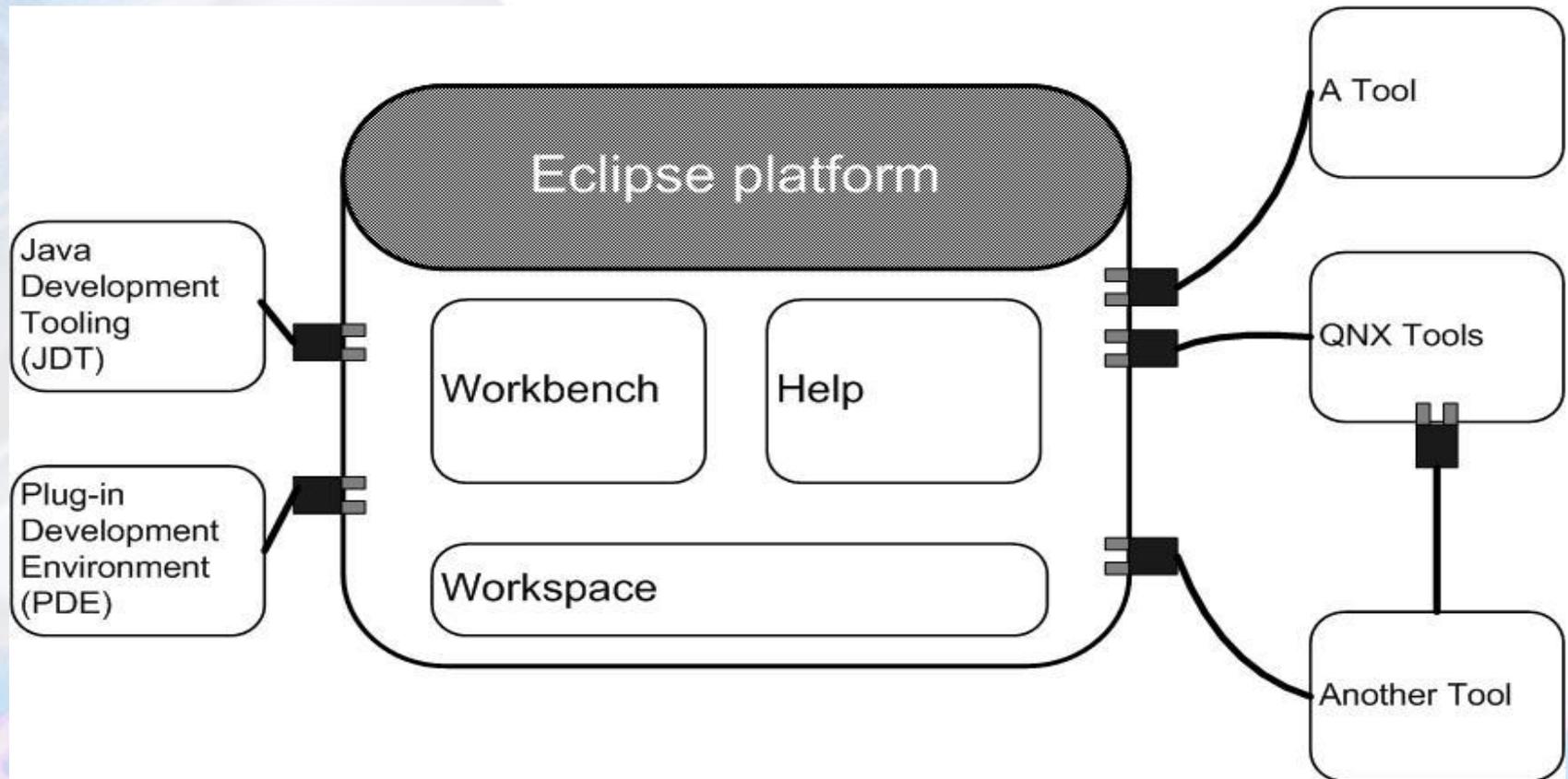


IDE QNX – практическая работа

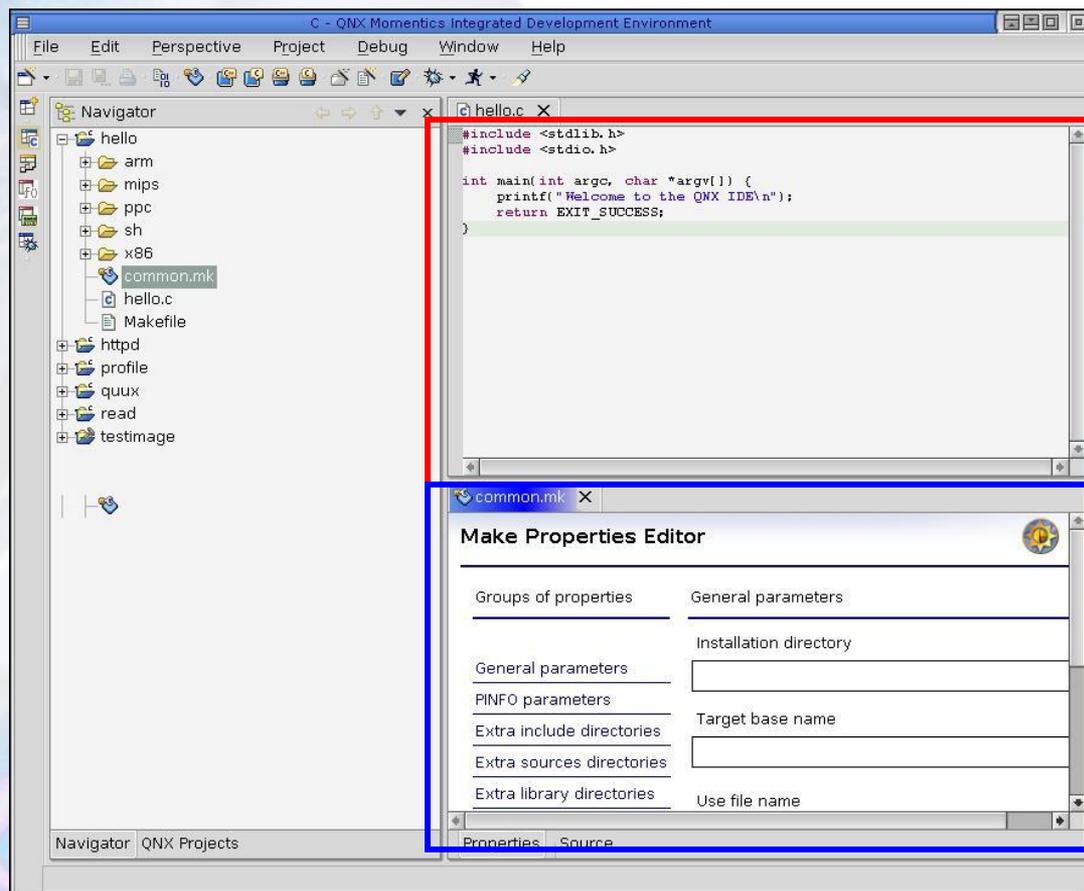
Трофимов Александр
SWD Software

Архитектура Eclipse



- Организация ресурсов (проекты, папки, файлы)
- Редактирование ресурсов
- Организация работы над проектами в команде
- Компиляция, запуск и отладка программ
- Наблюдение и анализ работы системы
- Построение загрузочных образов для встраиваемых систем

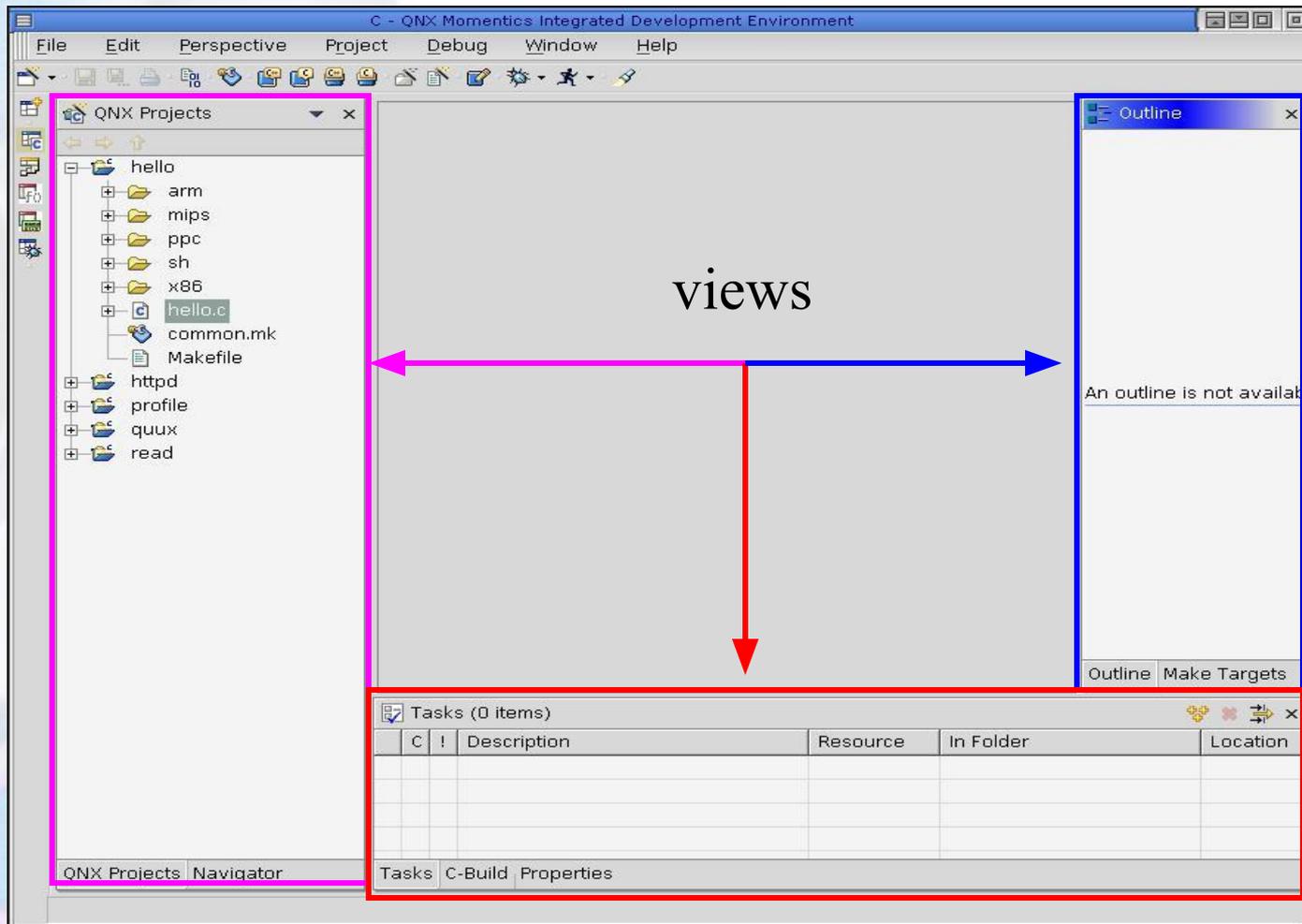
- *perspectives* – зависящий от конкретной задачи набор *view*'ов и *editor*'ов.
- *views* – специальный инструмент для просмотра и работы над ресурсами, обеспечивающий организацию информации и навигацию по ней, а также некоторую поддержку для *editor*'ов.
- *editors* - визуальные компоненты для редактирования и просмотра ресурсов (например, файл с исходным C++ кодом).



C editor

Make Properties
Editor

Пример view



Резидентная среда разработки:

QNX Momentics IDE

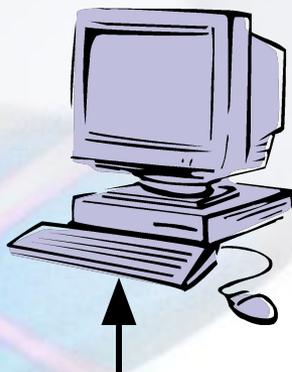


PC под управлением
QNX Neutrino

*Процесс, который вы
запускаете, отлаживаете,
профилируете*

Кросс разработка:

Инструментальная система под
управлением Windows/Solaris/Neutrino



QNX Momentics IDE

Целевая система под управлением
QNX Neutrino



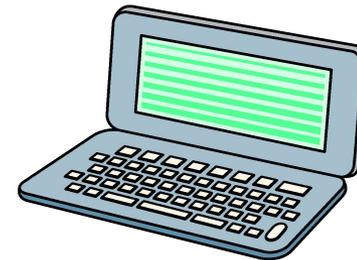
*Процесс, который вы
запускаете, отлаживаете,
профилируете*

транспорт

Инструментальная система с IDE



Целевая система под управлением QNX Neutrino



TCP/IP

Ethernet

- io-net
- devc-pty
- qconn
- pdebug
- *ваша_программа*

Что может быть целевой системой



- ARM/Xscale
- MIPS
- PowerPC
- SH-4
- x86

Настройка целевой системы в IDE

New QNX Target System

Target Name:

Use default location

Location:

QNX Connector Selection

Use local QNX Connector

Hostname or IP: Port:

Target Configuration

QNX Connector is on the target machine:

Hostname:

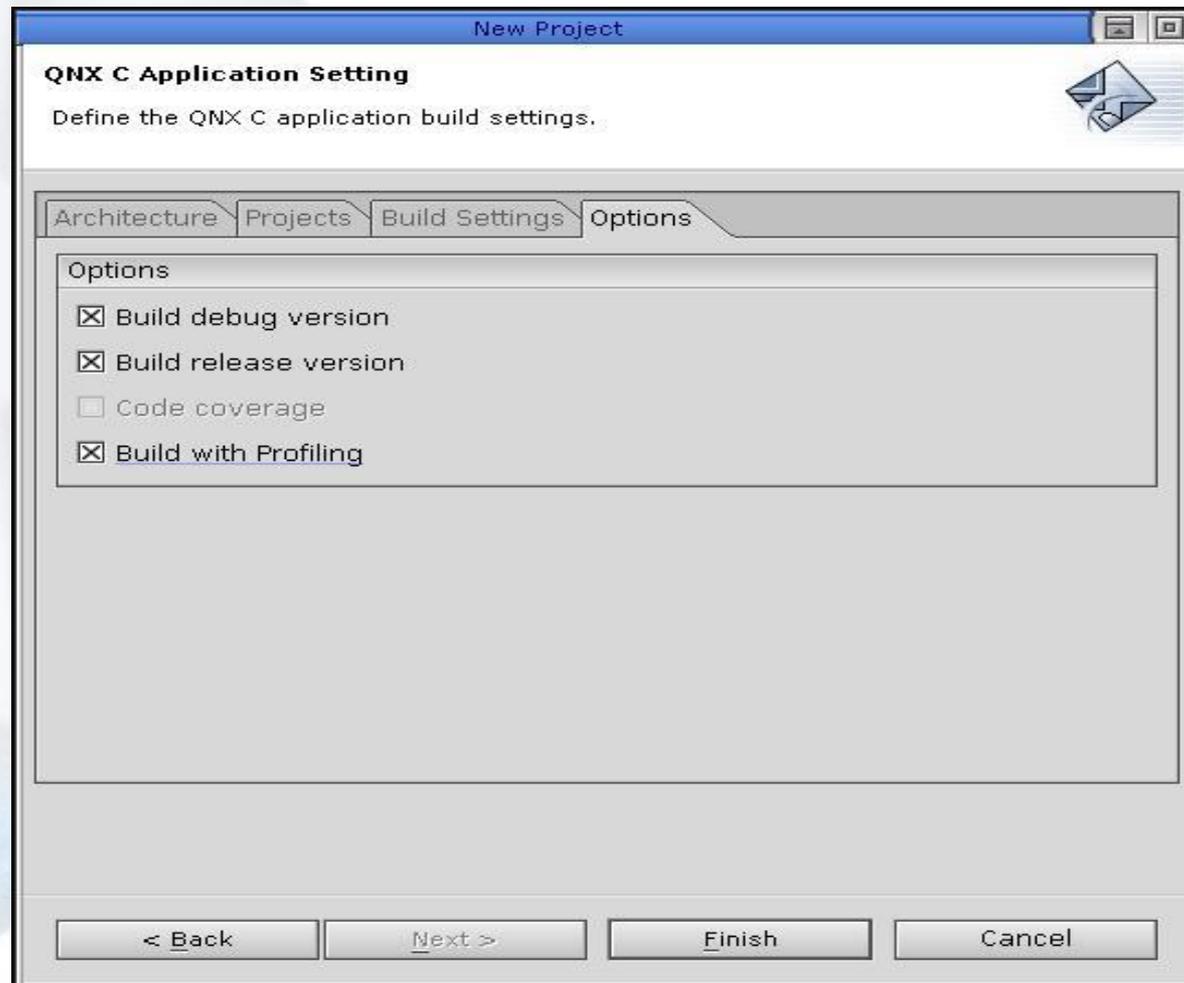
< Back Next > Finish Cancel

Что такое проект?

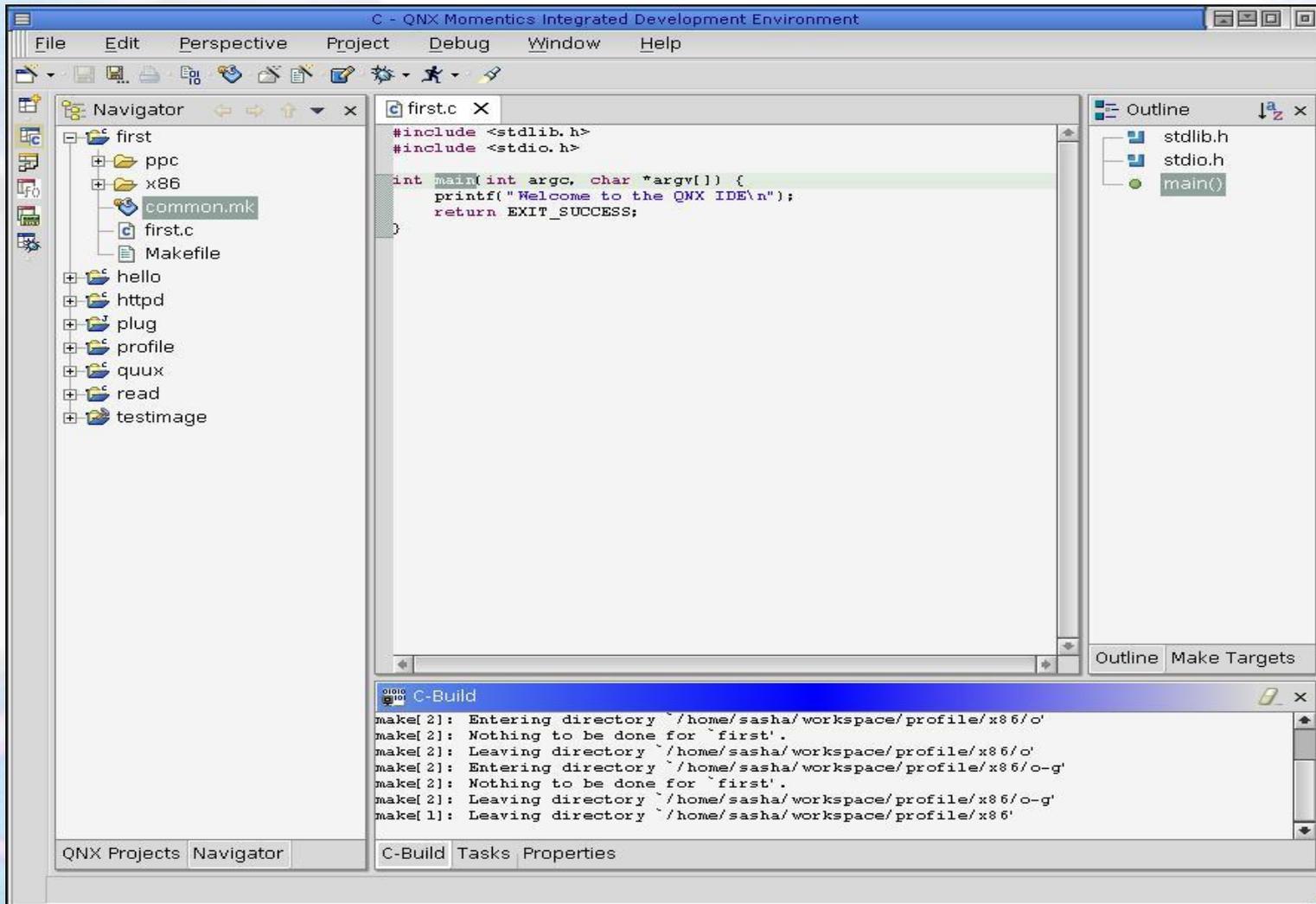
В IDE проект – это набор связанных ресурсов (например, каталогов или файлов) для управления вашими программами.

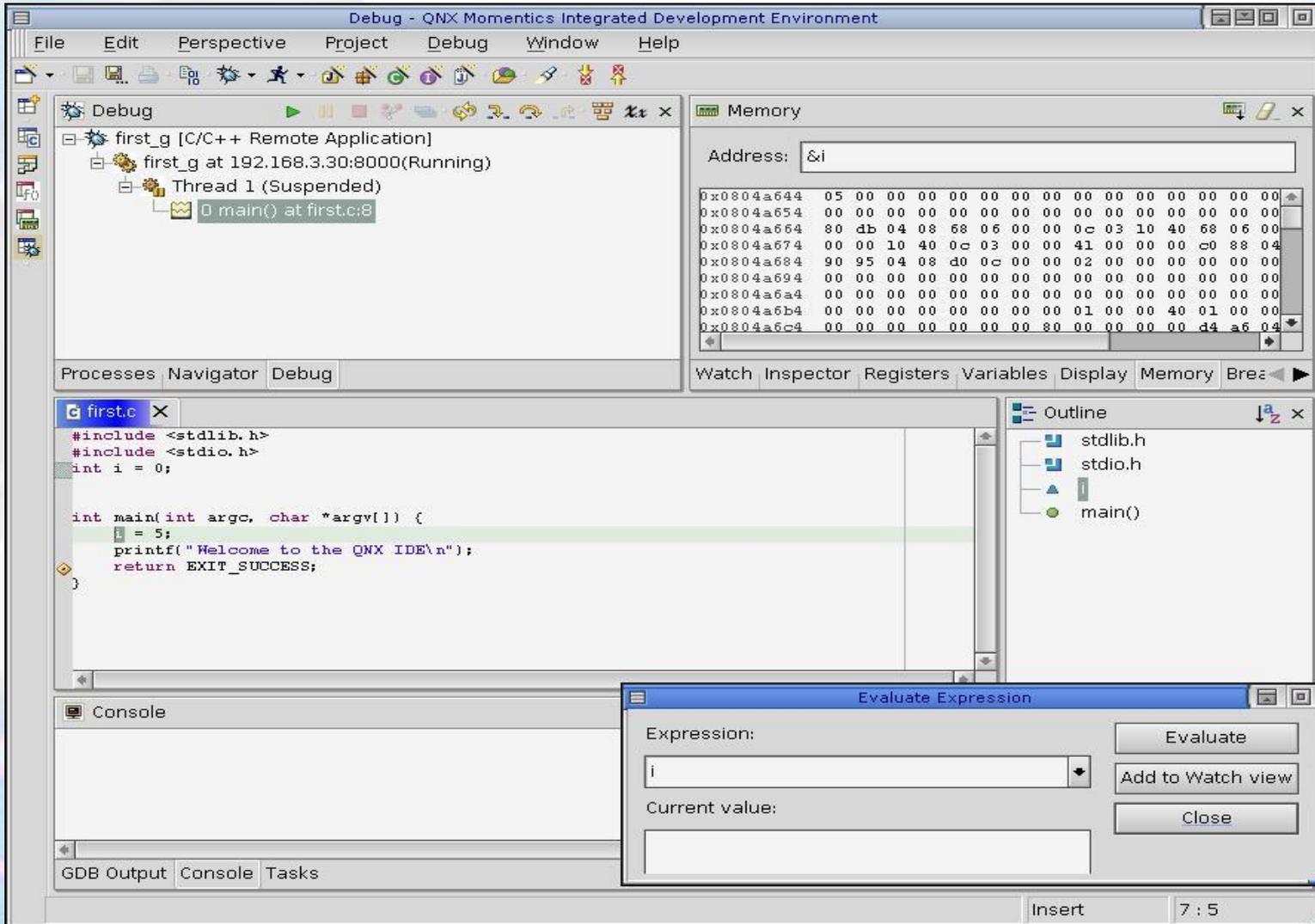
Большинство того, что делается в IDE – проектно-ориентированно – построение проектов, контроль версий, совместная работа над проектами.

Создание QNX C/C++ проекта



C Perspective





Debug - QNX Momentics Integrated Development Environment

File Edit Perspective Project Debug Window Help

Debug

- first_g [C/C++ Remote Application]
 - first_g at 192.168.3.30:8000(Running)
 - Thread 1 (Suspended)
 - 0 main() at first.c:8

Memory

Address: &i

0x0804a644	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0804a654	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0804a664	80	db	04	08	68	06	00	00	0c	03	10	40	68	06	00	00	00	00	00
0x0804a674	00	00	10	40	0c	03	00	00	41	00	00	00	00	c0	88	04	00	00	00
0x0804a684	90	95	04	08	d0	0c	00	00	02	00	00	00	00	00	00	00	00	00	00
0x0804a694	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0804a6a4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0804a6b4	00	00	00	00	00	00	00	00	00	01	00	00	40	01	00	00	00	00	00
0x0804a6c4	00	00	00	00	00	00	00	80	00	00	00	00	d4	a6	04	00	00	00	00

Processes Navigator Debug

first.c

```
#include <stdlib.h>
#include <stdio.h>
int i = 0;

int main(int argc, char *argv[]) {
    i = 5;
    printf("Welcome to the QNX IDE\n");
    return EXIT_SUCCESS;
}
```

Outline

- stdlib.h
- stdio.h
- main()

Console

Evaluate Expression

Expression: i

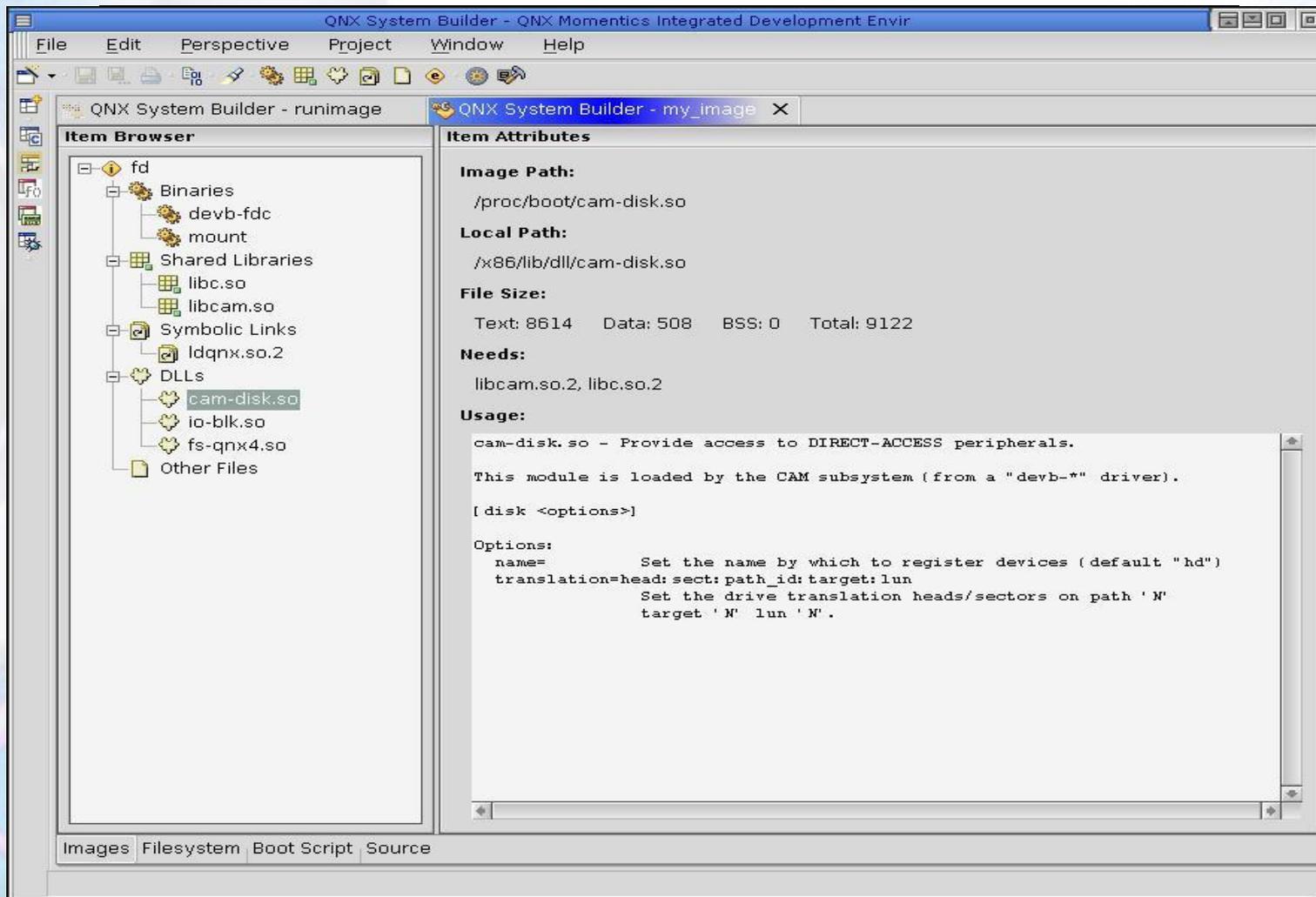
Current value:

Evaluate Add to Watch view Close

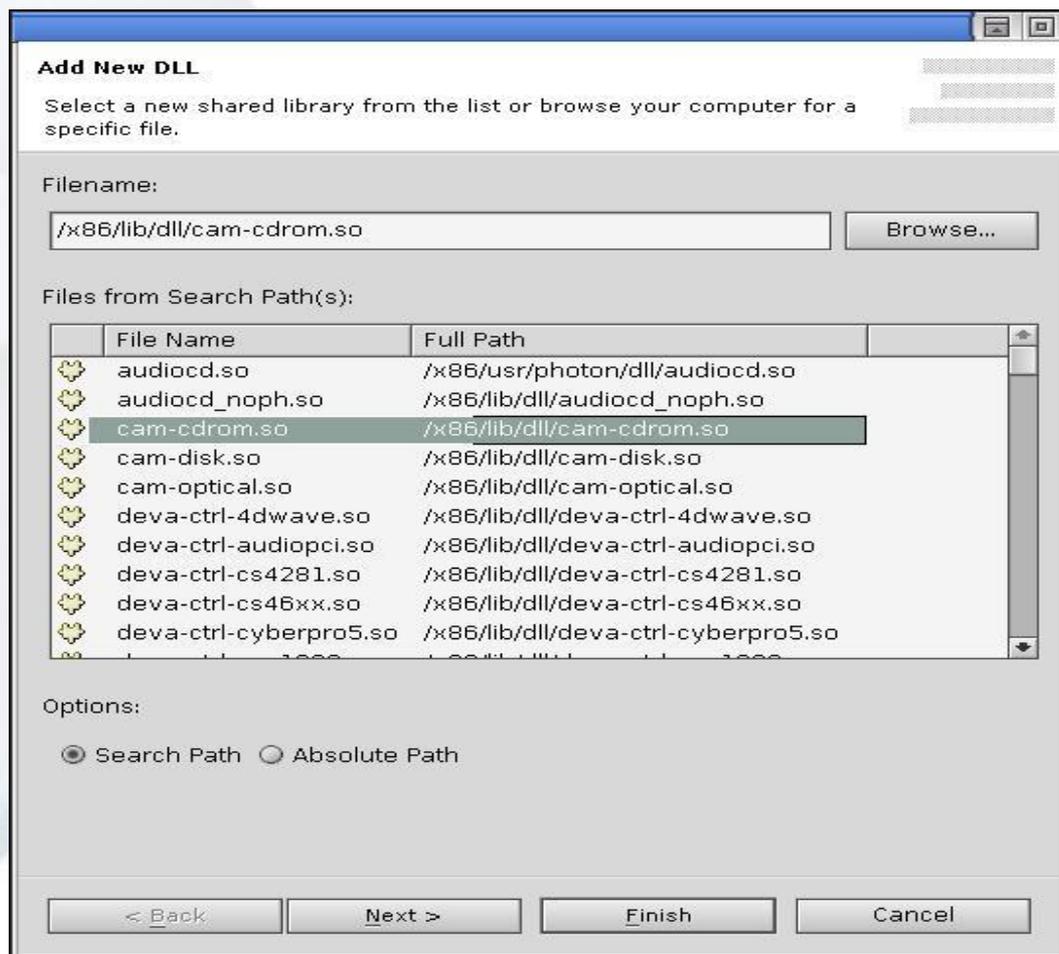
GDB Output Console Tasks

Insert 7 : 5

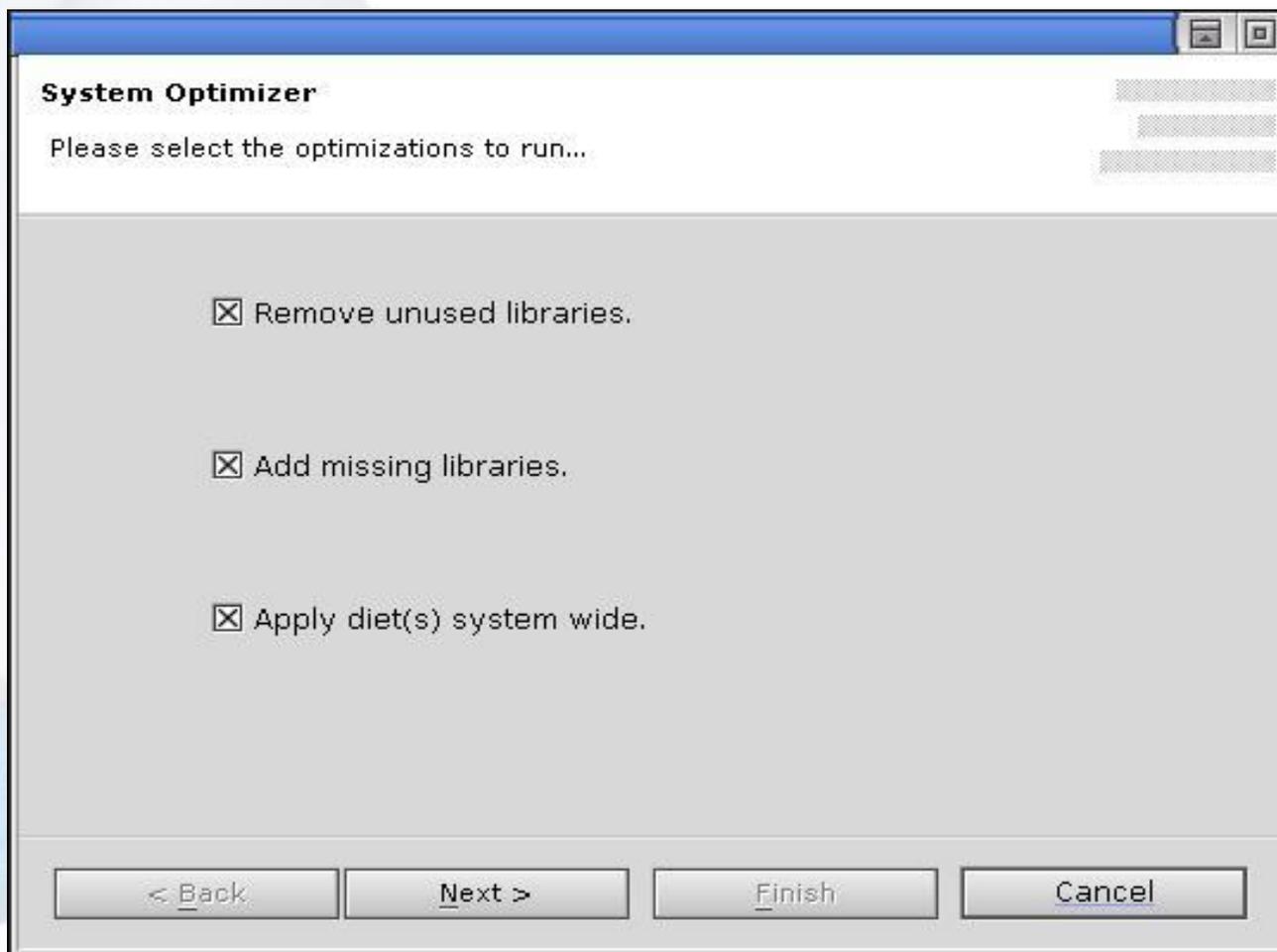
- Построение загрузочных образов системы
- Построение образов ПЗУ
- Оптимизация образа
- Разрешение зависимостей
- Взаимодействие с ROM monitor или QNX IPL на целевой машине для загрузки образа
- Содержит TFTP сервер для загрузки целевых машин, умеющих загружаться по сети с использованием TFTP протокола



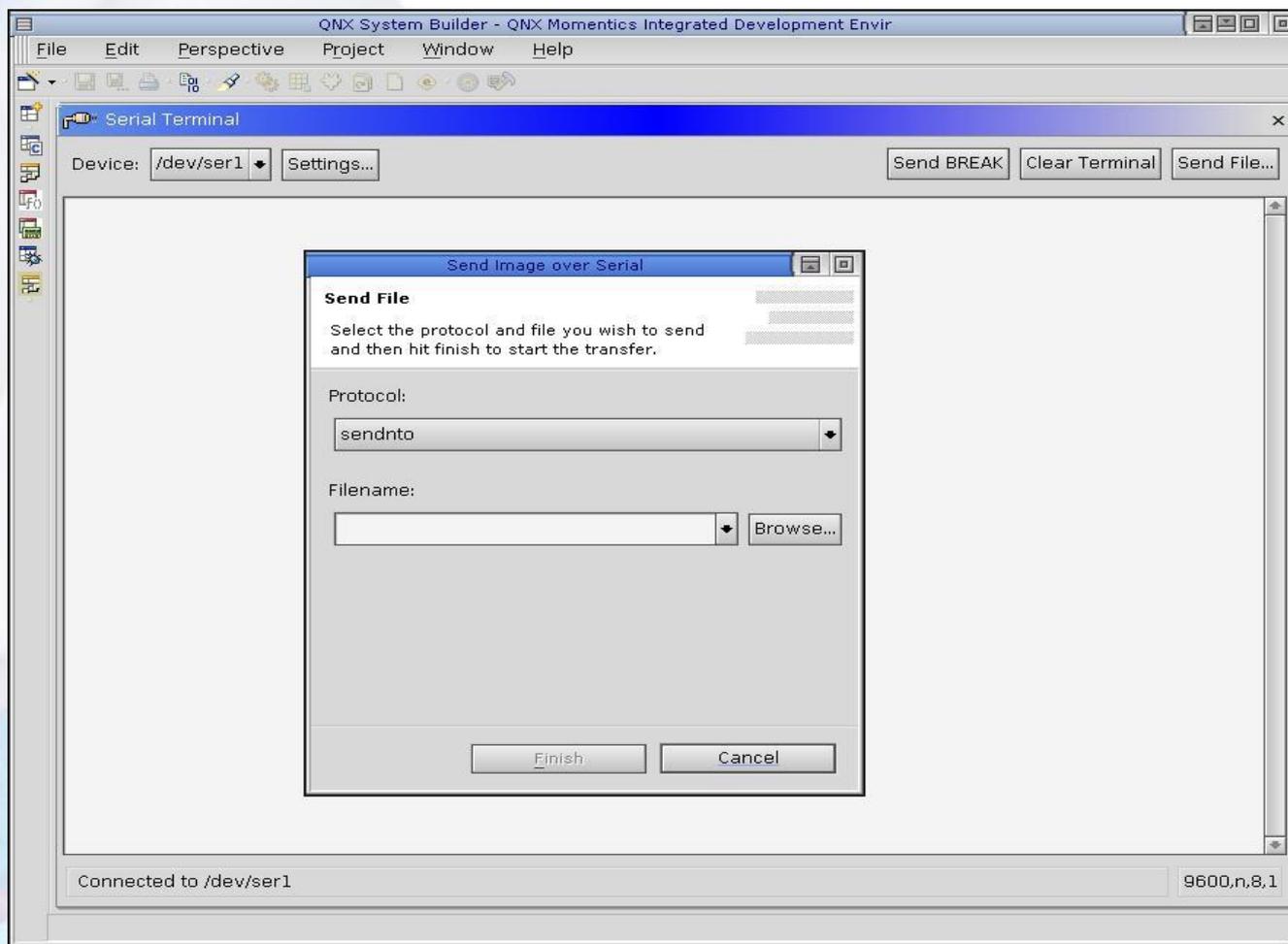
Добавление модулей в образ



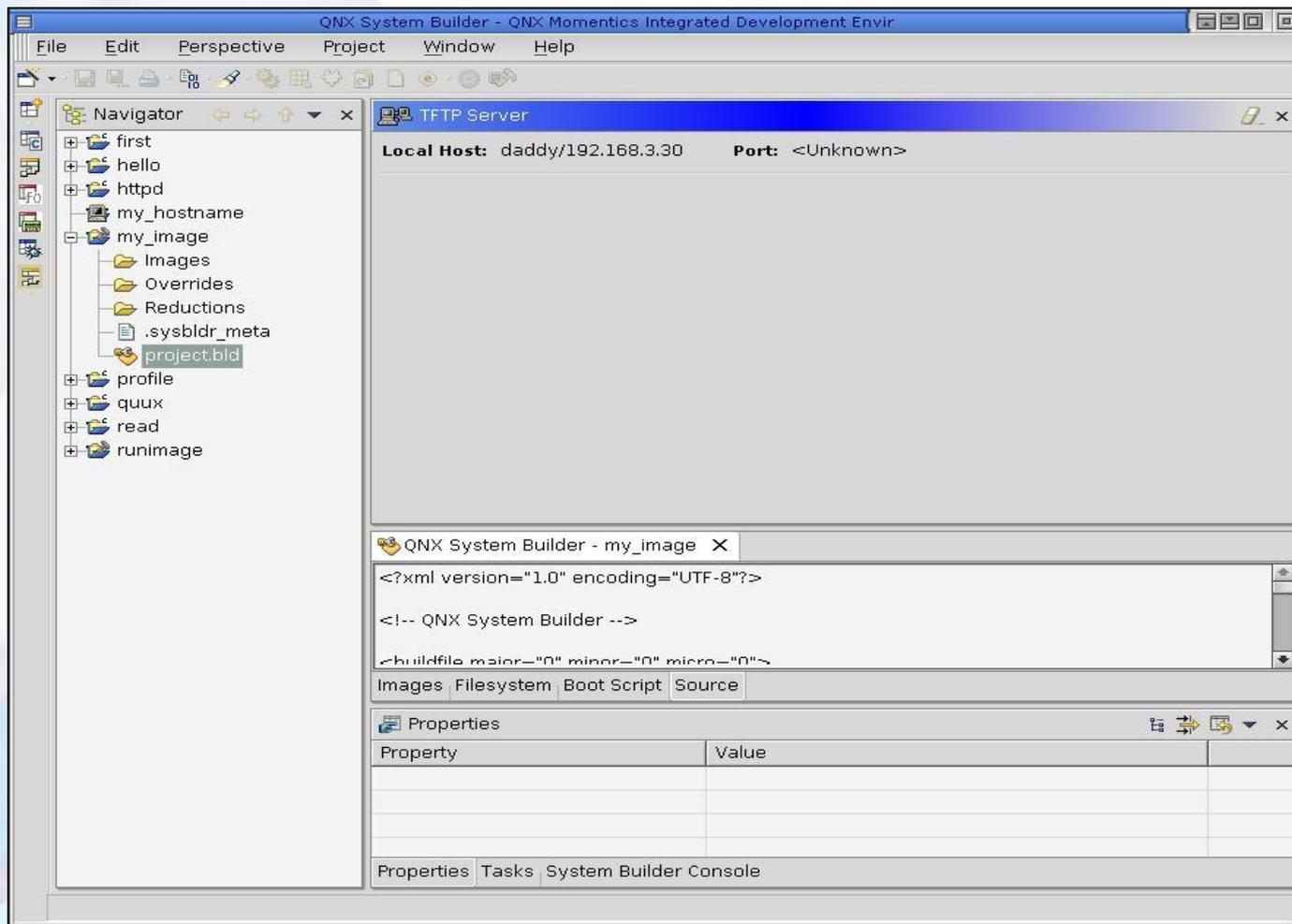
Оптимизация образа



Последовательный терминал



TFTP сервер



The screenshot shows the QNX System Builder IDE interface. The main window is titled "TFTP Server" and displays the following configuration:

- Local Host:** daddy/192.168.3.30
- Port:** <Unknown>

The left sidebar shows a project tree with the following structure:

- first
- hello
- httpd
- my_hostname
- my_image
 - Images
 - Overrides
 - Reductions
 - .sysbldr_meta
 - project.bld
- profile
- quux
- read
- runimage

The bottom panel shows the XML configuration for the "my_image" component:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QNX System Builder -->
<buildfile major="0" minor="0" micro="0">
Images | Filesystem | Boot Script | Source
```

The Properties panel at the bottom is empty, showing a table with columns for Property and Value.

Property	Value

IDE предполагает, что вы будете использовать систему контроля версии для доступа к исходным кодам ваших проектов.

На данный момент IDE поддерживает следующие системы контроля версии:

✓ CVS

✓ Rational ClearCase

Добавление репозитория

New

CVS Repository Location

Remember the location of an existing CVS repository.

Connection type: pserver

User name: anoncvs

Host name: cvs.qnx.com

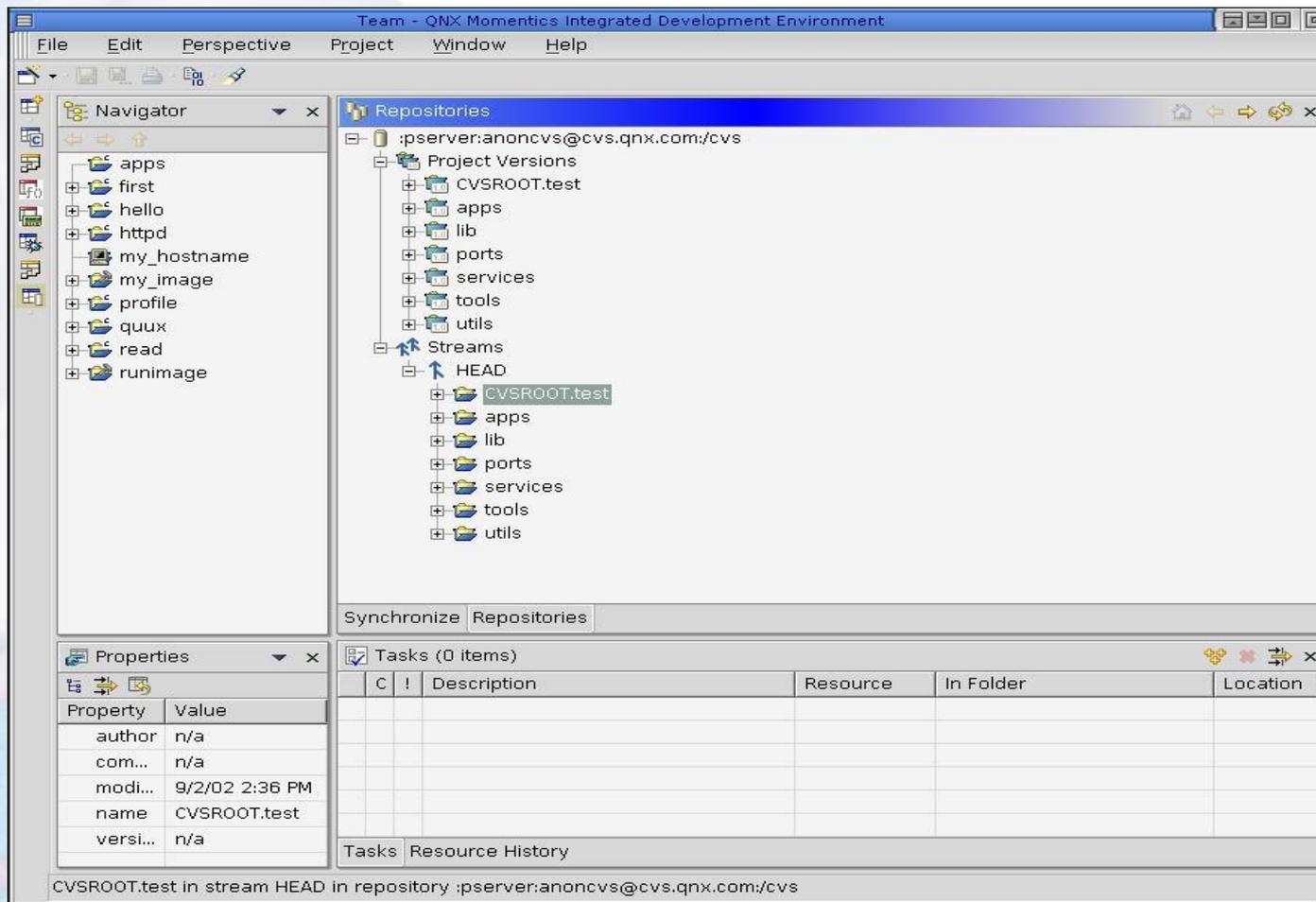
Repository path: /cvs

CVS location: :pserver:anoncvs@cvs.qnx.com:/cvs|

Validate location on finish

Finish Cancel

Team perspective



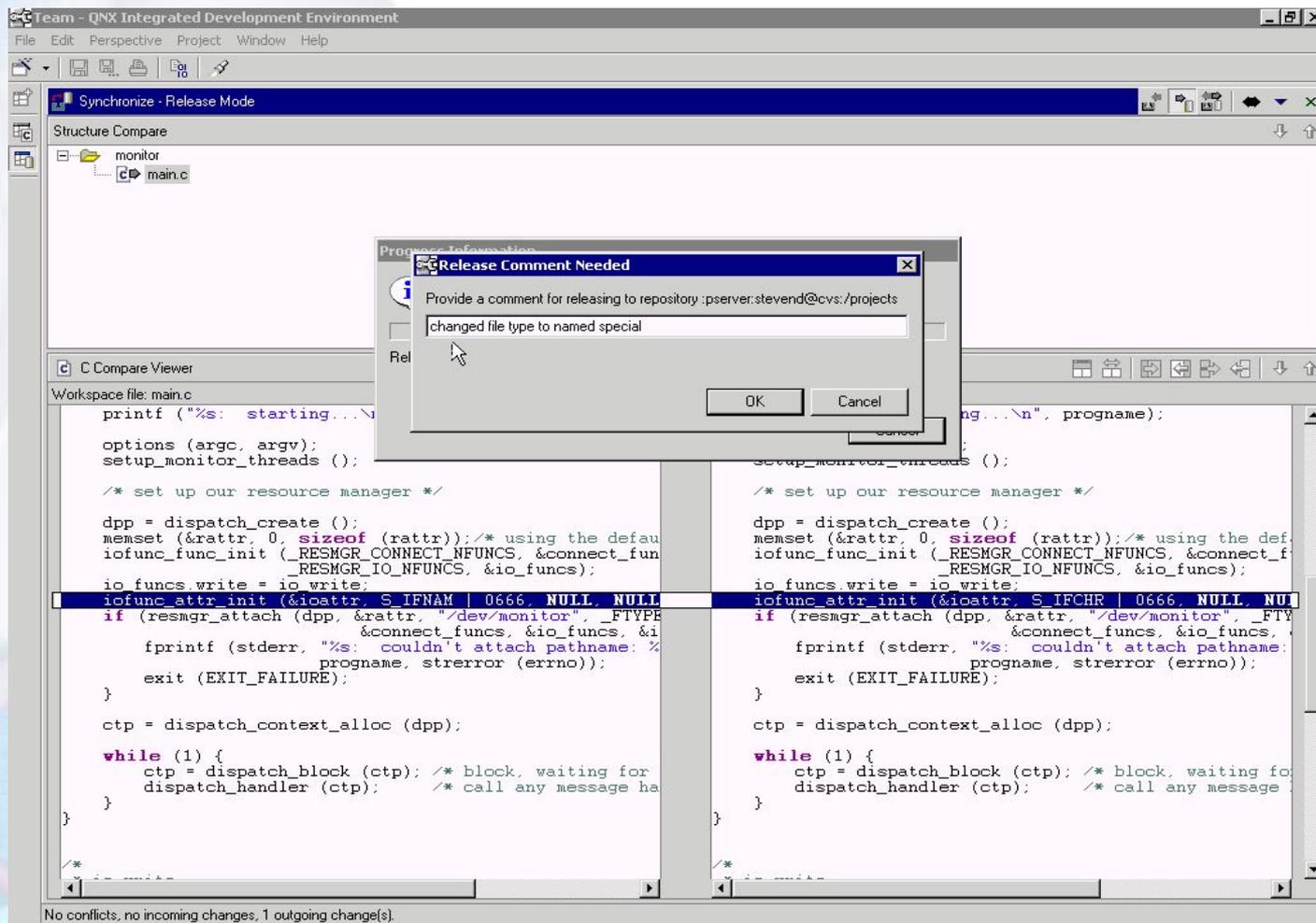
The screenshot displays the Team perspective in the QNX Momentics IDE. The interface is divided into several panes:

- Navigator:** Shows a project tree with folders like apps, first, hello, httpd, my_hostname, my_image, profile, quux, read, and runimage.
- Repositories:** Shows a CVS repository structure for :pserver:anoncvs@cvs.qnx.com/cvs. The tree includes Project Versions, CVSROOT.test, apps, lib, ports, services, tools, utils, Streams, and HEAD. The HEAD folder is expanded to show a sub-tree with CVSROOT.test, apps, lib, ports, services, tools, and utils.
- Properties:** Displays metadata for the selected CVSROOT.test folder:

Property	Value
author	n/a
com...	n/a
modi...	9/2/02 2:36 PM
name	CVSROOT.test
versi...	n/a
- Tasks:** A table with 0 items, showing columns for C, Description, Resource, In Folder, and Location.

At the bottom of the IDE, a status bar indicates: CVSROOT.test in stream HEAD in repository :pserver:anoncvs@cvs.qnx.com/cvs

Синхронизация с репозитарием



The screenshot displays the Team - QNX Integrated Development Environment interface. A 'Synchronize - Release Mode' window is open, showing a 'Structure Compare' view of a project with files 'monitor' and 'main.c'. Below this, a 'C Compare Viewer' shows a side-by-side comparison of the 'main.c' file. A dialog box titled 'Release Comment Needed' is overlaid on the code, prompting the user to provide a comment for releasing to the repository. The comment entered is 'changed file type to named special'. The dialog has 'OK' and 'Cancel' buttons. At the bottom of the IDE, a status bar indicates 'No conflicts, no incoming changes, 1 outgoing change(s)'.

```
printf ("%s: starting...\n", progname);

options (argc, argv);
setup_monitor_threads ();

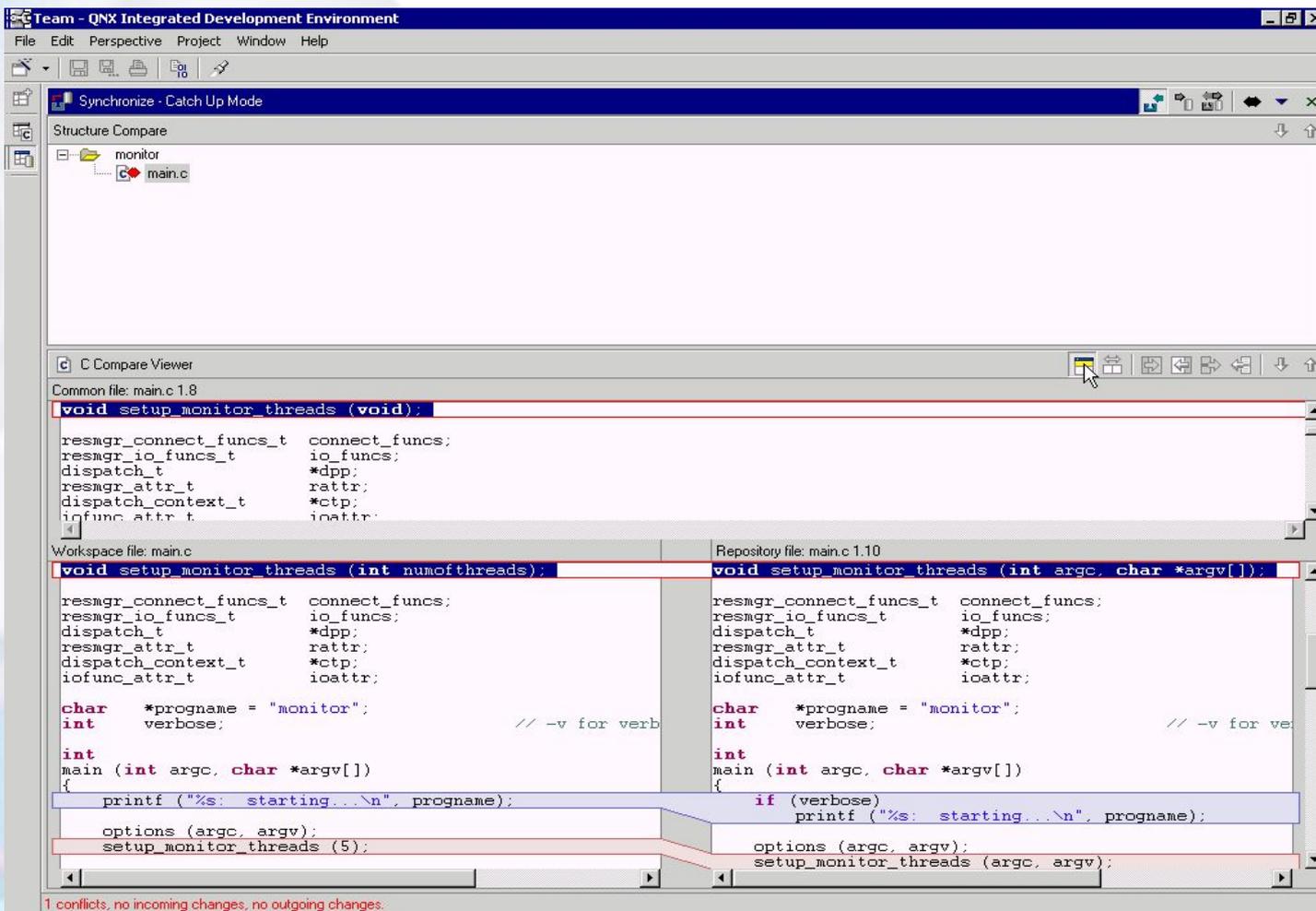
/* set up our resource manager */
dpp = dispatch_create ();
memset (&rattr, 0, sizeof (rattr)); /* using the default
iofunc_func_init (_RESMGR_CONNECT_NFUNCS, &connect_func,
                 _RESMGR_IO_NFUNCS, &io_funcs);
io_funcs.write = io_write;
iofunc_attr_init (&ioattr, S_IFNAM | 0666, NULL, NULL);
if (resmgr_attach (dpp, &rattr, "/dev/monitor", _FTYPE
                 &connect_funcs, &io_funcs, &ioattr))
    fprintf (stderr, "%s: couldn't attach pathname: %s\n",
            progname, strerror (errno));
    exit (EXIT_FAILURE);
}

ctp = dispatch_context_alloc (dpp);

while (1) {
    ctp = dispatch_block (ctp); /* block, waiting for
    dispatch_handler (ctp);    /* call any message handler
}

/*
 *
 */
```

Устранение конфликтов



The screenshot shows the Team IDE interface with a Structure Compare window open. The window compares the file 'main.c' between the workspace (version 1.8) and the repository (version 1.10). The workspace version has a function signature `void setup_monitor_threads (void);`, while the repository version has `void setup_monitor_threads (int argc, char *argv[]);`. The workspace version includes a `main` function that calls `setup_monitor_threads (5);`. The repository version includes a `main` function that calls `setup_monitor_threads (argc, argv);`. The IDE highlights the differences between the two versions.

Team - QNX Integrated Development Environment

Synchronize - Catch Up Mode

Structure Compare

monitor
main.c

C Compare Viewer

Common file: main.c 1.8

```
void setup_monitor_threads (void);  
  
resmgr_connect_funcs_t connect_funcs;  
resmgr_io_funcs_t io_funcs;  
dispatch_t *dpp;  
resmgr_attr_t rattr;  
dispatch_context_t *ctp;  
iofunc_attr_t ioattr;
```

Workspace file: main.c

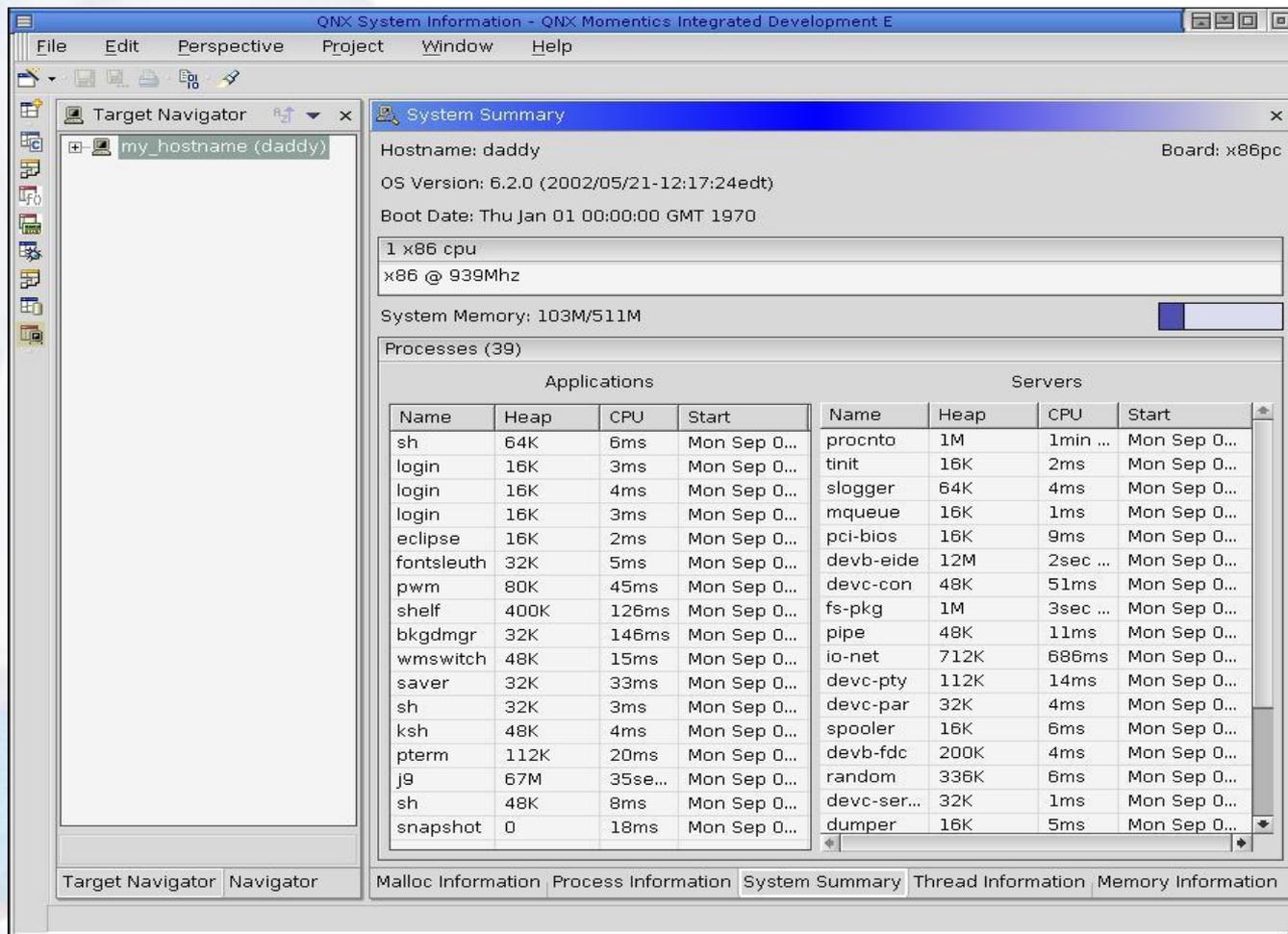
```
void setup_monitor_threads (int numofthreads);  
  
resmgr_connect_funcs_t connect_funcs;  
resmgr_io_funcs_t io_funcs;  
dispatch_t *dpp;  
resmgr_attr_t rattr;  
dispatch_context_t *ctp;  
iofunc_attr_t ioattr;  
  
char *progrname = "monitor";  
int verbose; // -v for verb  
  
int  
main (int argc, char *argv[])  
{  
    printf ("%s: starting...\n", progrname);  
    options (argc, argv);  
    setup_monitor_threads (5);  
}
```

Repository file: main.c 1.10

```
void setup_monitor_threads (int argc, char *argv[]);  
  
resmgr_connect_funcs_t connect_funcs;  
resmgr_io_funcs_t io_funcs;  
dispatch_t *dpp;  
resmgr_attr_t rattr;  
dispatch_context_t *ctp;  
iofunc_attr_t ioattr;  
  
char *progrname = "monitor";  
int verbose; // -v for verb  
  
int  
main (int argc, char *argv[])  
{  
    if (verbose)  
        printf ("%s: starting...\n", progrname);  
    options (argc, argv);  
    setup_monitor_threads (argc, argv);  
}
```

1 conflicts, no incoming changes, no outgoing changes.

System Information Perspective



QNX System Information - QNX Momentics Integrated Development E

File Edit Perspective Project Window Help

Target Navigator my_hostname (daddy)

System Summary

Hostname: daddy Board: x86pc

OS Version: 6.2.0 (2002/05/21-12:17:24edt)

Boot Date: Thu Jan 01 00:00:00 GMT 1970

1 x86 cpu
x86 @ 939Mhz

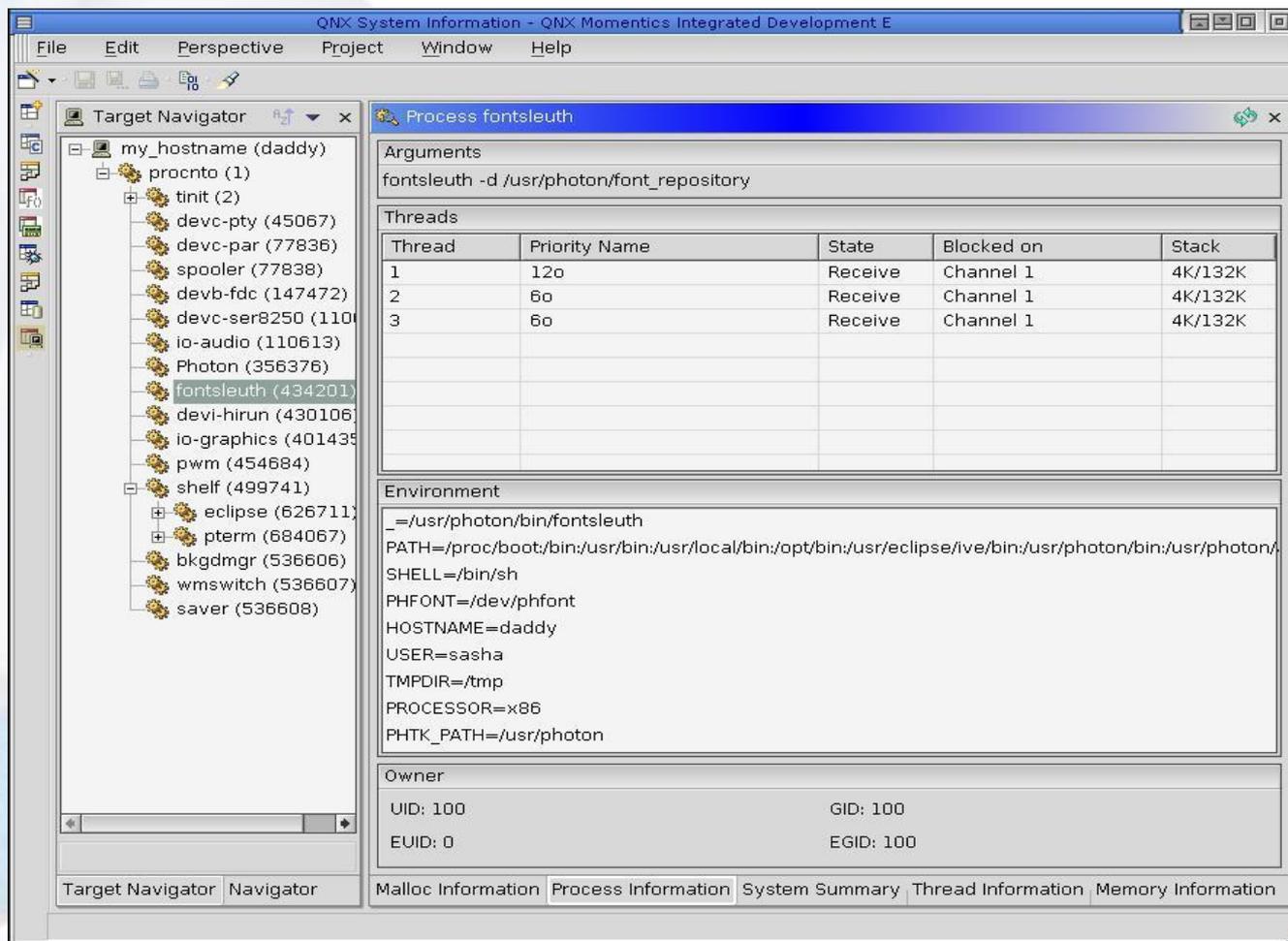
System Memory: 103M/511M

Processes (39)

Applications				Servers			
Name	Heap	CPU	Start	Name	Heap	CPU	Start
sh	64K	6ms	Mon Sep 0...	procnto	1M	1min ...	Mon Sep 0...
login	16K	3ms	Mon Sep 0...	tinit	16K	2ms	Mon Sep 0...
login	16K	4ms	Mon Sep 0...	slogger	64K	4ms	Mon Sep 0...
login	16K	3ms	Mon Sep 0...	mqueue	16K	1ms	Mon Sep 0...
eclipse	16K	2ms	Mon Sep 0...	pci-bios	16K	9ms	Mon Sep 0...
fontsl euth	32K	5ms	Mon Sep 0...	devb-eide	12M	2sec ...	Mon Sep 0...
pwm	80K	45ms	Mon Sep 0...	devc-con	48K	51ms	Mon Sep 0...
shelf	400K	126ms	Mon Sep 0...	fs-pkg	1M	3sec ...	Mon Sep 0...
bkgdmgr	32K	146ms	Mon Sep 0...	pipe	48K	11ms	Mon Sep 0...
wmswitch	48K	15ms	Mon Sep 0...	io-net	712K	686ms	Mon Sep 0...
saver	32K	33ms	Mon Sep 0...	devc-pty	112K	14ms	Mon Sep 0...
sh	32K	3ms	Mon Sep 0...	devc-par	32K	4ms	Mon Sep 0...
ksh	48K	4ms	Mon Sep 0...	spooler	16K	6ms	Mon Sep 0...
pterm	112K	20ms	Mon Sep 0...	devb-fdc	200K	4ms	Mon Sep 0...
j9	67M	35se...	Mon Sep 0...	random	336K	6ms	Mon Sep 0...
sh	48K	8ms	Mon Sep 0...	devc-ser...	32K	1ms	Mon Sep 0...
snapshot	0	18ms	Mon Sep 0...	dumper	16K	5ms	Mon Sep 0...

Target Navigator Navigator Malloc Information Process Information System Summary Thread Information Memory Information

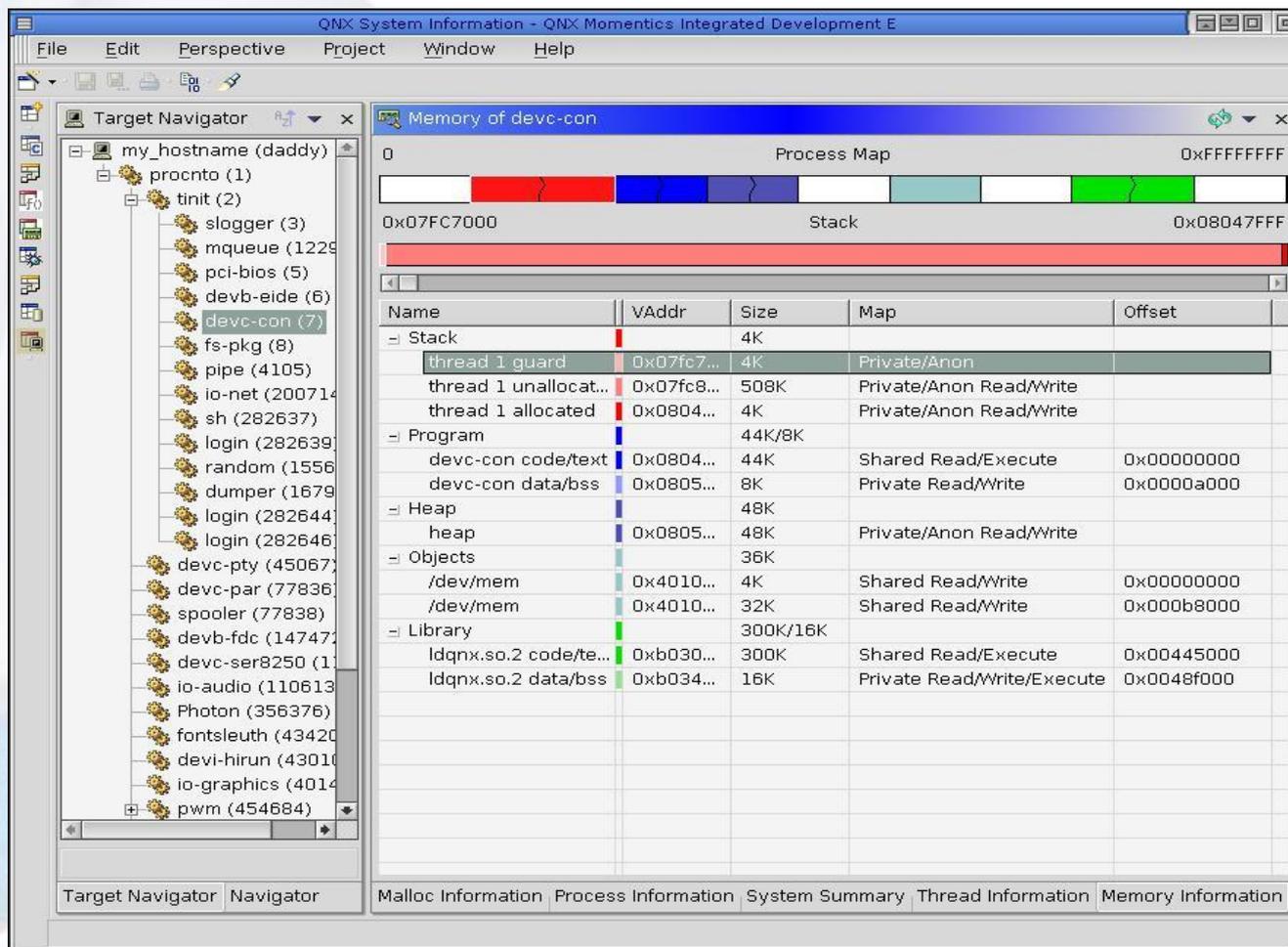
Информация о процессах



The screenshot displays the QNX System Information tool interface. The main window is titled "QNX System Information - QNX Momentics Integrated Development E". The interface is divided into several panes:

- Target Navigator:** A tree view showing the system hierarchy. The process "fontsleuth (434201)" is selected.
- Process fontsleuth:** The main pane showing details for the selected process.
 - Arguments:** fontsleuth -d /usr/photon/font_repository
 - Threads:** A table showing three threads, all in a "Receive" state, blocked on "Channel 1".
 - Environment:** A list of environment variables including PATH, SHELL, PHFONT, HOSTNAME, USER, TMPDIR, PROCESSOR, and PHTK_PATH.
 - Owner:** UID: 100, GID: 100, EUID: 0, EGID: 100.
- Bottom Panel:** A set of tabs for "Malloc Information", "Process Information", "System Summary", "Thread Information", and "Memory Information".

Информация о памяти

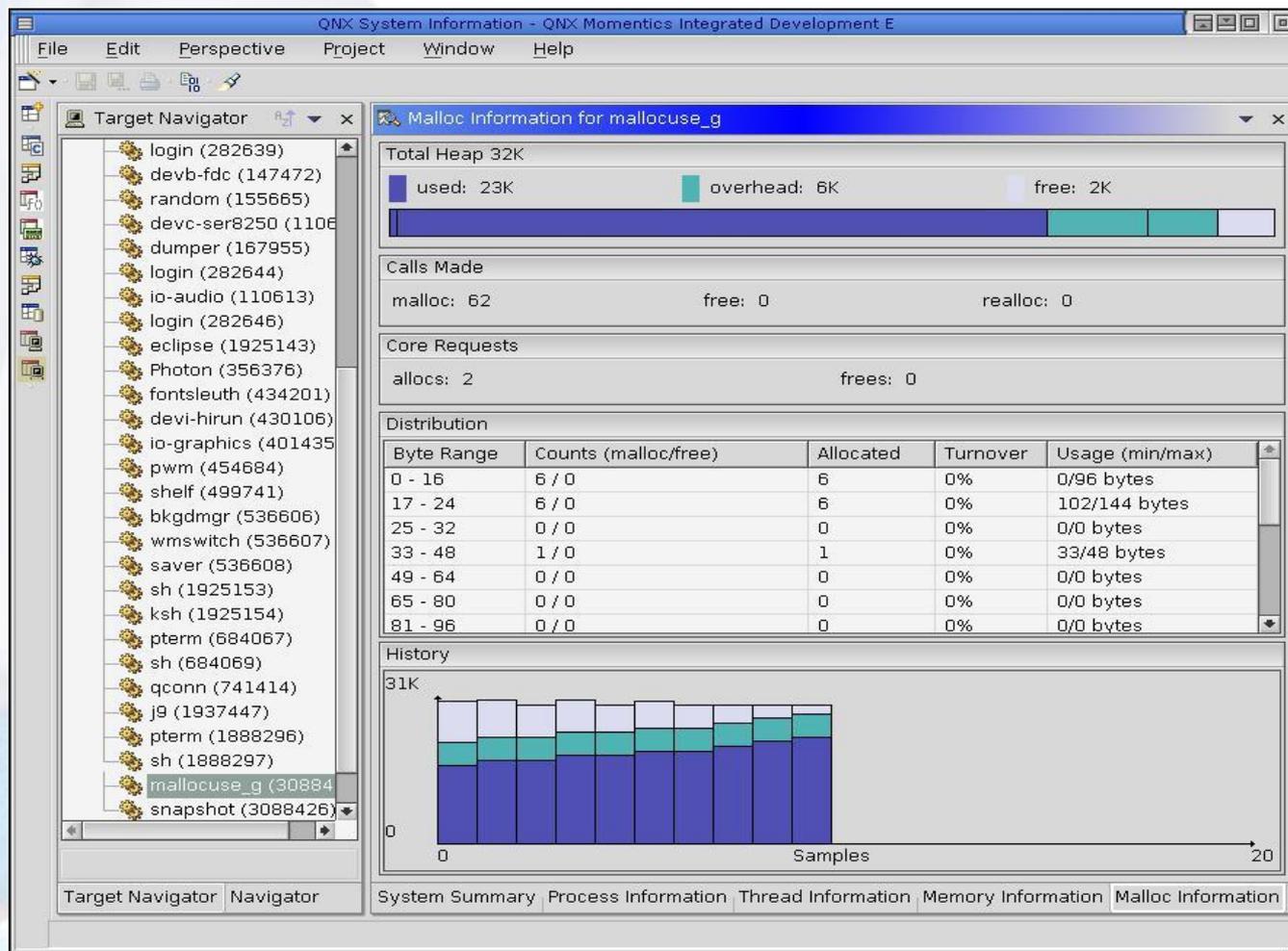


The screenshot displays the QNX System Information tool interface. The main window is titled "Memory of devc-con". On the left, the "Target Navigator" shows a tree view of the system hierarchy, with "devc-con (7)" selected. The main area shows a "Process Map" at the top, followed by a "Stack" section with a red bar representing the stack memory. Below this is a table listing memory segments for the process.

Name	VAddr	Size	Map	Offset
Stack		4K		
thread 1 guard	0x07fc7...	4K	Private/Anon	
thread 1 unallocat...	0x07fc8...	508K	Private/Anon Read/Write	
thread 1 allocated	0x0804...	4K	Private/Anon Read/Write	
Program		44K/8K		
devc-con code/text	0x0804...	44K	Shared Read/Execute	0x00000000
devc-con data/bss	0x0805...	8K	Private Read/Write	0x0000a000
Heap		48K		
heap	0x0805...	48K	Private/Anon Read/Write	
Objects		36K		
/dev/mem	0x4010...	4K	Shared Read/Write	0x00000000
/dev/mem	0x4010...	32K	Shared Read/Write	0x0000b8000
Library		300K/16K		
ldqnx.so.2 code/te...	0xb030...	300K	Shared Read/Execute	0x00445000
ldqnx.so.2 data/bss	0xb034...	16K	Private Read/Write/Execute	0x0048f000

At the bottom of the window, there are tabs for "Malloc Information", "Process Information", "System Summary", "Thread Information", and "Memory Information".

Информация о выделении памяти



QNX System Information - QNX Momentics Integrated Development E

File Edit Perspective Project Window Help

Target Navigator

- login (282639)
- devb-fdc (147472)
- random (155665)
- devc-ser8250 (1106)
- dumper (167955)
- login (282644)
- io-audio (110613)
- login (282646)
- eclipse (1925143)
- Photon (356376)
- fontslouth (434201)
- devi-hirun (430106)
- io-graphics (401435)
- pwm (454684)
- shelf (499741)
- bkgdmgr (536606)
- wmswitch (536607)
- saver (536608)
- sh (1925153)
- ksh (1925154)
- pterm (684067)
- sh (684069)
- qconn (741414)
- j9 (1937447)
- pterm (1888296)
- sh (1888297)
- mallocuse_g (30884)
- snapshot (3088426)

Malloc Information for mallocuse_g

Total Heap 32K

used: 23K overhead: 6K free: 2K

Calls Made

malloc: 62 free: 0 realloc: 0

Core Requests

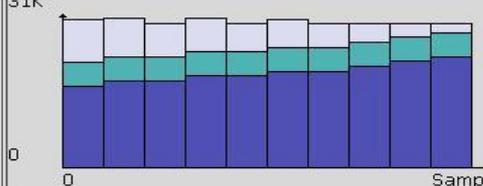
allocs: 2 frees: 0

Distribution

Byte Range	Counts (malloc/free)	Allocated	Turnover	Usage (min/max)
0 - 16	6 / 0	6	0%	0/96 bytes
17 - 24	6 / 0	6	0%	102/144 bytes
25 - 32	0 / 0	0	0%	0/0 bytes
33 - 48	1 / 0	1	0%	33/48 bytes
49 - 64	0 / 0	0	0%	0/0 bytes
65 - 80	0 / 0	0	0%	0/0 bytes
81 - 96	0 / 0	0	0%	0/0 bytes

History

31K



0

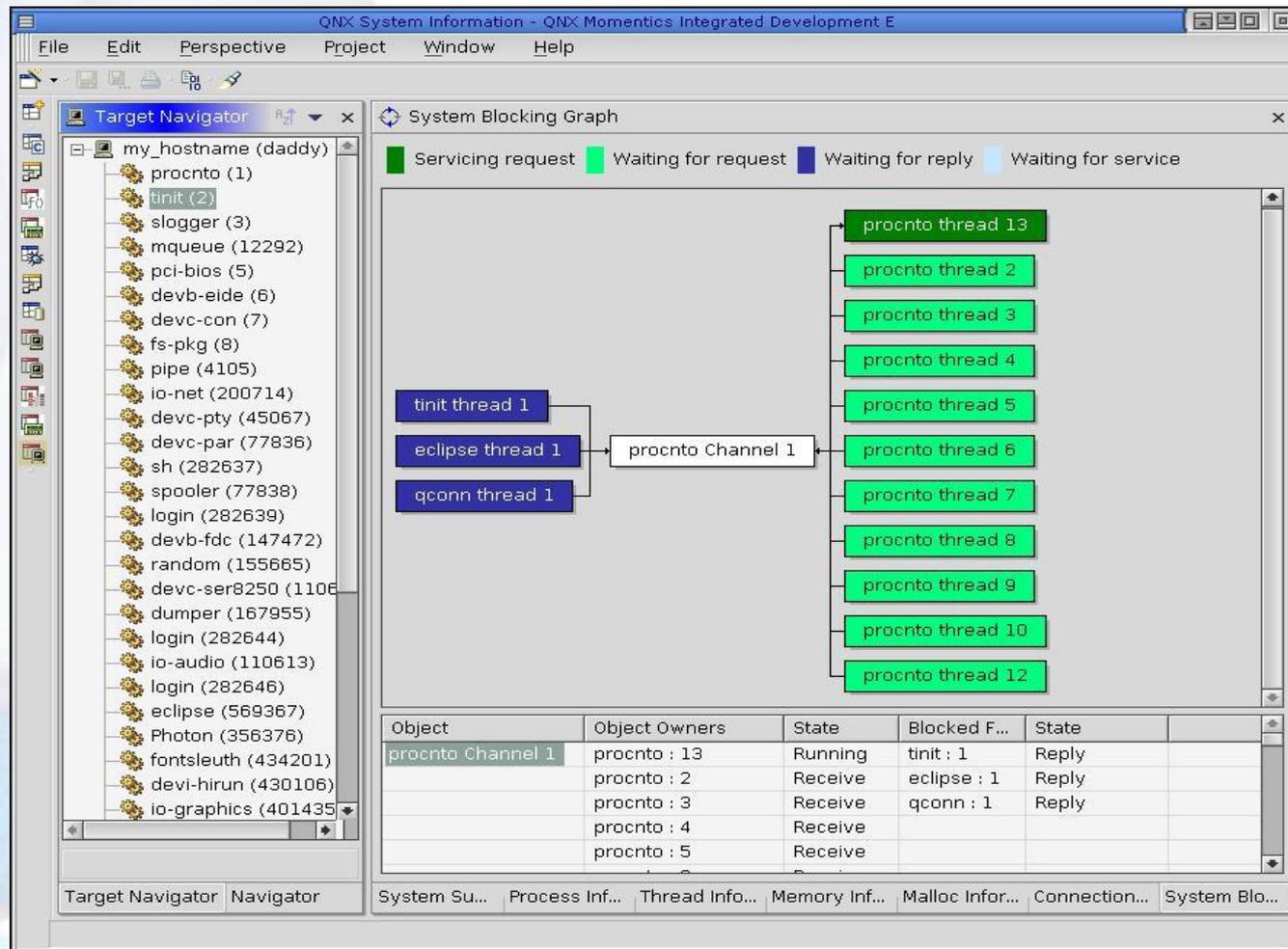
Samples

20

Target Navigator Navigator

System Summary Process Information Thread Information Memory Information Malloc Information

System Blocking Graph



Target Navigator

- my_hostname (daddy)
 - procnto (1)
 - tinit (2)
 - slogger (3)
 - mqueue (12292)
 - pci-bios (5)
 - devb-eide (6)
 - devc-con (7)
 - fs-pkg (8)
 - pipe (4105)
 - io-net (200714)
 - devc-pty (45067)
 - devc-par (77836)
 - sh (282637)
 - spooler (77838)
 - login (282639)
 - devb-fdc (147472)
 - random (155665)
 - devc-ser8250 (110611)
 - dumper (167955)
 - login (282644)
 - io-audio (110613)
 - login (282646)
 - eclipse (569367)
 - Photon (356376)
 - fontsleuth (434201)
 - devi-hirun (430106)
 - io-graphics (401435)

System Blocking Graph

Legend: Servicing request (dark green), Waiting for request (light green), Waiting for reply (dark blue), Waiting for service (light blue)

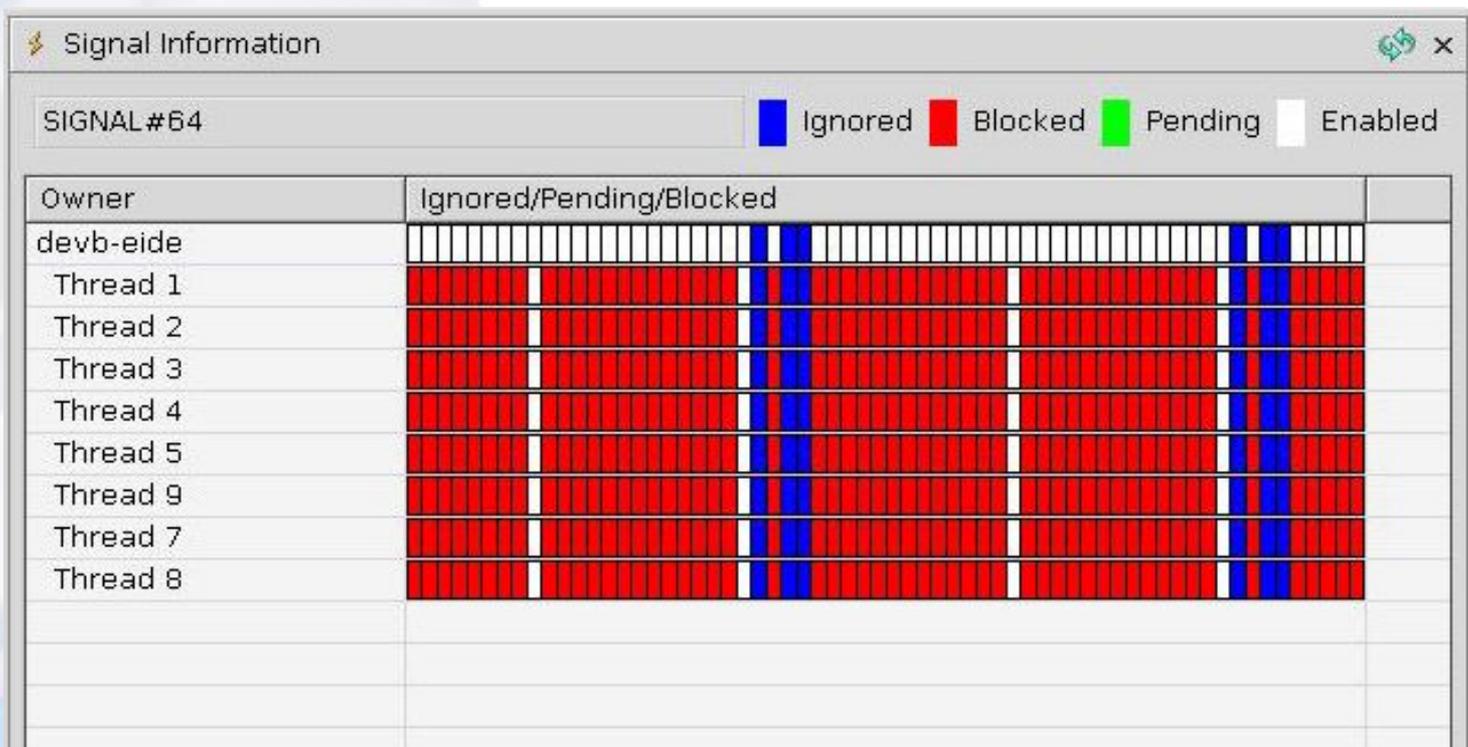
Graph components:

- tinit thread 1 (dark blue)
- eclipse thread 1 (dark blue)
- qconn thread 1 (dark blue)
- procnto Channel 1 (white)
- procnto thread 2-13 (light green)

Object	Object Owners	State	Blocked F...	State
procnto Channel 1	procnto : 13	Running	tinit : 1	Reply
	procnto : 2	Receive	eclipse : 1	Reply
	procnto : 3	Receive	qconn : 1	Reply
	procnto : 4	Receive		
	procnto : 5	Receive		

Target Navigator Navigator System Su... Process Inf... Thread Info... Memory Inf... Malloc Infor... Connection... System Blo...

Информация о сигналах



Profiling (профилирование) – это определение участков кода, где ваша программа проводит большинство времени

Profiling помогает вам:

- ✓ сократить количество вызовов функций
- ✓ оптимизировать алгоритмы

Данные profiling состоят из двух частей:

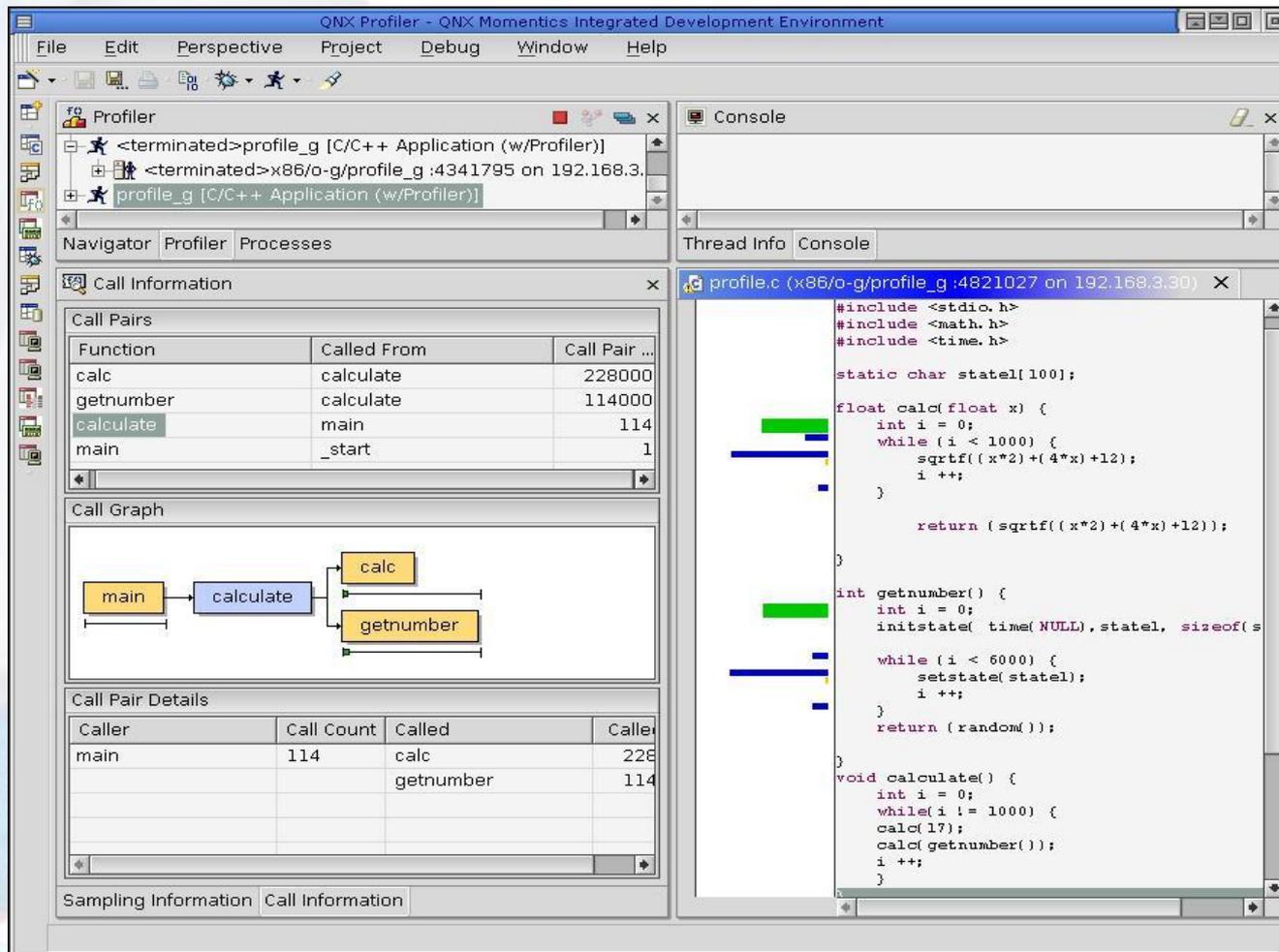
Sampling:

- ✓ Какой код выполняется
- ✓ Насколько часто выполняется код

Информация о вызовах:

- ✓ Количество вызовов функций
- ✓ Информация об инициаторе вызова

Profile perspective



The screenshot displays the QNX Profiler interface within the QNX Momentics IDE. The main window is titled "QNX Profiler - QNX Momentics Integrated Development Environment". The interface is divided into several panes:

- Profiler:** Shows the current application being profiled, "profile_g [C/C++ Application (w/Profiler)]".
- Call Information:** Contains two sub-panes:
 - Call Pairs:** A table showing the relationship between functions and their callers.
 - Call Graph:** A visual representation of the call pairs, showing a flow from main to calculate, which then branches to calc and getnumber.
 - Call Pair Details:** A table providing more granular data for the selected call pair.
- Code Editor:** Displays the source code for "profile.c", which includes functions like calc, getnumber, and calculate. The code is annotated with colored bars (green and blue) indicating execution time for different sections.

Function	Called From	Call Pair ...
calc	calculate	228000
getnumber	calculate	114000
calculate	main	114
main	_start	1

Caller	Call Count	Called	Caller
main	114	calc	228000
		getnumber	114000

Sampling information

Sampling Information					
Function	Total Time (s)	Time since ...	Call Count	usec/Call	% Time Usage
setstate	32.886	0.000			
sqrtf	19.100	0.000			
_FDunscale	4.535	0.000			
_FDscale	3.735	0.000			
calc	3.177	0.000	680000	4.672	
getnumber	2.966	0.000	340000	8.724	
.plt	0.650	0.000			
sqrtf	0.292	0.000			
random	0.213	0.000			
srandom	0.066	0.000			
_udivdi3	0.007	0.000			
initstate	0.006	0.000			
time	0.004	0.000			
pthread_self	0.002	0.000			
calculate	0.001	0.000	340	2.941	
main	0.000	0.000	1		

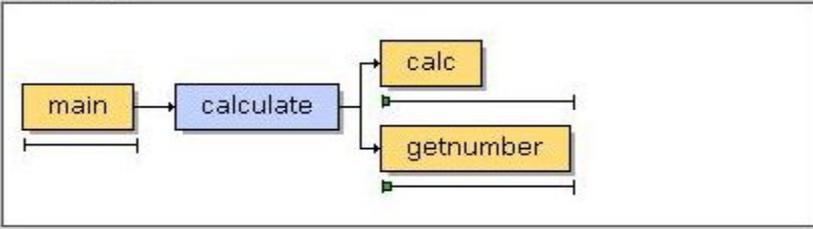
Информация о вызовах

Call Information

Call Pairs

Function	Called From	Call Pair ...
calc	calculate	228000
getnumber	calculate	114000
calculate	main	114
main	_start	1

Call Graph



```
graph LR; main[main] --> calculate[calculate]; calculate --> calc[calc]; calculate --> getnumber[getnumber];
```

Call Pair Details

Caller	Call Count	Called	Called Count
main	114	calc	228000
		getnumber	114000

Sampling Information | Call Information

Вывод информации о времени исполнения в исходном коде

```
profile.c (x86/o-g/profile_g :4821027 on 192.168.3.30) X
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

static char statel[100];

float calc(float x) {
    int i = 0;
    while (i < 1000) {
        sqrtf((x*2)+(4*x)+12);
        i ++;
    }

    return (sqrtf((x*2)+(4*x)+12));
}

int getnumber() {
    int i = 0;
    initstate( time(NULL),statel, sizeof(statel));

    while (i < 6000) {
        setstate(statel);
        i ++;
    }
    return (random());
}

void calculate() {
    int i = 0;
    while(i != 1000) {
        calc(17);
        calc(getnumber());
        i ++;
    }
}

int main(int argc, char *argv[]) {
    printf("Welcome to the QNX IDE\n");
}
```

Спасибо!



SWD Software Ltd.

Официальный дистрибьютор QNX

196135, Санкт-Петербург, пр. Юрия Гагарина 23

тел.: (812) 102-0833

тел.: (812) 373-0260

факс: (812) 373-0497

web: <http://www.swd.ru/qnx/>

e-mail: qnx@swd.ru