениями;
- > находятся в собственном защищенном адресном пространстве;
- > никак не могут нарушить работу ядра или других компонентов;
- > запуск, остановка, модернизация на лету;
- > при сбоях драйверов не требуется перезапуск системы.



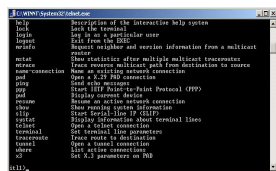
- **ОСРВ QNX® Neutrino® обеспечивает базовую структуру:**
 - ▶ инкапсуляция приложений и служб ОС на основе обмена сообщениями;
 - ▶ надежность и безопасность благодаря аппаратной защите памяти.
- **Адаптивная декомпозиция служит расширением микроядро Neutrino для создания безопасных групп и обеспечения гарантий выделения процессорного времени:**
 - ▶ группа состоит из набора процессов и потоков;
 - ▶ группе назначается изменяемый процент процессорного времени в зависимости от средней нагрузки на процессор в течение некоторого временного периода;
 - ▶ действует в дополнение к существующей схеме планирования потоков.

Максимальная производительность

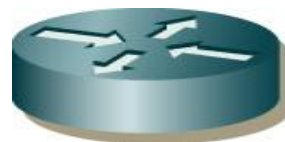


Что значит “адаптивность”?

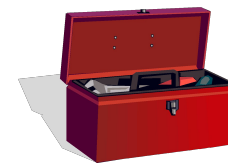
Интерфейсы управления (CLI, SNMP)



Маршрутизация и перенаправление

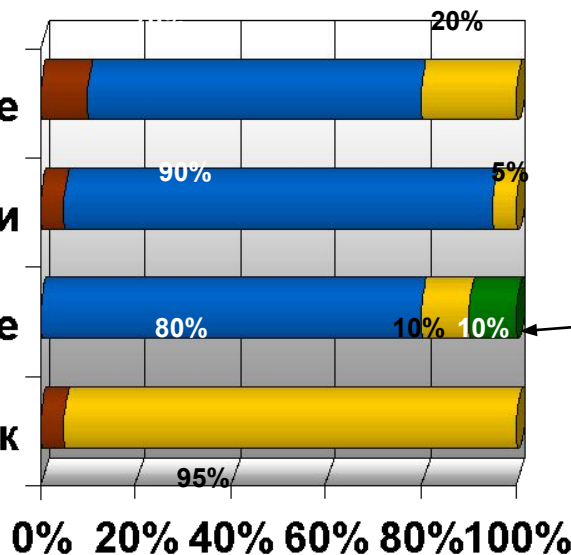


Служебные утилиты



Сценарии обработки нагрузки

Переконфигурирование
Изменение топологии
Стабильное состояние
Системный перезапуск



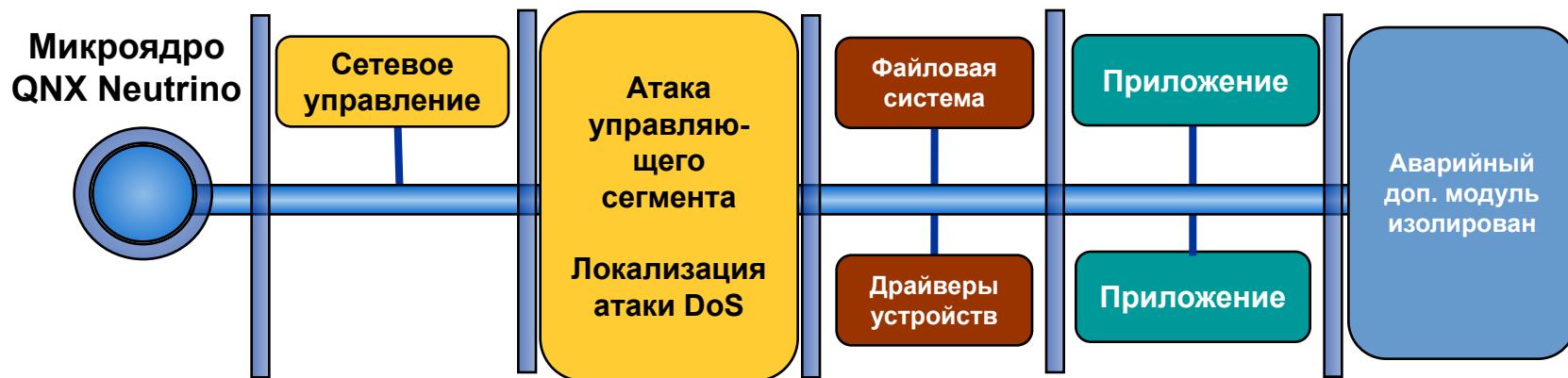
← Простаивание

□ Без группирования:

- ▶ аварийные приложения могут лишить базовые приложения процессорных ресурсов;
- ▶ распределенные DOS-атаки могут нагрузить систему задачами сетевой обработки.

□ При возможности группирования:

- ▶ создание групп с фиксированным бюджетом для надежной защиты важнейших системных ресурсов;
- ▶ ограничение угроз и защита базовых приложений и служб.



Гарантии выделения процессорного времени

10%

50%

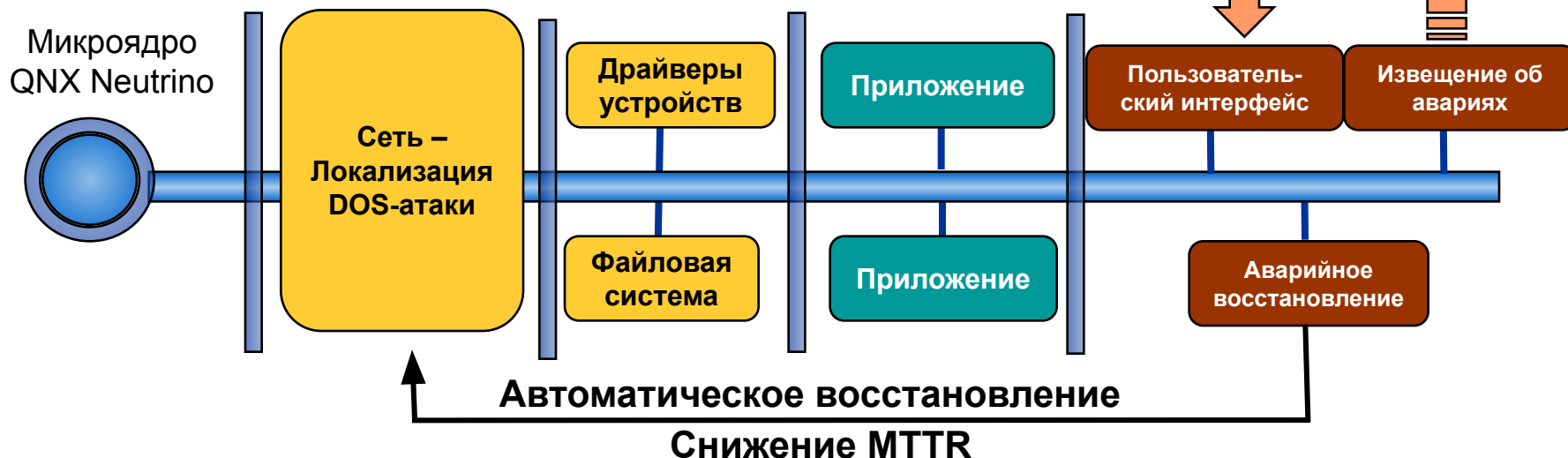
5%

25%

10%

Гарантии выделения процессорного времени: повышение уровня готовности

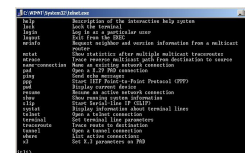
- **Гарантированный процессорный ресурс для операций восстановления:**
 - ▶ аварийные компоненты изолируются и не могут влиять на процессы восстановления.
- **Гарантированный процессорный ресурс для отправки извещений и выполнения действий пользователя:**
 - ▶ удаленные пользовательские интерфейсы сохраняют работоспособность при любых обстоятельствах.



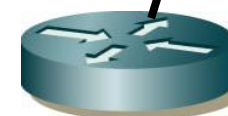
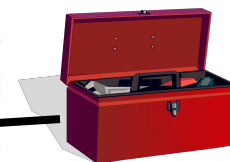
Сложность программного обеспечения Взгляд со стороны разработчиков

- **Большие команды, географически распределенная разработка**
 - ▶ возможность совмещения различных географических и временных зон.
- **Распределение обязанностей, функций и областей компетенции:**
 - ▶ сочетание различных наборов навыков разработки
- **Лицензирование и интеграция технологий третьих сторон для сокращения затрат на разработку:**
 - ▶ невозможность контролировать разработку технологий третьих сторон.
- **Параллельная разработка с последующей системной интеграцией и испытаниями**

Управляющие
интерфейсы



Обслуживание



Маршрутизация и
перенаправление

- **Системная интеграция – важный этап в проекте по разработке:**
 - ▶ всегда является критической фазой.

- **Проблемы, обнаруженные на поздних этапах в цикле разработки, порождают самые большие затраты:**
 - ▶ затраты на первичное тестирование для выявления ошибок;
 - ▶ как правило, приводит к задержкам всего проекта;
 - ▶ требуются системные инженеры для выявления и устранения неисправностей;
 - ▶ затраты на повторную реализацию и тестирование.

- **Изменения проекта на поздних этапах приводят к повышению рисков:**
 - ▶ как правило, "простые" решения применяются для снижения трудоемкости и соблюдения сроков сдачи проекта;
 - ▶ в итоге снижается качество и производительность продукта.

- **На этапе интеграции типичные проблемы связаны с производительностью, ошибками памяти и зависаниям процессов.**

Компания QNX продолжает развитие на тех рынках, которые уже принесли успех, и весьма активно осваивает новые рынки для создания основы для будущего роста.

Наши планы технологического развития нацелены на сохранение лидерства и удовлетворение требований целевых рынков.

