

Разработка приложений для операционной системы реального времени «Нейтрино»

Владимир Махилёв

ООО «СВД ВС»

Руководитель группы разработки

Отдел операционных систем



«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
- Отделы: разработка ОС, картография, АСУ ТП
- 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»



2

- Картографический пакет



- ИИ «Синаптика»



Защищенная операционная система реального времени «Нейтрино»

- **архитектура на основе микроядра**
 - отказоустойчивость, простота, надёжность
- **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, в планах – RISC-V
- **российские требования**
 - ФСТЭК - ОС типа А и В второго класса защиты
 - сертификат соответствия Минобороны
 - входит в реестр ПО Минцифры



Поддержка российских процессоров и архитектур в «Нейтрино»



«Эльбрус» («МЦСТ»)

- Эльбрус-1С+, 2С+, 4С, 8С
- СнК Эльбрус-2С3 - в работе



«КОМДИВ64» (НИИСИ РАН)

- СнК 1890BM108 (MIPS совместимый)
- 1890BM8Я (MIPS совместимый)



«Мультикор» («ЭЛВИС»)

- MCom-02 (ARMv7)
- NVCom-02T (MIPS)
- RoboDeus (MIPS) - в работе
- СКИФ (ARMv8) - на уровне архитектуры



«Байкал Электроникс»

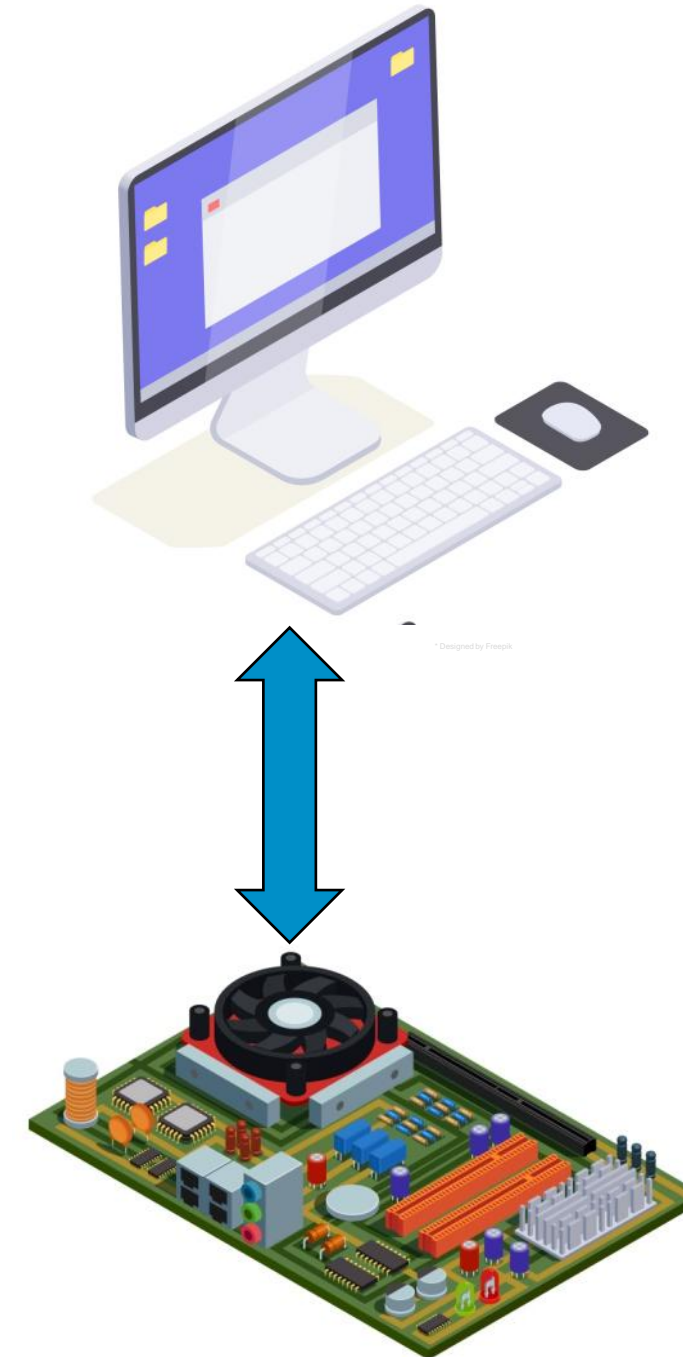
- BE-T1000 (Baikal-T1, MIPS32)
- Baikal-M (ARMv8) – на уровне архитектуры



Комплект разработчика ЗОСРВ «Нейтрино»

Кросс-разработка

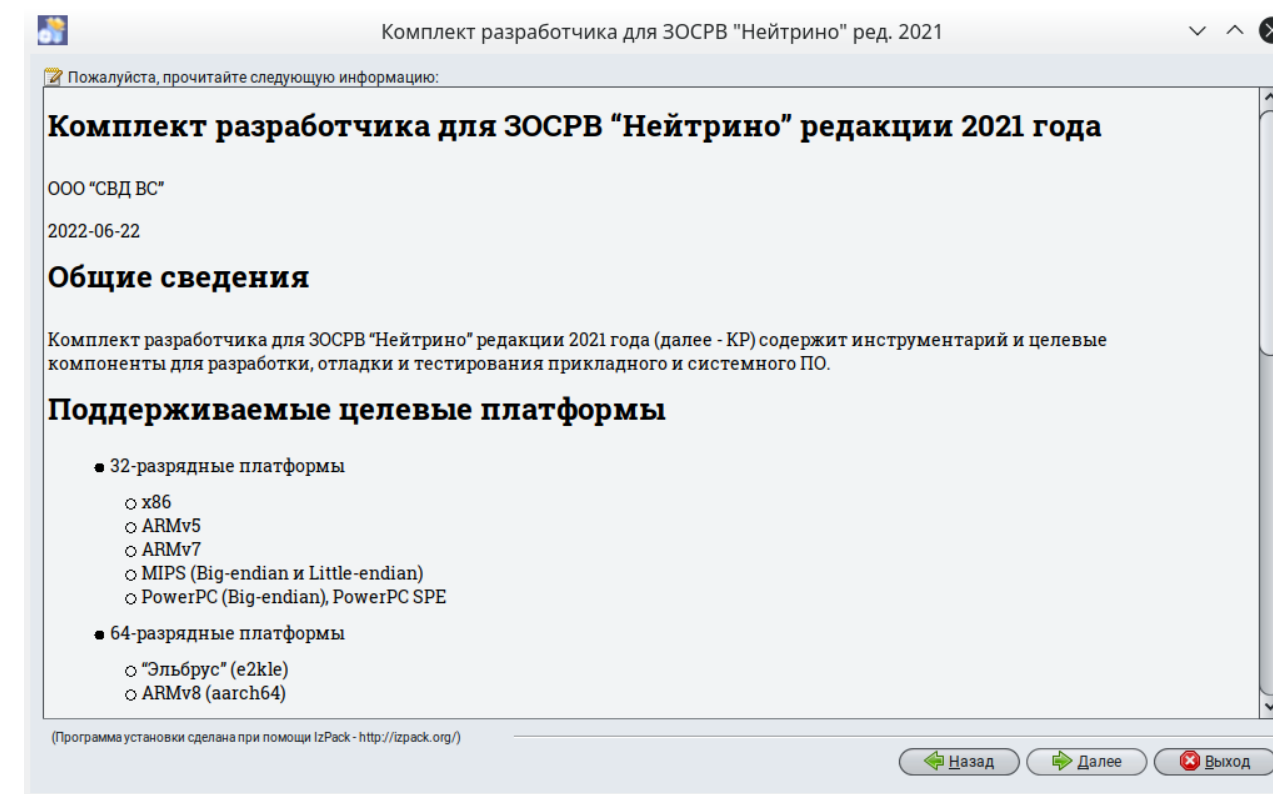
- Инструментальная система хост-система
 - GNU/Linux
 - средства разработки
 - IDE
- Целевая система
 - «Нейтрино»
 - отладочная консоль
 - сеть



Комплект разработчика ЗОСРВ «Нейтрино»

Требования к инструментальной системе

- Требования к дистрибутиву GNU/Linux
 - 64x разрядный дистрибутив - используется 64x разрядный тулчейн (GNU gcc)
 - glibc >= 2.19
 - Astra Linux SE 1.6, Альт Рабочая Станция, Debian >= 8, Ubuntu >= 16.04
- Инсталлятор
 - инструментарий для хост-системы
 - компоненты целевой системы
 - графическая версия
 - консольная версия (--console)



Комплект разработчика ЗОСРВ «Нейтрино»

Поддержка российских дистрибутивов Linux

- **Astra Linux**

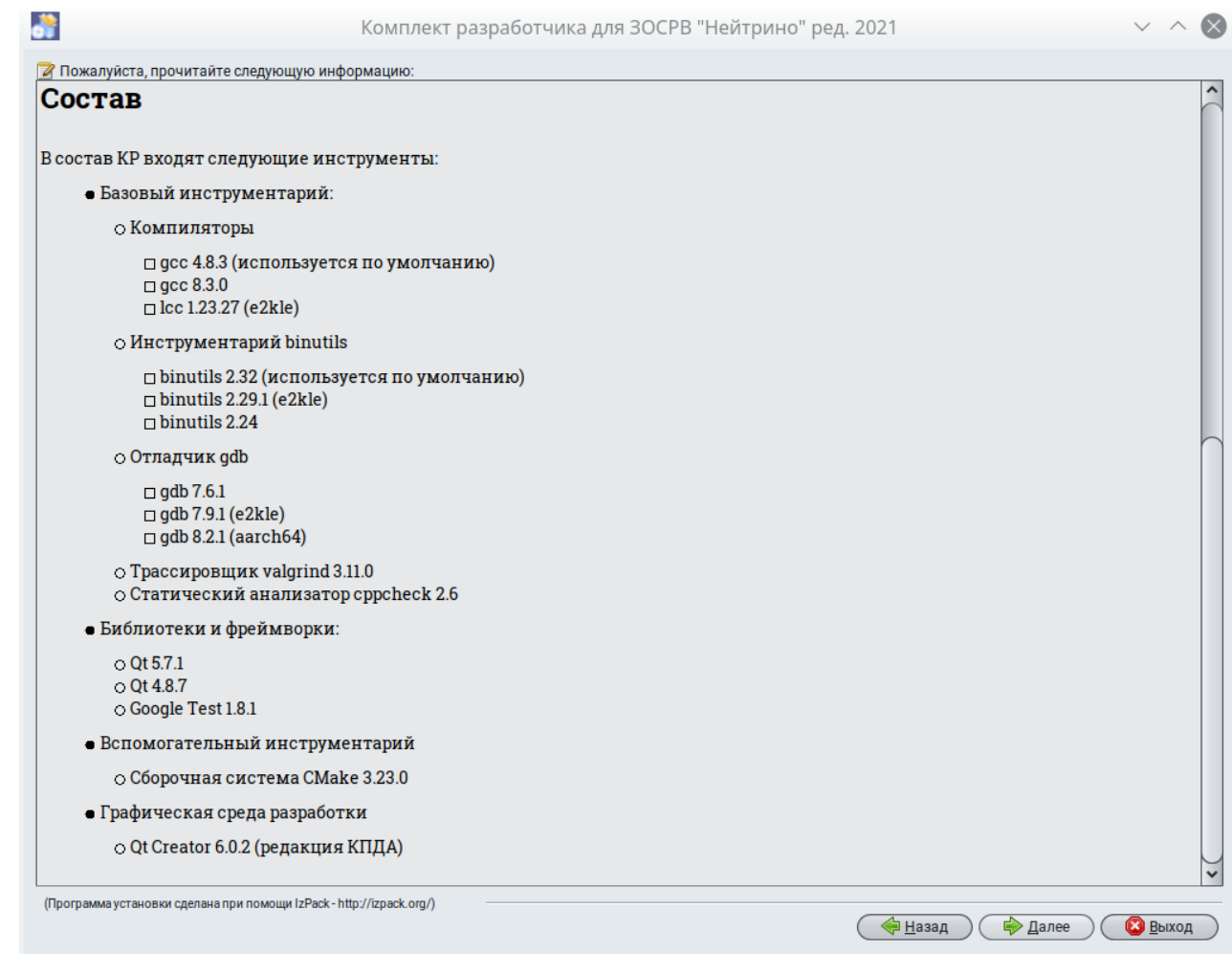
- Astra Linux SE 1.6
- партнерское соглашение с ГК «Астра»
- разработка ПО ответственного назначения комплектом разработчика «Нейтрино» на хост-платформе Astra Linux
- Совместимость по мандатным меткам протоколов сетевого взаимодействия



Комплект разработчика «Нейтрино»

Набор инструментальных средств

- компиляторы **gcc 8.3** и gcc 4.8.3
 - nto*-gcc и nto*-g++ для архитектур x86, ARMv7, AArch64, MIPS LE/BE, PowerPC BE/SPE
 - Icc для Эльбрус
- binutils
- gdb
- GNU make
- статический анализатор cppcheck
- трассировщик Valgrind
- библиотеки модульного тестирования
- CUnit, CppUnit и GoogleTest
- утилиты для сборки загрузочного образа ОС и образа ФС



Комплект разработчика для ЗОСРВ «Нейтрино»

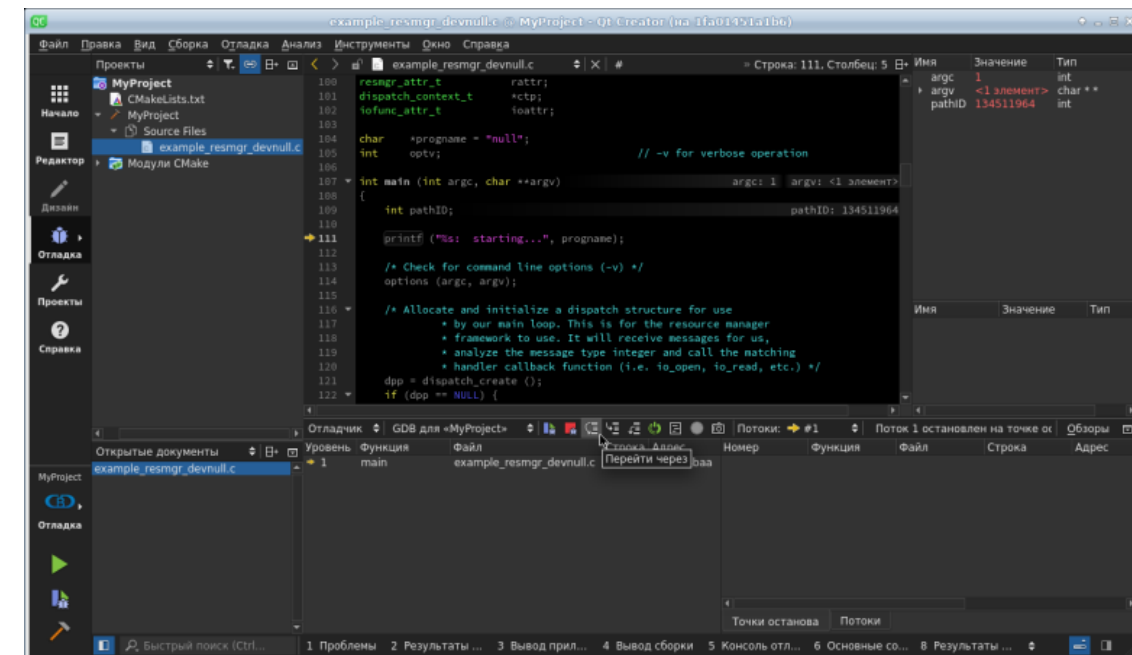
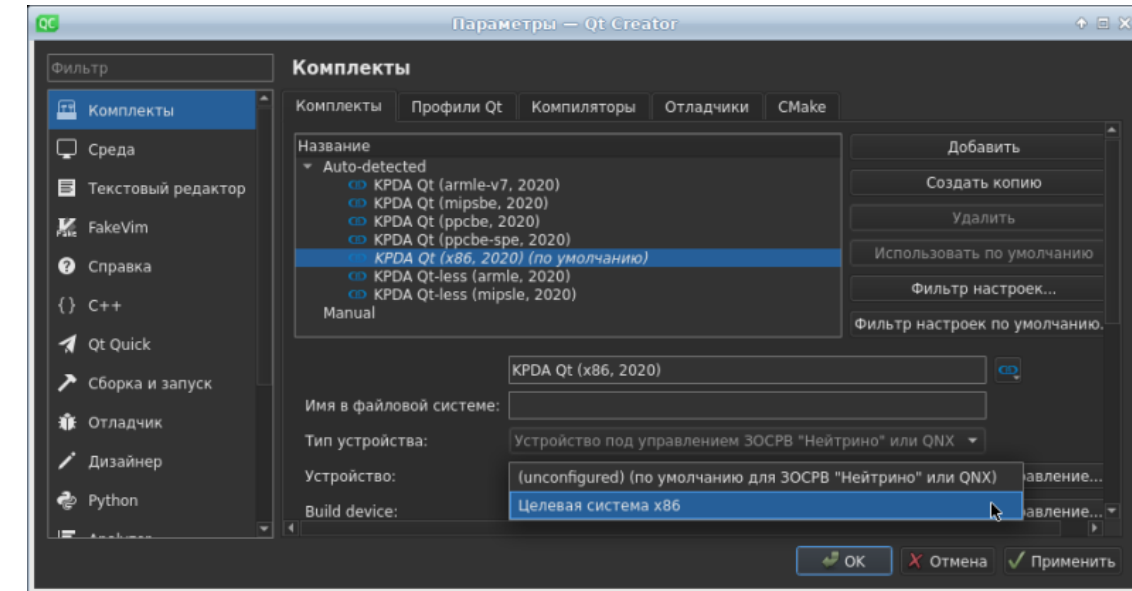
Языки программирования и стандарты

- совместимость с группой стандартов **POSIX** для ОС
- **C/C++**
 - `-std=c*` или `-std=gnu*`
 - C99, C11, C17
 - C++11, C++17, boost
 - OpenMP, OpenMPI
 - Address Sanitizer, Valgrind
- **Python 3.9.0**
 - кросс-компиляция расширений Python - `cross-python3`



Комплект разработчика для ЗОСРВ «Нейтрино» Qt Creator

- IDE на базе **QT Creator 6.0.2**
 - инструментарий для «Нейтрино»
 - компиляторы
 - целевые библиотеки Qt 4/5
- поддержка CMake и qmake проектов
- переключение версий инструментария
- взаимодействие с целевыми системами (удалённый запуск и отладка)
- встроенные примеры приложений для «Нейтрино»
- локализация



Комплект разработчика для ЗОСРВ «Нейтрино»

Сборочная система для всех процессорных архитектур

- Сборка проекта для нескольких или всех поддерживаемых процессорных архитектур
 - компиляторы
 - nto*-gcc и nto*-g++ для x86, ARMv8 (AArch64), ARMv7, MIPS LE/BE, PowerPC BE/SPE
 - Icc для Эльбрус
 - скрипт switch_devtools для переключения версий компилятора
- Сборочные системы
 - make/Makefile
 - GNU Autotools (autoconf/automake)
 - CMake
 - Meson
 - qmake (Qt 4.x и 5.x)



Пример сборки проекта с открытым исходным кодом

- Копируем оригинальные исходные коды

```
$ git clone https://git.code.sf.net/p/libpng/code libpng-code
```

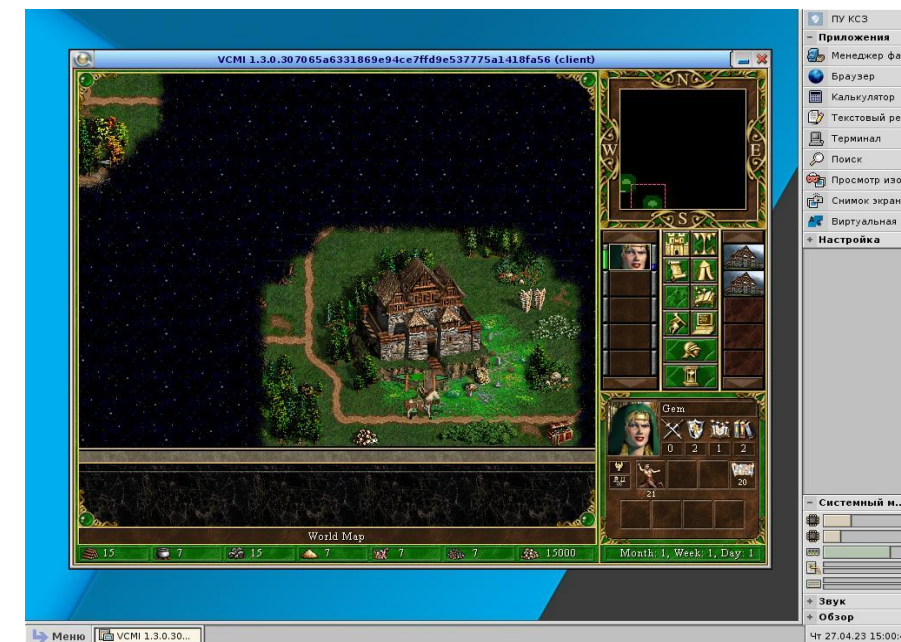
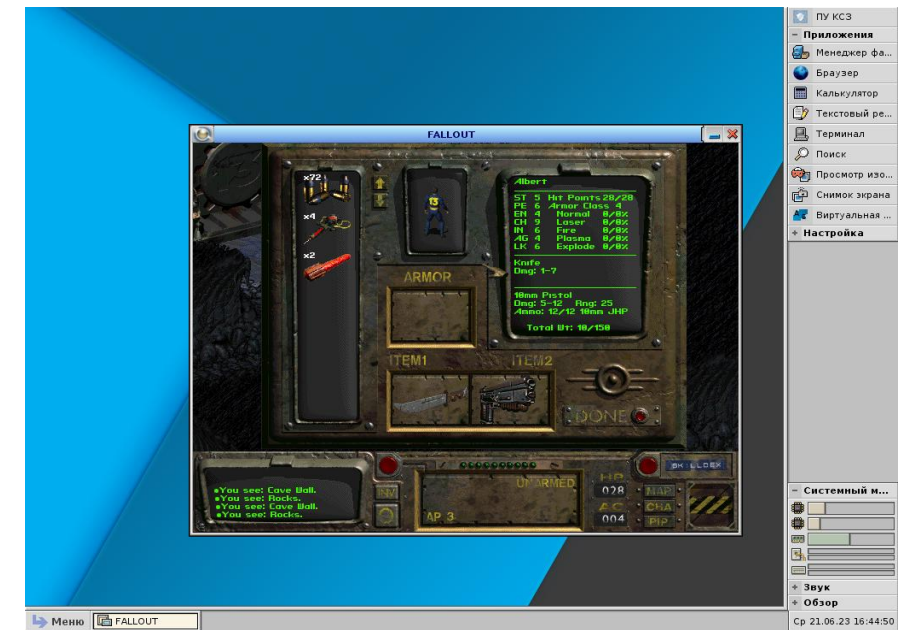
```
$ cd libpng-code
```
- Добавляем сборочное окружение для «Нейтрино»

```
$ addvariant -n -v -i OS nto x86 o
```

```
$ addvariant -n -v -i OS nto arm le.v7
```
- Структура сборочных каталогов и файлов
 - common.mk — конфигурационный файл сборочной системы «Нейтрино»
 - Makefile — рекурсивный файл построения
 - nto/x86, nto/arm, nto/... — каталоги со сборкой для целевой архитектуры
- Собираем

```
$ cd ./nto
```

```
$ make && make install
```



Пример сборки проекта с открытым исходным кодом

- Конфигурационный файл `common.mk` для рекурсивной сборочной системы и сборки проекта с помощью **CMake**

```
1  ▼ ifndef QCONFIG
2  QCONFIG=qconfig.mk
3  endif
4  include $(QCONFIG)
5
6  CMAKEFLAGS += -DPNG_STATIC=OFF -DPNG_TESTS=ON
7
8  USEFILE=
9  include $(MKFILES_ROOT)/cmake.mk
10 include $(MKFILES_ROOT)/qttargets.mk
11 |
```



Пример сборки проекта с открытым исходным кодом

- Альтернативный конфигурационный файл `common.mk` для использования **GNU Autotools** и **configure**
 - пример задания опции для конкретной процессорной архитектуры
 - сборка командами `make && make install` (без вызова `configure`)

```
1  ▼ ifndef QCONFIG
2  QCONFIG=qconfig.mk
3  endif
4  include $(QCONFIG)
5
6  CONFIGUREFLAGS += -C
7  CONFIGUREFLAGS += -prefix=/usr
8  CONFIGUREFLAGS += --disable-static
9
10 CONFIGUREFLAGS_armle-v7 += --enable-arm-neon=api
11
12 CONFIGUREFLAGS += $(CONFIGUREFLAGS_$(CPUVARDIR))
13
14 USEFILE=
15 include $(MKFILES_ROOT)/autotools.mk
16 include $(MKFILES_ROOT)/qtargets.mk
17
```



Портирование

- Для портирования можно использовать ОС-специфичные макросы
 - определены сборочной системой: `__KPPDA__`, `__KPPDANTO__`
 - дополнительные макросы подключаются из `<sys/platform.h>`
 - версия ОС `_KPPDA_VERSION`
 - endian: `__BIGENDIAN__`
- Примеры кода

```
11
12 ▼ #ifndef __KPPDANTO__
13     #include <linux/ctype.h>
14     #else
15     #include <common.h>
16     #endif
17
```

```
95
96 ▼ #ifdef __KPPDANTO__
97     »     »     »     delay(write_delay);
98     #else
99     »     »     »     usleep(write_delay*1000);
100    #endif
101
```



Особенности сборки приложений для нескольких процессорных архитектур

- Потенциальные проблемы с open-source
 - не всегда учтена разная размерность типов для 64x-32x разрядных систем
 - Big Endian/Little Endian
 - выравнивание данных и обращение к невыровненным данным
 - кэширование



- `gdb` для прикладных приложений
 - сервис для удалённого подключения к целевой системе (**`pdebug`**)
- Утилита **`dumper`** для создания дампов при аварийном завершении процессов
- Системные приложения/драйверы/системный загрузчик
 - отладочная информация для аппаратных **JTAG**
 - упрощение отладки драйверов за счёт микроядерной архитектуры



Использование статического анализатора svace (ИСП РАН)

- Использование svace совместно с тулчейном «Нейтрино»

- Сборка проекта

```
$ cd ./project
```

```
$ svace build make -j$(nproc) all
```

- Создание отчёта

```
$ svace remote ...
```

```
$ svace import
```

```
$ svace upload
```

The screenshot displays the svace web interface. On the left, a sidebar shows a summary of detected issues: CRITICAL (0), MAJOR (2), NORMAL (2), MINOR (0), and UNDEFINED (0). The main area shows a table of detected issues with columns for ID, Detector, File, Line, and Description. Below the table, there is a comparison section titled 'Информация о различиях' (Information about differences) comparing two snapshots: 'Текущий (V1)' (Current V1) and 'Целевой (V2)' (Target V2). The comparison table shows details like ID, Name, Import Time, Created By, Host Name, Analysis Parameters, Statistics, and Attached Files.

ID	Детектор	Файл	Строка	Описание
1	UNCHECKED_FUNC_RES_STAT	vfs.c	909	Return value of function 'MsgSendPulse', called at vfs.c:909, is ...
2	UNCHECKED_FUNC_RES_STAT	sim_ata.c	1929	Return value of function 'eide_command', called at sim_ata.c:1...
3	UNCHECKED_FUNC_RES_STAT	sim_eide.c	1820	Return value of function 'eide_reg_wait', called at sim_eide.c:1...
4	UNCHECKED_FUNC_RES_STAT	xpt.c	567	Return value of function 'pthread_setname_np', called at xpt.c:...
5	UNINIT_LOCAL_VAR	sim_eide.c	1900	Uninitialized data is read from local variable 'tgt' at sim_eide.c...
6	UNINIT_LOCAL_VAR	sim_eide.c	1633	Uninitialized data is read from local variable 'chip' at sim_eide.c...
7	UNREACHABLE_CODE	vnodeops.c	1019	This statement in the source code might be unreachable duri...
8	UNREACHABLE_CODE_RET	sim_eide.c	2091	This statement in the source code might be unreachable duri...

Текущий (V1)		Целевой (V2)	
ID	Имя	ID	Имя
	Snapshot 2023-06-16 15:31:58 +0300		Snapshot 2023-06-16 12:57:17 +0300
Время импорта	16.06.2023 15:31	Время импорта	16.06.2023 12:57
Создано	importer	Создано	importer
Имя хоста	[kpda-neutrino/trunk]	Имя хоста	[kpda-neutrino/trunk]
Параметры анализа	Показать различия	Параметры анализа	Показать различия
Статистика	Различия разметок	Статистика	Различия разметок
Приложенные файлы	Показать	Приложенные файлы	Показать



Полезные примеры и документация

- Публичный git репозиторий с примерами исходных кодов драйверов и утилит для «Нейтрино»

<https://git.kpda.ru/explore/repos>

- Примеры проектов с открытым исходным кодом

[Midnight Commander](#) (зависимости: [S-Lang](#), [PCRE](#))

[smartmontools](#) — диагностика дисков S.M.A.R.T.

[nano](#) — текстовый редактор

- [Русскоязычная документация](#)

[Комплект разработчика](#)

[Рекурсивная сборочная система](#)

[Qt Creator](#)



- Использование привычного инструментария для разработки ПО для встраиваемых систем реального времени
- Единая среда разработки для всех поддерживаемых процессорных архитектур
- Удобство переноса проектов с открытым исходным кодом



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

