



Сергей Прожогин

Neutrino: быстрее, выше и... меньше

Вниманию читателей предлагается описание особенностей и возможностей операционной системы реального времени QNX/Neutrino.

QNX/Neutrino появилась не на пустом месте: ей предшествовала ОС QNX, которая была первой коммерческой ОС, построенной на принципах микроядра и обмена сообщениями. Данная ОС реализована в виде взаимосвязанных, но в то же время независимых (взаимодействие посредством обмена сообщениями) процессов различных уровней. ОС QNX содержит в себе предсказуемость, что крайне необходимо для решения задач жесткого реального времени, масштабируемость и эффективность в управлении ресурсами, расширяемость, проявляющаяся в добавлении своего драйвера устройства без перекомпилирования ядра, собственный сетевой протокол FLEET, по скорости приближающийся к физической скорости передачи данных. Тем не менее, QNX имеет ряд недостатков и ограничений из-за своей ориентации на аппаратную архитектуру Intel x86, а также из-за строгого соответствия требованиям ОС реального времени (ОС РВ). В число этих недостатков входят:

- отсутствие поддержки SMP (многопроцессорных систем);
- отсутствие свопинга виртуальной памяти на диск;
- нестандартная поддержка нитей;
- много ограничений файловой системы;
- отсутствие Unix-domain sockets.

Если обобщить все проблемы QNX, то получатся четыре основных пункта:

- 1) недостаточная поддержка POSIX;
- 2) отсутствие механизма поддержки стандартных нитей;
- 3) отсутствие поддержки SMP;
- 4) непереносимость на другие аппаратные платформы.

После знакомства с этими пунктами становится очевидно, что цель, которой пытается достичь QSSL (<http://www.qnx.com>) — разработчик QNX/Neutrino, состоит в создании POSIX совместимой ОС, пригодной для использования как в интеллектуальных контроллерах, так и в мощных серверах и кластерах.

ПОДДЕРЖКА POSIX-СТАНДАРТА

Принято считать, что поддержка POSIX-стандарта операционной системой делает ее слишком громоздкой для встраиваемых систем. Тем не менее, QNX/Neutrino является POSIX совместимой ОС (в отличие от QNX, слабо поддерживающей POSIX) для встраиваемых систем реального времени. QNX/Neutrino обладает высокой эффективностью, присущей предыдущим версиям QNX, и включает в себя стандартный программный интерфейс, что должно существенно облегчить разработку приложений и перенос их с других платформ.

В микроядре Neutrino реализованы основные функции POSIX:

- 1003.1, 1003.1a — управление процессами, вводом-выводом и файловой системой;
- 1003.1b — выполнение в реальном масштабе времени;
- 1003.1c — создание и управление нитями в рамках процесса;
- 1003.1d — расширения ОС РВ (обработчики прерываний и т.п.).

ПОДДЕРЖИВАЕМЫЕ АППАРАТНЫЕ ПЛАТФОРМЫ

На данный момент QNX/Neutrino поддерживает основные аппаратные платформы, используемые во встраиваемых системах, включая

- семейство x86 — 386, i386 EX, Am386SE/DE, AMD ElanSC400/410, 486, Cyrix MediaGX, Pentium, Pentium Pro, Pentium II, STPC;
- семейство PowerPC — 401, 403, 603e, 604e, 750, MPC860, MPC821, MPC823;
- семейство MIPS — R4000, R5000, NEC VR4300/4102/4111, VR5000, IDT R4700, QED RM5260/5270/5261/5271.

ОСНОВНЫЕ ПОНЯТИЯ QNX/NEUTRINO

Ключевыми свойствами, делающими ОС QNX/Neutrino крайне привлекательной для использования во встроженных системах, являются:

- многозадачность;
- управляемая приоритетами диспетчеризация;
- минимальные накладные расходы на обработку аппаратных прерываний и переключение задач;
- масштабируемость;
- прозрачная сетевая поддержка;
- соответствие стандарту POSIX;
- поддержка многопроцессорных систем;

МИКРОЯДРО QNX/NEUTRINO

Уникальность QNX/Neutrino заключается в эффективности использования вычислительных ресурсов, в модульности и простоте. Основные принципы построения ОС:

- архитектура «микроядро»;
- межпроцессорное взаимодействие, в основу которого положен принцип обмена сообщениями.

Архитектура «микроядро» означает, прежде всего, построение ОС по принципу модульности, а размер ядра игра-

ет не главную роль. Функциональные возможности такой системы наращиваются не путем добавления новых функций в ядро, а за счет подключения новых процессов, обеспечивающих дополнительный сервис. Поэтому поддержка файловой системы, сетевой обработки, графического интерфейса и других сервисов не является функцией микроядра, как в большинстве ОС. Эти сервисы обеспечиваются отдельными процессами. Целью создания ОС на основе микроядра является не желание сделать ее как можно меньше по размеру, а стремление обеспечить большую гибкость за счет модульности структуры. Небольшой размер является, несомненно, положительным, но все-таки побочным эффектом.

Компактное ядро Neutrino имеет очень небольшой размер — менее 32 кбайт, выполняя нескольких основных служб операционной системы:

- службы обмена сообщениями между потоками всей системы;
- службы синхронизации потоков;
- службы диспетчеризации потоков;
- службы управления таймерами и аппаратными прерываниями.

В отличие от потоков (процессов), само ядро Neutrino не подлежит диспетчеризации. Код ядра выполняется только при вызове функции ядра каким-нибудь потоком или при обработке аппаратного прерывания. Все остальные службы операционной системы реализованы посредством стандартных процессов. Пользователь имеет возможность создавать необходимую ему конфигурацию операционной системы путем добавления или удаления необходимых в данной реализации процессов. В QNX/Neutrino не существует различия между пользовательскими и системными процессами: и те и другие используют одинаковые механизмы взаимодействия с ядром ОС. Драйверы устройств в QNX/Neutrino не отличаются от остальных процессов в системе, и их можно создавать и отлаживать, как обычные пользовательские приложения.

Микроядерная структура присутствовала и в предыду-

щих версиях ОС QNX, однако она, в отличие от Neutrino, не могла работать без менеджера процессов. Так как ядро QNX/Neutrino обеспечивает достаточный набор функций для управления потоками, то пользователь имеет возможность создать конфигурацию, состоящую из одних только пользовательских потоков, не включая в ядро менеджер процессов (рис. 1).

Все это необходимо для реализации встроенных систем реального време-

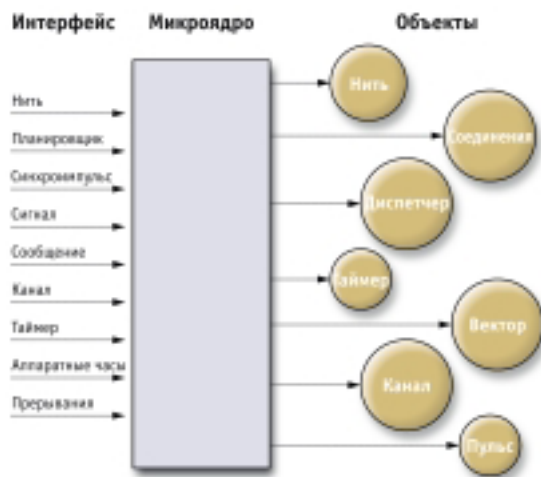


Рис. 1. Микроядро QNX/Neutrino

ни. Дополнительно к этим функциям в ядре реализован механизм обмена сообщениями, являющийся отличительной чертой ОС QNX. Введено новое понятие «пульс» — краткое служебное сообщение, не требующее ответа и используемое для асинхронных передач. Ядро управляет объектами и поддерживает 57 вызовов для поддержки потоков, сообщений, сигналов, таймеров, обработчиков прерываний, примитивов синхронизации (семафоров, критических секций, условных переменных). Все вызовы ядра являются полностью прерываемыми. Прерывания отключаются на 8 команд при входе в код ядра, на 14 команд при выходе из него и на 40 команд при возникновении исключения.

Микроядро Neutrino обеспечивает перераспределение процессорного времени между потоками (процессами) за счет различных механизмов диспетчеризации. Исполнение потока приостанавливается всякий раз при возникновении одного из следующих условий:

- поток вызывает функцию ядра;
- происходит аппаратное прерывание;
- возникает исключение.

ПОДДЕРЖКА МУЛЬТИПРОЦЕССОРНОСТИ

При работе QNX/Neutrino мы увидим почти линейный рост производительности при добавлении дополнительных процессоров (всего до 8). Это достигается за счет того, что все системные вызовы QNX/Neutrino могут вытесняться при необходимости обработать вызов от нити с более высоким приоритетом даже в процессе передачи сообщений. Такое качество микроядра в совокупности со сравнительной простотой и малым размером позволяет минимизировать невытесняемые последовательности кода в системе. Это влечет за собой ряд положительных последствий. Во-первых, улучшаются временные характеристики системы. Во-вторых, невысокие требования к памяти упрощают разработку встроенных систем низшего уровня. В-третьих, уменьшается количество блокировок в коде, необходимых для поддержки симметричных мультипроцессорных архитектур (SMP) и повышается эффективность использования дополнительных процессоров (рис. 2).

ДИСПЕТЧЕРИЗАЦИЯ ПРОЦЕССОВ

Решение о том, какой поток вновь получит управление, принимает

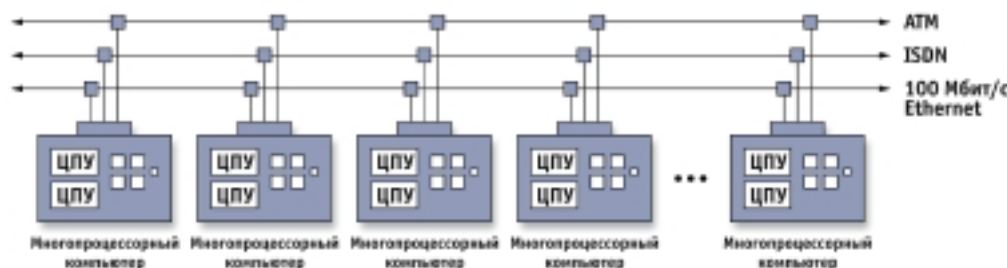


Рис. 2. Поддержка многопроцессорных конфигураций в QNX/Neutrino

QNX/Neutrino. Если управление получает поток, отличный от текущего, то производится переключение контекста активной задачи. Переключение контекста производится в том случае, если текущий поток блокируется (например, в результате вызова функции ожидания сообщений), добровольно передает управление (например, посредством вызова функции `sched_yield()`), или если в системе существует поток, выполнение которого по ряду причин является более предпочтительным (рис. 3).

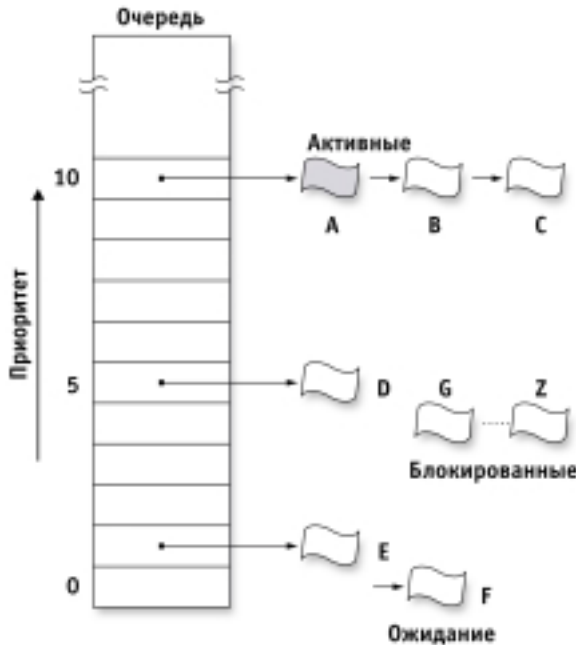


Рис. 3. Состояния и приоритеты нитей в очереди

Каждый поток имеет определенный приоритет от 0 (наименьший) до 31 (наивысший). QNX/Neutrino выбирает для выполнения поток, готовый к выполнению (незаблокированный) и имеющий наивысший приоритет.

Для обеспечения большей гибкости при перераспределении процессорного времени QNX/Neutrino использует 3 алгоритма диспетчеризации для каждой очереди потоков с одинаковым приоритетом:

- FIFO,
- Round-robin,
- адаптивный алгоритм.

Поток может установить для себя любой из данных алгоритмов, однако он будет применяться в том случае, если существует хотя бы еще один поток с таким же приоритетом и данным алгоритмом диспетчеризации. Перераспределение времени процессора, согласно одному из приведенных алгоритмов, нарушается в том случае, если поток с большим приоритетом становится готовым к выполнению, в этом случае он немедленно получает управление, прерывая выполнение менее приоритетного потока. Алгоритмы FIFO и Round-robin в комментариях не нуждаются. Адаптивный алгоритм диспетчеризации подразумевает следующее:

- если поток исчерпал выделенный ему интервал времени, то его приоритет понижается на 1 при условии,

что существует готовый к выполнению поток с таким же приоритетом; ● если поток переходит в заблокированное состояние, то его приоритет восстанавливается до первоначального значения.

УПРАВЛЕНИЕ АППАРАТНЫМИ ПРЕРЫВАНИЯМИ

QNX/Neutrino полностью обеспечивает управление аппаратными прерываниями. Для обработки прерывания приложению требуется привязать необходимую функцию обработки к искомому прерыванию, используя вызовы ядра. Данный обработчик прерывания не содержит кода, связанного с аппаратной реализацией прерывания (вход в прерывание, выход из прерывания, сохранение текущего контекста), и ничем не отличается от обычной функции.

К одному прерыванию может быть привязано несколько обработчиков, получающих управление в соответст-

вии с приоритетами потоков, которым они принадлежат.

СОСТАВ ОС QNX/NEUTRINO

В операционную систему QNX/Neutrino наряду с микроядром Neutrino входят следующие компоненты:

- менеджер процессов — `procnto`;
- менеджеры файловых систем — `io-blk`, `fs-nfs`, `fs-cifs`;
- менеджеры символьных устройств;
- графический интерфейс пользователя — `Photon`;
- менеджер очередей;
- менеджер TCP/IP.

Модуль `procnto` (рис. 4) служит для поддержки определенных POSIX-процессов, каждый из которых может содержать множество потоков. Менеджер процессов добавляет к функциям, обеспечиваемым ядром, три новые:

- управление процессами — создание, уничтожение процессов и управление их атрибутами;
- управление памятью — обеспечение различных механизмов защиты памяти, поддержка разделяемых библиотек и работа с общей памятью для межпроцессного взаимодействия;
- управление пространством имен, используемым менеджерами ресурсов.

ФАЙЛОВЫЕ СИСТЕМЫ QNX/NEUTRINO

QNX/Neutrino поддерживает несколько типов файловых систем. Каждому типу файловой системы соответствует свой менеджер, который может быть запущен или выгружен в любой момент времени.

Файловая система QNX наряду с высокой производительностью обеспечивает необходимый уровень надежности.

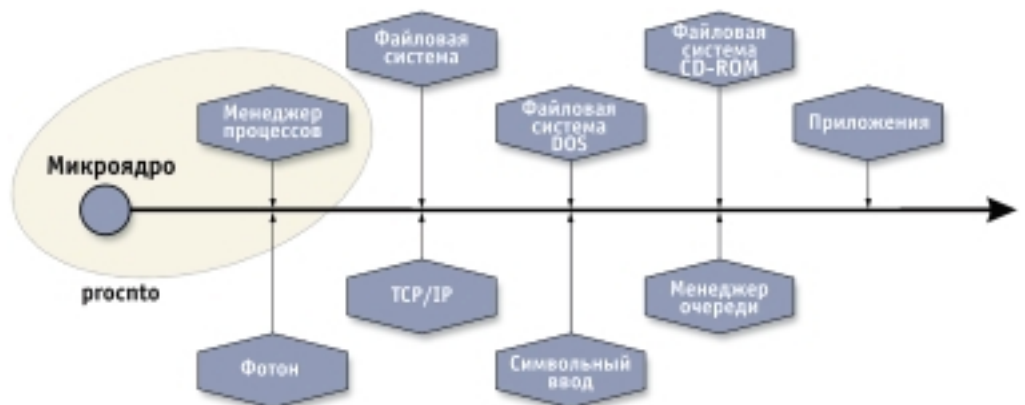
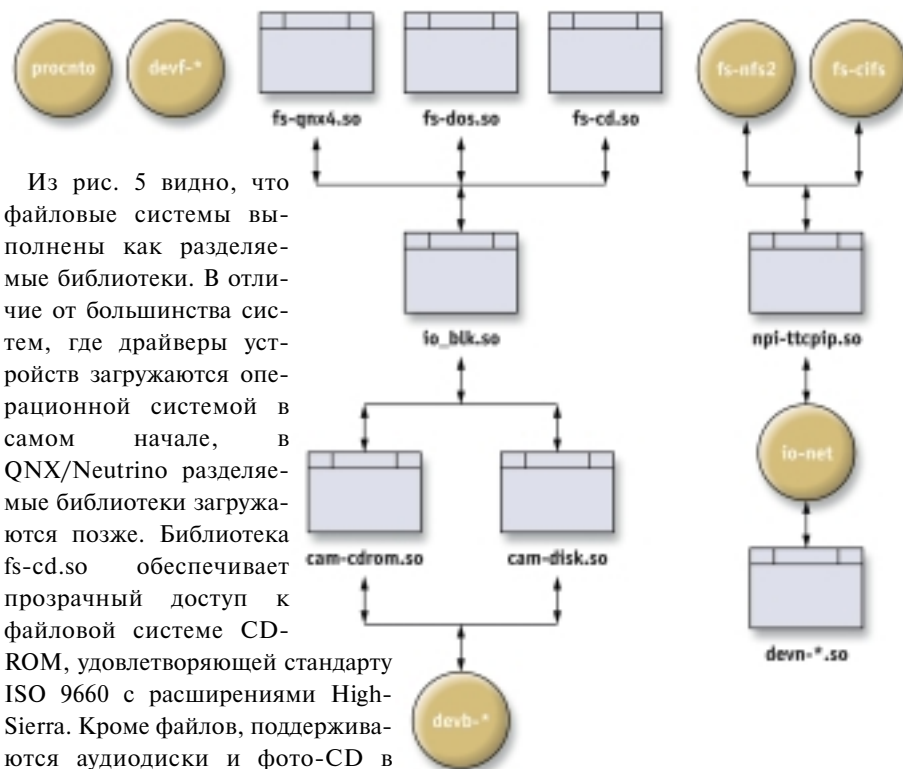


Рис. 4. Модульность архитектуры QNX/Neutrino позволяет легко добавлять любые модули для поддержки устройств



Из рис. 5 видно, что файловые системы выполнены как разделяемые библиотеки. В отличие от большинства систем, где драйверы устройств загружаются операционной системой в самом начале, в QNX/Neutrino разделяемые библиотеки загружаются позже. Библиотека fs-cd.so обеспечивает прозрачный доступ к файловой системе CD-ROM, удовлетворяющей стандарту ISO 9660 с расширениями High-Sierra. Кроме файлов, поддерживаются аудиодиски и фото-CD в формате Kodak.

Модули fs-cifs и fs-nfs позволяют рабочей станции QNX осуществлять доступ к файлам на компьютере под управлением Windows NT/95 и в сетевых файловых системах (Network File System) по протоколу TCP/IP. Запросы к файлам преобразуются в запросы протокола CIFS (Common Internet File System) или NFS и посылаются соответствующему серверу. В качестве протокола транспортного уровня fs-cifs и fs-nfs используют TCP/IP, поэтому для выполнения CIFS и NFS в системе необходимо наличие менеджера TCP/IP, который поддерживается разделяемой библиотекой npi-ttcpip.so.

СЕТЕВОЙ ПРОТОКОЛ QNX/NEUTRINO

QNX/Neutrino имеет собственный надежный и эффективный сетевой протокол транспортного уровня FLEET, который используется для организации обмена в сети и не зависит от аппаратной реализации сети. Эта независимость обеспечивается за счет использования сетевых драйверов. В QNX/Neutrino имеются драйверы для таких сетей, как Ethernet, Arcnet, Token Ring. Кроме этого, есть возможность организации сети через последовательный канал или модем.

TCP/IP в QNX/NEUTRINO

Менеджер TCP/IP выполнен в виде разделяемой библиотеки, что позволяет многократно использовать про-

Рис. 5. Файловая модульная система QNX/Neutrino

граммный код. Менеджер TCP/IP может обслуживать несколько клиентов одновременно, в порядке их приоритетов, что приводит к более рациональной загрузке процессора (рис. 6).

ГРАФИЧЕСКАЯ ПОДСИСТЕМА PHOTON MICRO GUI

Neutrino имеет свою графическую подсистему Photon. Существует немало ОС, пригодных для создания систем реального времени, тем не менее, в большинстве из них нет сколько-нибудь удобного и отвечающего требованиям реального времени графического интерфейса пользователя (GUI). Те немногие системы, которые способны поддержать полноценный GUI, не позволяют использовать это на практике, поскольку реализация традиционных GUI, например типа X11, связана с очень высокими затратами ресурсов, особенно памяти.

Micro GUI Photon является графической подсистемой, по внешнему виду и структуре похожей на X11/Motif, но куда менее требовательной к ресурсам. Графический интерфейс Photon построен по принципу модульности с применением механизма обмена сообщениями, что и сделало его пригодным к использованию во встраиваемых системах реального времени.

Низкие требования к оперативной памяти являются отличительной чер-

той Photon по сравнению с другими графическими средами.

Photon имеет очень удобную, гибкую и расширяемую архитектуру, которая позволяет разработчикам GUI-приложений существенно сокращать сроки создания программного обеспечения.

СРЕДА РАЗРАБОТКИ

При всех перечисленных достоинствах операционная система не может быть сколько-нибудь интересной без удобного и полного средства разработки. Для QNX/Neutrino таковым является Code Warrior IDE, созданный фирмой MetroWerks, мировым лидером в системах разработки встраиваемых систем. Эта кросс-платформенная интегрированная система разработки позволяет разрабатывать приложения не только из QNX/Neutrino, но и под управлением Windows NT, а также легко настраивается для работы с любым C/C++ компилятором и включает в

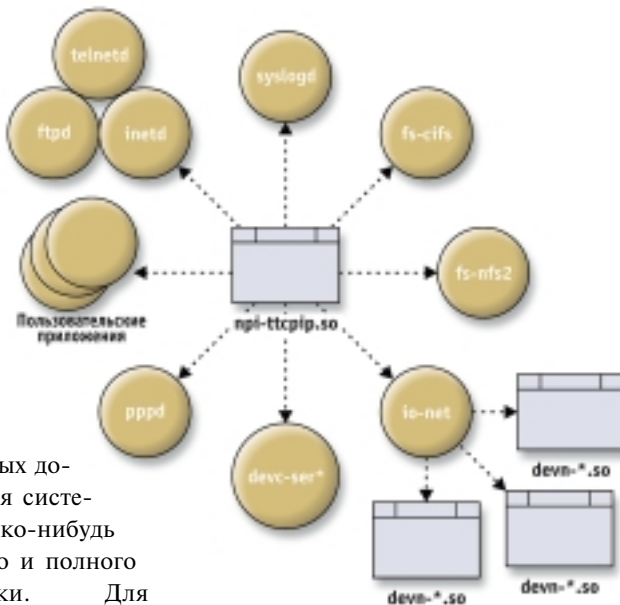


Рис. 6. TCP/IP стек в QNX/Neutrino

себя многооконный интерфейс, отладчик и систему контекстных подсказок.

На данный момент в среду QNX 4.x перенесен GNU C/C++ компилятор с библиотеками, позволяющий генерировать приложения для QNX/Neutrino. К концу года ожидается GNU-версия компилятора для QNX/Neu-

trino, работающая непосредственно под управлением Neutrino (self-hosted).

ЗАКЛЮЧЕНИЕ

Уже сейчас понятно, что у QNX/Neutrino огромные возможности для применения, благодаря соответствию интерфейса прикладного программирования стандартам POSIX и настоящему микроядру. Эти свойства одинаково привлекательны как для использования во встраиваемых системах (интеллектуальные контроллеры, маршрутизаторы и т.д.), так и для реализации функций Internet-сервера, файл-сервера и сервера приложений. Все это, несомненно, делает ОС PB QNX/Neutrino перспективной системой для новых разработок во встраиваемых системах и в системах управления производством. ●

С.С. Прожогин – сотрудник SWD Real Time Systems
 196135, Россия,
 Санкт-Петербург,
 пр. Ю. Гагарина, 23.
 Тел.: (812) 443-0260
 Факс: (812) 443-0497
 E-mail: sergey.prozhogin@swd.ru