

Технологии **QNX**® для создания
высоконадежных систем автоматизации



Сергей Зыль, СВД Встраиваемые Системы
Состав и функциональные возможности
комплекта разработчика QNX SDP



Немного истории



QNX RTP, NIP 6.1 – *2001*



QNX Momentics NC, SE, PE 6.2.x – *2002-2003*

QNX Momentics NC, SE, PE 6.3.x – *2004-2007*



QNX SDP 6.4.x – *2008-2009*

QNX SDP 6.5.0 – *2010*



Формально QNX SDP состоит из двух частей:

- ❑ QNX Momentics - инструментальный комплект
- ❑ QNX Neutrino – компоненты среды исполнения



Хост-платформы QNX Momentics:

MS Windows



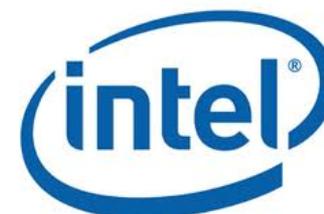
Linux



QNX Neutrino



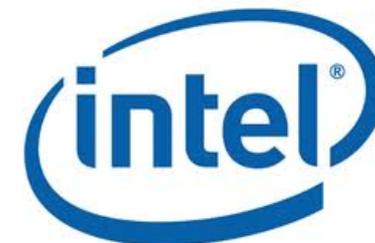
Все – x86





QNX Neutrino 6.5.0 поддерживает следующие аппаратные платформы:

- ARM-LE
- ARMv7-LE
- MIPS-BE
- MIPS-LE
- PPC-BE
- PPC-LE
- SH4-LE
- X86-LE



QNX Neutrino 6.5.0 соответствует стандартам:

- ✓ IEEE POSIX (PSE52)
- ✓ ISO/IEC 15408 (EAL 4+)
- ✓ IEC 61508 (SIL 3)
- ✓ OpenGL® ES
- ✓ IEC 62304
- ✓ ISO 26262





Функционально QNX SDP состоит из следующих частей:

- ❑ Набор компонентов-«кирпичиков» среды исполнения QNX Neutrino для каждой из поддерживаемых целевых аппаратных платформ
- ❑ Технологические комплекты TDK
- ❑ Инструменты для формирования и анализа целевых систем QNX Neutrino
- ❑ Инструменты для разработки, отладки и тестирования собственных компонентов

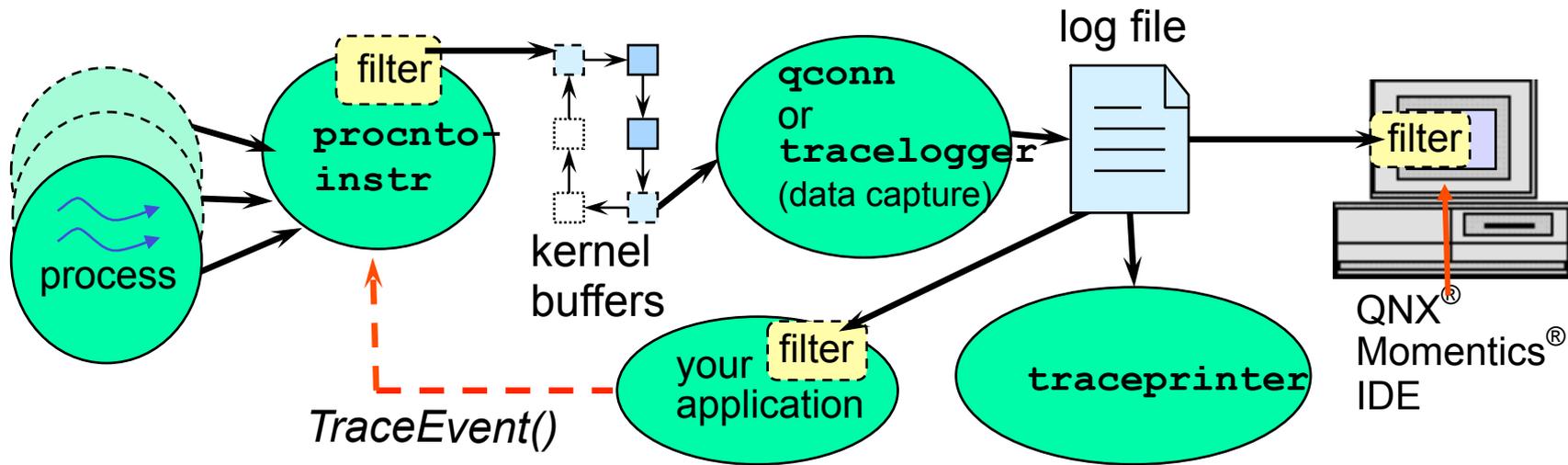


QNX Neutrino 6.5.0 включает как традиционные подсистемы:

- Neutrino Core Technology
- Instrumented Kernel Technology**
- Embedded/Flash Technology
- Mass Storage Filesystem Technology
- Networking Technology
- Photon GUI Technology
- драйвера, утилиты POSIX и т.д.



Системное профилирование



- ❑ Instrumented Kernel Technology



QNX Neutrino 6.5.0 включает как традиционные подсистемы:

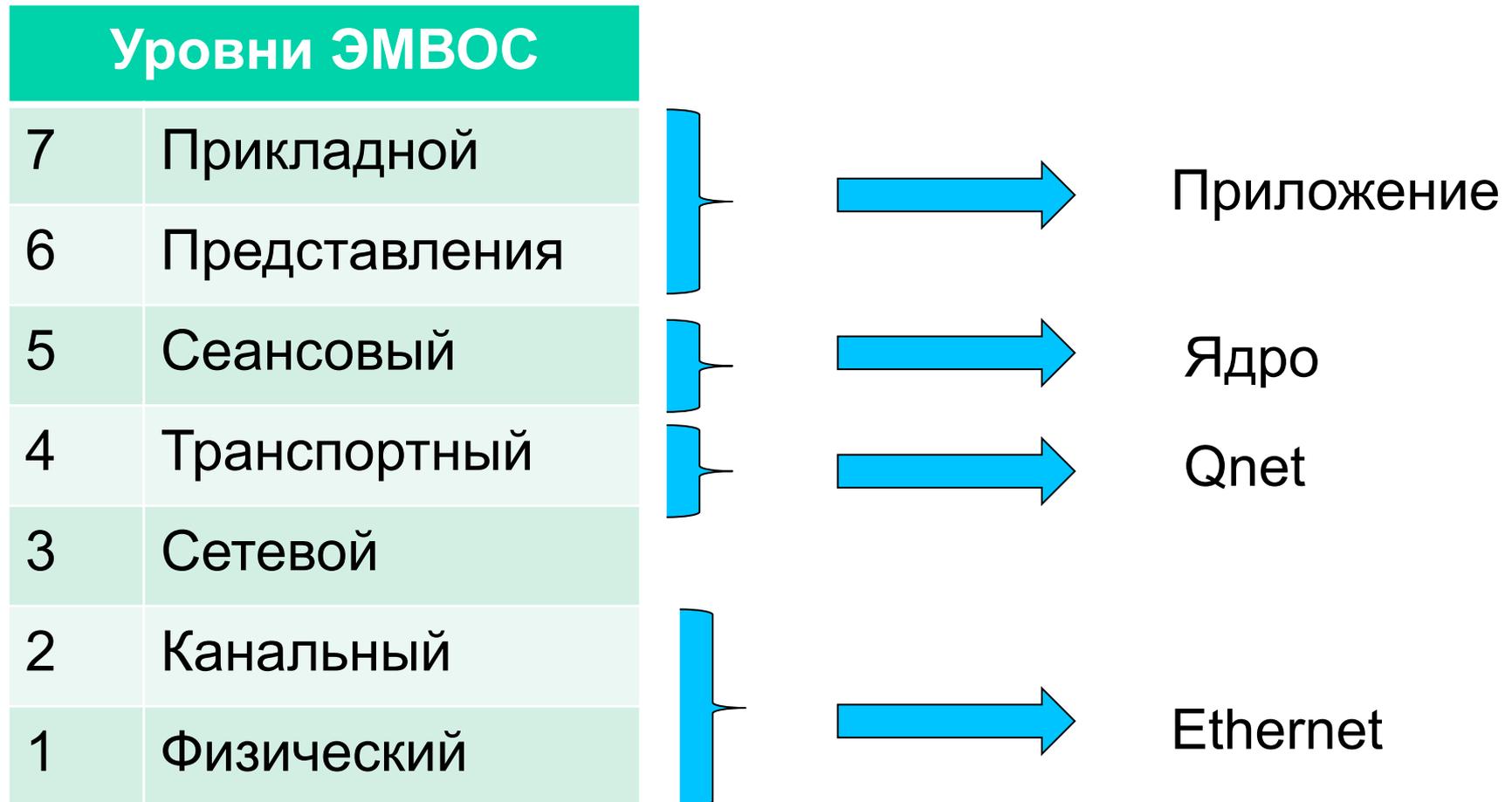
- Neutrino Core Technology
- Instrumented Kernel Technology
- Embedded/Flash Technology
- Mass Storage Filesystem Technology
- Networking Technology**
- Photon GUI Technology
- драйвера, утилиты POSIX и т.д.



Прозрачно распределенная сеть Qnet

СВД Встраиваемые Системы

Транспортный уровень протокола Qnet (4 уровень модели OSI) располагается ниже логического уровня микроядра QNX!!!





Функционально QNX SDP состоит из следующих частей:

- ❑ Компоненты среды исполнения QNX Neutrino
- ❑ Технологические комплекты TDK
- ❑ Инструменты для формирования из «кирпичиков» целевых систем QNX Neutrino
- ❑ Инструменты для разработки, отладки и тестирования собственных компонентов-«кирпичиков»

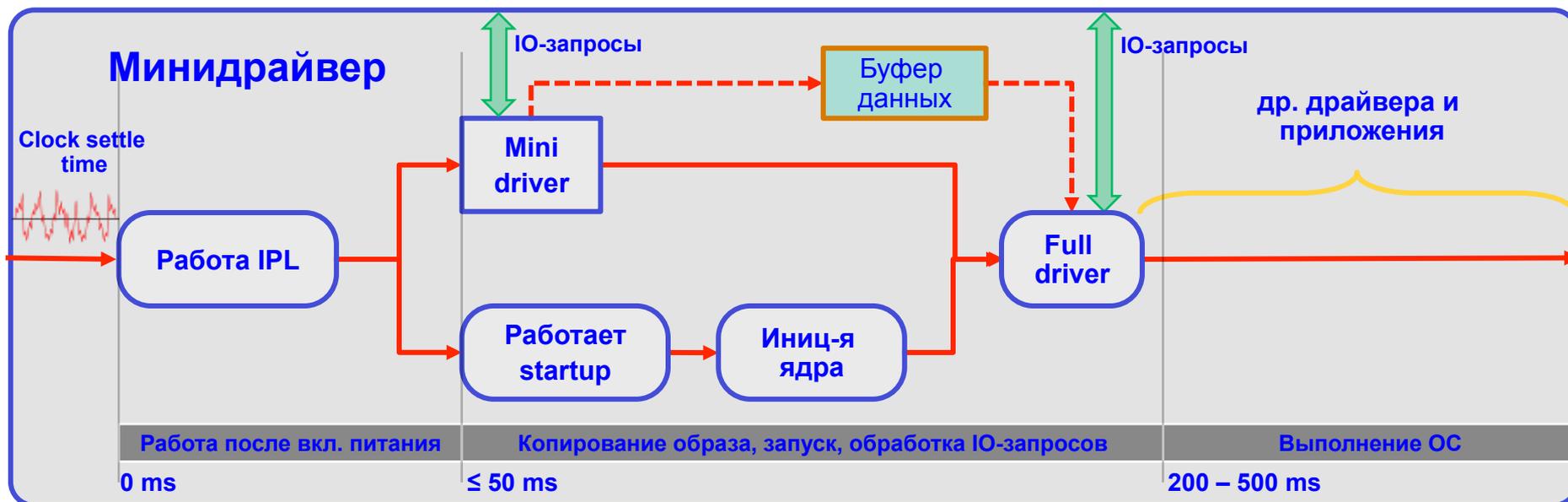


Так и механизмы, ранее входившие в состав ряда TDK:

- Multi-Core Technology
- Instant Device Activation Technology**
- Adaptive Partitioning Technology
- Extended Networking Technology
- High Availability Toolkit
- Core Graphics Technology



Быстрая активация устройств



- ❑ Instant Device Activation Technology

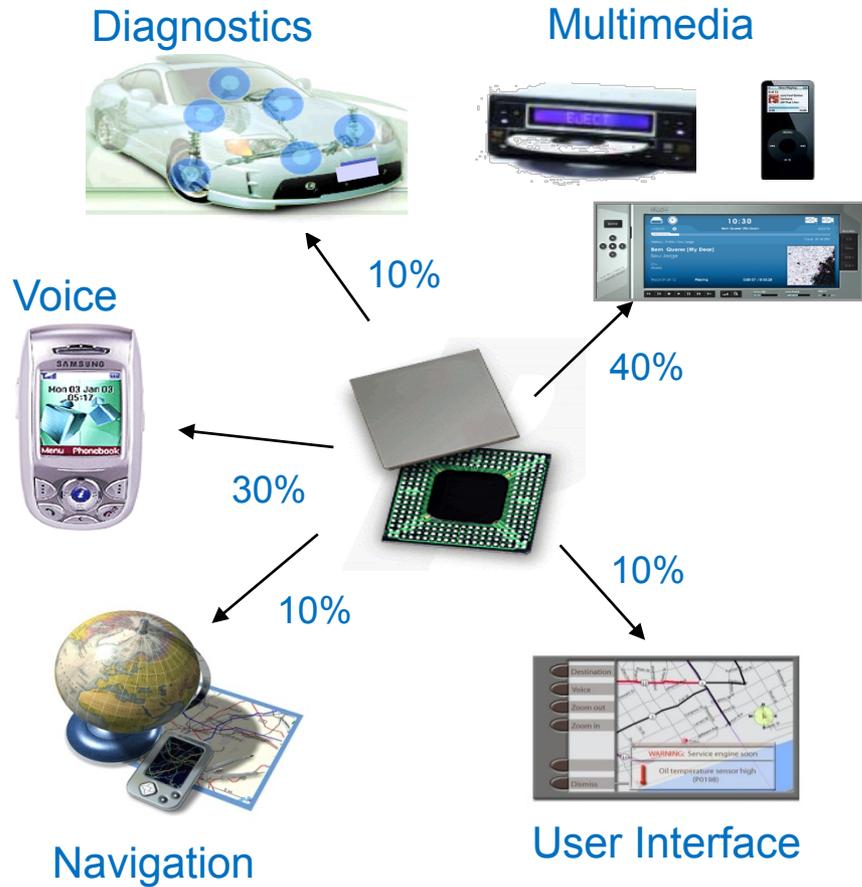


Так и механизмы, ранее входившие в состав ряда TDK:

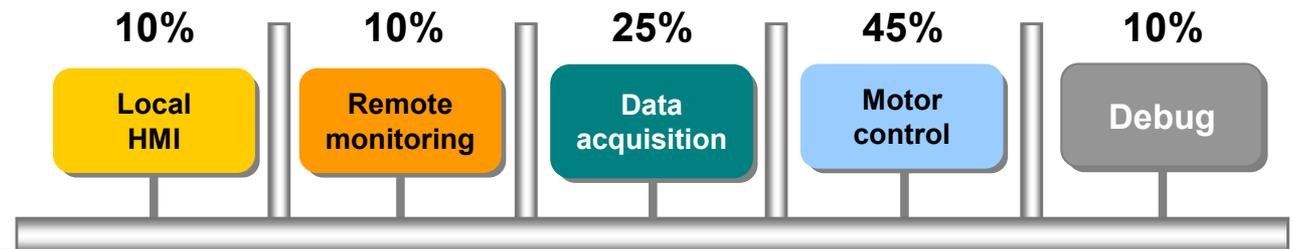
- Multi-Core Technology
- Instant Device Activation Technology
- Adaptive Partitioning Technology**
- Extended Networking Technology
- High Availability Toolkit
- Core Graphics Technology



Адаптивное квотирование ЦПУ



□ Adaptive Partitioning Technology





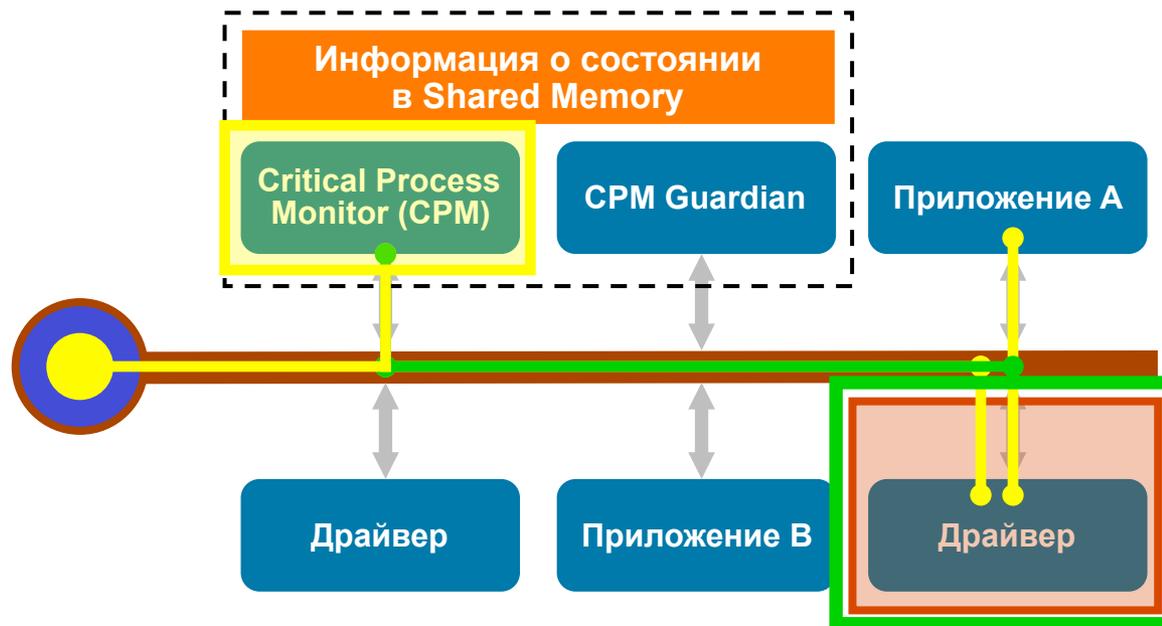
Так и механизмы, ранее входившие в состав ряда TDK:

- Multi-Core Technology
- Instant Device Activation Technology
- Adaptive Partitioning Technology
- Extended Networking Technology
- High Availability Toolkit**
- Core Graphics Technology



Монитор ключевых процессов

1. Сбой драйвера (некорректный доступ к памяти)
2. Ядро извещает CPM о сбое процесса



3. Сохраняется диагностическая информация (core-файл и kevg-файл)
4. Драйвер завершается возвращает системе ресурсы; канал IPC разрушен
5. CPM перезапускает драйвер
6. Клиентская библиотека CPM восстанавливает каналы IPC
7. Драйвер восстанавливает сервис по состоянию на последней точке синхронизации из CPM



Так и механизмы, ранее входившие в состав ряда TDK:

- Multi-Core Technology
- Instant Device Activation Technology
- Adaptive Partitioning Technology
- Extended Networking Technology
- High Availability Toolkit
- Core Graphics Technology



Функционально QNX SDP состоит из следующих частей:

- ❑ Компоненты среды исполнения QNX Neutrino
- ❑ Технологические комплекты TDK
- ❑ Инструменты для формирования из «кирпичиков» адаптированных под конкретное назначение целевых систем QNX Neutrino и их анализа - для каждой из 3-х поддерживаемых хост-платформ
- ❑ Инструменты для разработки, отладки и тестирования собственных компонентов



QNX SDP – конструктор

СВД Встраиваемые Системы

Построение целевых систем (QNX System Builder)

The screenshot displays the QNX System Builder IDE interface. The main workspace is divided into three panes: a left-hand 'System Builder' tree showing project components like 'Images', 'Overrides', and 'Reductions'; a central 'Images' pane showing a hierarchical tree of binaries and shared libraries for the 'qnxbase' image; and a right-hand 'Filesystem' pane showing the file structure for the 'proc' image. A 'Properties' window on the right lists various configuration options such as 'Create Image?', 'Image Name', 'CPU Type', and 'Boot Script'. At the bottom, a 'Terminal' window shows the output of a 'C-Build' command, including file paths and sizes for various boot components.

Property	Value
General	
Create Image?	Yes
Remove File Time Stamps?	No
Image Name	qnxbase
CPU Type	x86
Page Align Image?	No
Image Mount Point	
Default Target Location	/proc/boot
Compressed	UCL
Boot Script	qnxbase.bsh
Directories	
Default Permissions	777
Default User ID	0
Default Group ID	0
System	
Auto Link Shared Libs?	Yes
Create startup sym file?	No
Create proc sym file?	No
Procnto	procnto-instr
Procnto Arguments	
Procnto \$PATH	/proc/boot/bin/...
Procnto \$LD_LIBRARY_PATH	/proc/boot/lib/...
Use APS?	No
Startup	startup-bios
Startup Arguments	-s64k
Boot File	bios
Image Address	Default
RAM Address	Default
Combine	
IPL file	
Pad IPL to	
Align File System to	
Offset	0x0
ROM size	
Combined Image format	
Images to combine with	
Mount unbootable IFSs?	Yes

```
C-Build [qas]
700000 5386 8049428 --- proc/boot/fesh=C:\QNX650\target\qnx6\x86\bin\fesh (4152905123)
706000 152cd 804a0f8 --- proc/boot/devc-con=C:\QNX650\target\qnx6\x86\sbin\devc-con (1413428113)
71c000 1a527 804a338 --- proc/boot/devc-con-hid=C:\QNX650\target\qnx6\x86\sbin\devc-con-hid (2889978970)
736527 1011 ---- proc/boot/umass-enum.cfg=C:\QNX650\target\qnx6\etc\umass-enum.cfg (4174073322)
737538 4 ---- --- Image-trailer
```



Анализ целевых систем (QNX System Information)

Object	Object Owners	State	Blocked Threads
procnto-smp-ksz...	procnto-smp-ksz (1) Thread 9	Running	tinit (36882) Thread 1
	procnto-smp-ksz (1) Thread 2	Receive	qconn (1433631) Thread 1
	procnto-smp-ksz (1) Thread 3	Receive	
	procnto-smp-ksz (1) Thread 4	Receive	
	procnto-smp-ksz (1) Thread 5	Receive	
	procnto-smp-ksz (1) Thread 7	Receive	
	procnto-smp-ksz (1) Thread 11	Receive	
	procnto-smp-ksz (1) Thread 13	Receive	
	procnto-smp-ksz (1) Thread 14	Receive	
	procnto-smp-ksz (1) Thread 15	Receive	
	procnto-smp-ksz (1) Thread 16	Receive	
devrntv (4098)	devrntv (4098) Thread 1	Receive	sh (2011166) Thread 1



QNX SDP – конструктор

СВД Встраиваемые Системы

Анализ целевых систем (QNX System Profiling)

The screenshot displays the QNX System Profiler interface for a target system. The main window is titled "QNX System Profiler - 192.168.5.93/192.168.5.93-trace-120406-164625.kev - QNX Momentics IDE".

Summary

System Activity

Name	%	Time
Idle	89,6%	2sec 657ms
User	5,7%	168ms428us
Kernel	1,9%	56ms917us
Interrupts	2,7%	81ms317us

Total Log File Time: 2sec 964ms
Total Number Of Events: 48410
Dropped Buffers: 0 of 93
Number of CPUs: 1
User Time: 168ms428us
System Time: 2sec 795ms
Busy Time: 306ms663us
Idle Time: 2sec 657ms

Process & Thread Activity

Process Name	Running Time	Running Time (Sum)	Ready Time	Ready Time (Sum)	Blocked Time	Blocked Time (S...)	Num Kern...	Num Mes...
pronto-smp-kz	2sec 839ms	2sec 839ms	125ms53us	245ms530us	0ns	32sec 490ms	3851	7
qconn	47ms407us	47ms413us	21ms459us	26ms866us	2sec 895ms	17sec 710ms	1512	1050
io-display	27ms724us	27ms724us	15ms331us	15ms331us	2sec 921ms	20sec 706ms	984	0
io-pkt-kz	18ms34us	18ms40us	2ms544us	3ms872us	2sec 943ms	20sec 728ms	499	29
shelf	6ms504us	6ms504us	3ms96us	4ms502us	2sec 954ms	5sec 917ms	143	86
Photon	6ms281us	6ms281us	3ms28us	3ms28us	2sec 954ms	2sec 954ms	182	0
pterm	5ms2us	5ms2us	2ms362us	2ms362us	2sec 956ms	2sec 956ms	180	96

Trace Event Log

Event	Time	Owner	Type	Data
0001			Time	msb 0xbb69 lsb 0x7b032683
0002	225us	pronto-smp-kz	Create Process Name	ppid 0 pid 1 name proc/boot/pronto-s...
0003	273us	pronto-smp-kz CPU 1 idle	Create Thread	pid 1 tid 1
0004	302us	pronto-smp-kz CPU 1 idle	Ready	pid 1 tid 1
0005	310us	pronto-smp-kz Thread 2	Create Thread	pid 1 tid 2
0006	310us	pronto-smp-kz Thread 2	Receive	pid 1 tid 2
0007	311us	pronto-smp-kz Thread 3	Create Thread	pid 1 tid 3
0008	312us	pronto-smp-kz Thread 3	Receive	pid 1 tid 3
0009	313us	pronto-smp-kz Thread 4	Create Thread	pid 1 tid 4
0010	313us	pronto-smp-kz Thread 4	Receive	pid 1 tid 4
0011	314us	pronto-smp-kz Thread 5	Create Thread	pid 1 tid 5



Функционально QNX SDP состоит из следующих частей:

- ❑ Компоненты среды исполнения QNX Neutrino
- ❑ Технологические комплекты TDK
- ❑ Инструменты для формирования и анализа целевых систем QNX Neutrino
- ❑ Инструменты для разработки, отладки и тестирования собственных компонентов-«кирпичиков» для работы на поддерживаемой аппаратуре – на каждой из 3-х хост-платформ



Анализ целевых систем (QNX Memory Analysis)

The screenshot displays the QNX Momentics IDE interface during a memory analysis session. The main window shows the source code for 'qwert.cc', which includes headers for `<cstdlib>` and `<iostream>`, and a `main` function that prints a welcome message and returns `EXIT_SUCCESS`.

The Memory Events window shows 237 events. The Memory Pro window displays 17 memory leak events, all of which are 'LEAK' severity. The events are summarized in the table below:

Severity	Description
LEAK	memory leak of size 4
LEAK	memory leak of size 4
LEAK	memory leak of size 4
LEAK	memory leak of size 4
LEAK	memory leak of size 4
LEAK	memory leak of size 4
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 8
LEAK	memory leak of size 20
LEAK	memory leak of size 20
LEAK	memory leak of size 20
LEAK	memory leak of size 20



QNX SDP – среда разработки

СВД Встраиваемые Системы

Профилирование приложения (QNX Application Profiler)

The screenshot displays the QNX Application Profiler interface. The top window shows the Profiler Sessions and Debug panels. The main window is divided into three panes:

- Execution Time:** A tree view showing the execution time of various threads and functions. The root node is "qwerty #34 - Threads Tree". The "main (self)" node is highlighted, showing a total execution time of 4,114 ms. The "main->std::basic_ostream<char, <...>>" node is also highlighted, showing a total execution time of 4,114 ms.
- Threads Tree:** A table showing the execution time of various threads and functions. The table has columns for Name, Deep Time, Shallow Time, Count, Location, and Binary.
- Source Code:** A window showing the source code of the application, specifically the main function in "qwerty.cc". The code is as follows:

```
#include <stdlib>
#include <iostream>

int main(int argc, char *argv[]) {
    std::cout << "Welcome to the QNX Momentics IDE" << s
    return EXIT_SUCCESS;
}
```



Тестирование приложения (QNX Code Coverage)

The screenshot displays the QNX Momentics IDE with the following components:

- Navigator:** Shows a tree view of the project structure. The 'ios' folder is highlighted, showing a coverage of 62.5%.
- Code Editor:** Displays the source code for 'qwerty.cc', including functions like 'init(_Strbuf)', 'virtual ~basic_ios()', 'void clear()', 'void setstate()', and 'void setstate(io_state_State)'. The 'clear' function is currently selected.
- Debug Console:** Shows the execution status of the application, including messages like '<terminated> qwerty [C/C++ QNX QConn (IP)]' and '<stopped> qwerty [GCC Code Coverage]'. The exit value is reported as 1.
- Code Coverage Report:** A window titled 'Code Coverage Report' generated on 16.04.12 14:21. It provides session details and a summary table.

Code Coverage Report Summary:

Project/Folder/File/Function	Total Code Coverage	Source Line Coverage			Total Lines
		Lines Not Covered	Lines Partially Covered	Lines Fully Covered	
C:\ide-4.7-workspace\qwerty	100.00%	0	0	4	4



QNX SDP версии 6.5.0

- ❑ Компоненты QNX Neutrino для всех поддерживаемых целевых аппаратных платформ – ARM, MIPS, SH4, PPC, x86
- ❑ Технологические комплекты TDK
- ❑ Инструменты для формирования целевых систем QNX Neutrino и их анализа для хост-платформ Windows, Linux, QNX
- ❑ Инструменты для разработки, отладки и тестирования собственных компонентов для всех хост-платформ



Спасибо за внимание!

Сергей Зыль

ООО «СВД Встраиваемые Системы»

www.kpda.ru

forum.kpda.ru