

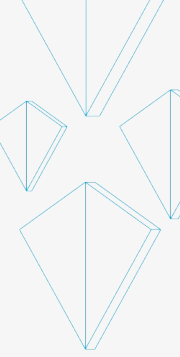
Инструментальные средства для ЗОСРВ «Нейтрино»: текущее состояние и планы развития

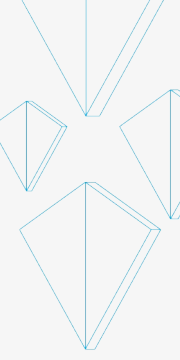
Игорь Рондарев

Руководитель отдела инструментальных средств

Инструментарий с точки зрения...

- ◆ ... «программиста» (разработчика ПО)
- ◆ ... «администратора» (разработчика ПО / сисадмина / DevOps)
- ◆ ... «менеджера» (специалиста по договорам, закупкам и т. д.)





... с точки зрения «программиста»

- ◆ **Поддерживаемые языки, стандарты и платформы («фреймворки»)**
- ◆ **Доступные инструменты (сборочные системы, графические средства разработки и т. д.)**

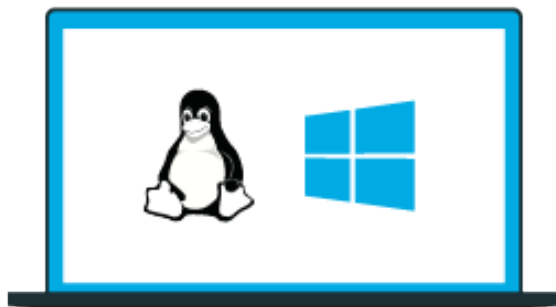
В сумме определяют возможности создания нового и портирования существующего ПО для выбранной операционной системы.

Инструментарий разработчика

- ◆ Разработка и отладка ПО выполняется на **«инструментальных»** системах, работающих под управлением ОС общего назначения (*GNU/Linux* или *MS Windows*) с последующим ручным или автоматизированным копированием результатов разработки (исполняемых файлов, библиотек и т.д.) на **«целевые»** системы под управлением ЗОСРВ «Нейтрино»
- ◆ Используется концепция т. н. **«кросс-компиляции» (cross compilation)**

Инструментальная система

(ПК с Windows/Linux)



Канал передачи данных
(сеть, последовательный порт)



Целевая система

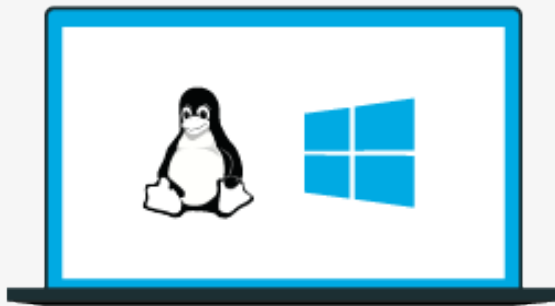
(ЗОСРВ «Нейтрино»)



Инструментарий разработчика

Все компоненты, о которых далее пойдёт речь, входят в единый автономный инструментарий (официальное название — «**Комплект разработчика для ЗОСРВ «Нейтрино»**»), который должен быть предварительно установлен **на инструментальную систему.**

Инструментальная система
(ПК с Windows/Linux)



Канал передачи данных
(сеть, последовательный порт)



Целевая система
(ЗОСРВ «Нейтрино»)



Инструментарий разработчика

- ◆ **Кросс-компиляторы** (на базе **GNU Compiler Collection: GCC, G++, GFortran**)
 - ◇ Актуальная версия: **gcc 8.3.0** (выпуск **2019 года**, поддержка стандартов до **C17 и C++17** включительно)
 - ◇ Также в состав КР входит **gcc 4.8.3** (основной компилятор для предыдущих версий ЗОСРВ «Нейтрино»)
 - ◇ Поддерживаемые *целевые* платформы: **x86, ARM, MIPS, PowerPC, ARMv8 (aarch64le)** и т.д.
 - ◇ Используются совместно с пакетом **binutils** версии **2.32** (ассемблеры, редакторы связей и т.д.)
- ◆ **Кросс-компилятор Icc** для платформы «Эльбрус» (**e2kle**)
- ◆ **Отладчики** (на базе **GDB**)
 - ◇ Используется концепция удалённой отладки (*remote debugging*)
- ◆ **Среда исполнения Python 3.9 + подсистема кросс-компиляции модулей расширения**
 - ◇ среда исполнения (python3) — на **целевых системах**, подсистема кросс-компиляции модулей расширения — на **инструментальных**.

Инструментарий разработчика

- ◆ **Графическая среда разработки** (на базе исходного кода **Qt Creator IDE**)
 - ◇ Интегрированная русскоязычная **документация** (в т.ч. описание программных интерфейсов ЗОСРВ «Нейтрино»)
- ◆ **Образ виртуальной машины ЗОСРВ «Нейтрино» (x86)**
 - ◇ Для использования в средах виртуализации (VirtualBox, VMware, QEMU), в т.ч. для начального знакомства с системой и решения задач быстрого прототипирования
- ◆ **Компоненты операционной системы ЗОСРВ «Нейтрино»** (библиотеки + заголовочные файлы, драйверы, утилиты и т. д.)
- ◆ **Вспомогательные утилиты**
 - ◇ Создание загрузочных образов (в т.ч. утилиты для создания образов виртуальных машин), анализ содержимого исполняемых файлов, просмотр журналов системных событий и т.д.

Сборочные системы

- ◆ **Сборочная система** — инструмент управления процессом преобразования исходного кода приложения в результирующий набор файлов (исполняемых, библиотек, файлов ресурсов, документации и т.д.)
- ◆ Задачи, решаемые сборочными системами
 - ◇ Управление процессом компиляции и связывания
 - ◇ Обеспечение доступа к зависимостям (напр., вспомогательным библиотекам)
 - ◇ Обеспечение возможности создания нескольких разновидностей исполняемых файлов, в т.ч. предназначенных для отладки («Release», «Debug» и т.д.), а также управления режимами оптимизации
 - ◇ Обеспечение переносимости разрабатываемого ПО (кроссплатформенность)
 - ◇ и т. д.
- ◆ Примеры сборочных систем: **Make-файлы** (GNU Make), **Autotools**, **CMake** (генератор), **Meson**, **Bazel**, **SCons** и т.д.

Сборочные системы

◆ Непосредственный (явный) вызов кросс-компилятора

- ◆ Тулчейны **nto<платформа>-<компонент>[-<версия>]**

- ◆ Пример:

- ◆ **ntoaarch64-g++-8.3.0 -std=c++17 -o test test.cpp**

- или

- ◆ **make CC=ntox86-g++-8.3.0 CFLAGS=-O3 test**

◆ Рекурсивная сборочная система («Makefile + common.mk + ...»)

- ◆ Используется инструмент **GNU Make + набор расширений**

- ◆ Основное назначение - **сборка унаследованных проектов**, разработанных для предыдущих редакций ЗОСРВ «Нейтрино», а также перенос проектов, разработанных для QNX RTOS (в т.ч. созданных в среде **Photon Application Builder**)

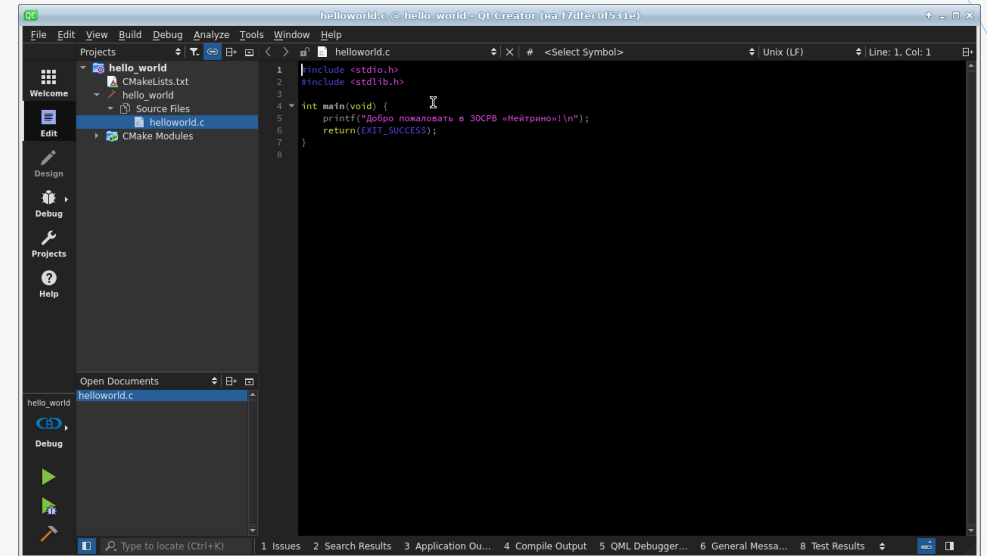
- ◆ Модульная поддержка внешних сборочных систем (модули **smake.mk, meson.mk, autotools.mk**, входящие в состав КР)

Сборочные системы

- ◆ **Кросс-компиляция с помощью CMake и Meson** (режим командной строки)
 - ◇ В состав КР входит соответствующий набор т.н. **toolchain-файлов** (конфигурационных файлов, обеспечивающих поддержку кросс-компиляции под выбранную платформу)
 - ◇ Пример:
 - ◇ \$ mkdir build && cd build
 - ◇ \$ export **CMAKE_TOOLCHAIN_FILE=\$KPDA_HOST/mk/cmake/toolchain-nto-armle-v7.cmake**
 - ◇ \$ cmake .. && cmake --build
- ◆ **Использование графической среды разработки** (Qt Creator редакции КПДА)
 - ◇ Нативная поддержка сборочных систем **CMake, qmake** (дополнительно — **Meson** и **Autotools**)
 - ◇ Базовая поддержка рекурсивной сборочной системы

Графическая среда разработки

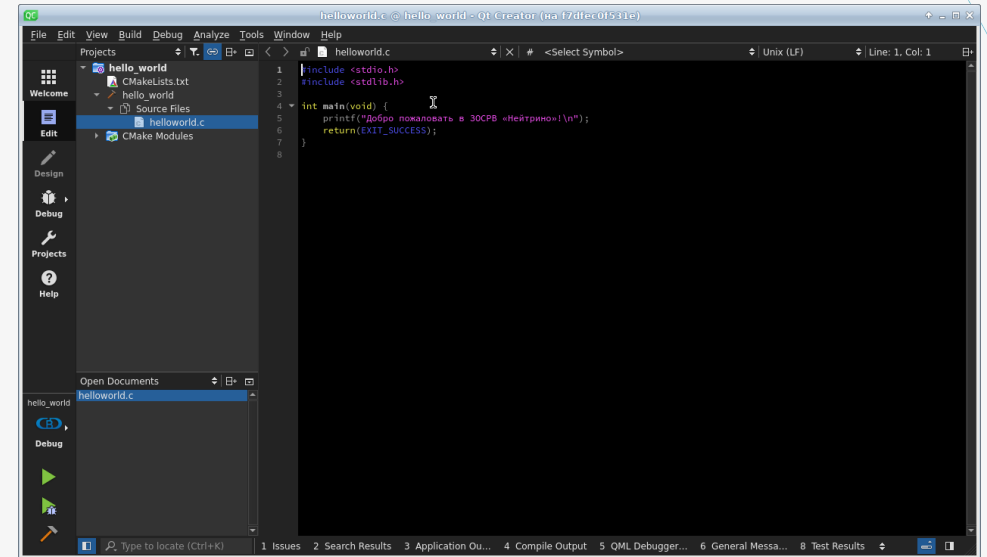
- ◆ Предназначена для разработки различных приложений, в т.ч. имеющих командно-строковый интерфейс и *не использующих библиотеки Qt* (напр., **утилиты, драйверы, менеджеры ресурсов** и т. д.)
- ◆ Представляет собой унифицированное окружение, предназначенное для работы над проектами (потенциально) любой сложности
- ◆ Использует штатные средства сборки и отладки проектов (компиляторы, отладчики), входящие в состав КР
- ◆ Имеет встроенную поддержку современных сборочных систем
- ◆ Имеет встроенные средства разработки графических интерфейсов (**Qt Widgets, Qt Quick/QML**)
- ◆ Учитывает особенности процесса разработки приложений для ЗОСРВ «Нейтрино»



Графическая среда разработки

◆ Встроенные механизмы, ориентированные на процесс разработки приложений для ЗОСРВ «Нейтрино»

- ◆ Предварительно настроенный инструментарий (**компиляторы, Комплекты Qt** и т. д.)
- ◆ Примеры (шаблоны) приложений
- ◆ Средства взаимодействия с целевой системой для автоматизированного запуска и удалённой отладки приложений
- ◆ Документация на среду разработки (см. меню «**Справка**» → «**Содержание**»)
 - ◆ Статьи в составе документации на КР («**Начало работы**», «**Вопросы и ответы**» и т. д.), а также общая англоязычная документация («**Qt Creator Manual**»)



Графическая среда разработки

◆ Расширение с помощью модулей

◆ Пример: модуль поддержки статического анализатора «PVS-Studio»

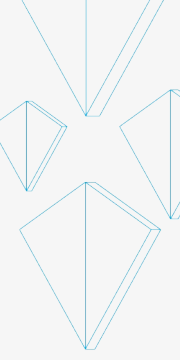
- ◆ Версия модуля для Комплекта разработчика создана специалистами ООО «ПВС» в сотрудничестве с ООО «СВД ВС»

The screenshot displays the Qt Creator IDE interface. On the left, a file explorer shows a project structure with files like `json_c_version.c`, `json_object.c`, and `json_pointer.c`. The main editor shows C++ code with a `goto set_single_path;` statement. Below the editor, a 'Problems' panel shows a list of errors with severity levels (High, Medium, Low) and counts (0, 7, 42). A detailed error message window is open, titled 'Подробнее о модуле PVS-Studio — Qt Creator'. The window contains the following information:

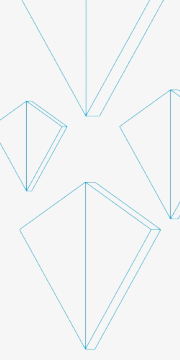
- Название: PVS-Studio
- Версия: 7.32.84714_1
- Совместимая версия: 6.0.2
- Поставщик: PVS-Studio LLC
- Группа: Code Analyzer
- URL: <https://pvs-studio.com>
- Размещение: /tmp/harbor_int_kpda_ru_kpda_neutrino_trunk.o2lik/.local/share/data/QtProject/qtcreator/plugins/6.0.2/libpvs-studio-qtcreator-6.0.x-qt5-kpda-7.32.84714.1.so
- Платформы: Все (текущая: «Linux (Ubuntu 20.04.6 LTS)»)
- Описание: PVS-Studio plugin for Qt Creator
- Авторское право: (C) 2023 PVS-Studio LLC
- Лицензия: Commercial
- Зависимости: Core (6.0.2), CppEditor (6.0.2), ProjectExplorer (6.0.2), TextEditor (6.0.2)

Графическая среда разработки

- ◆ Существует возможность интеграции инструментария, входящего в состав Комплекта разработчика, с внешними графическими средами разработки (**VS Code, CLion, сторонние сборки Qt Creator IDE** и т. д.)
 - ◆ интеграция осуществляется на уровне т. н. *тулчейнов* (совместимость с интерфейсами GCC и GDB)



TraceKey — инструмент анализа трасс



- ◆ **Специализированный инструмент**, позволяющий упростить анализ системных трасс, собранных с узлов под управлением ЗОСР В «Нейтрино»
 - ◇ Используется технология т. н. «системного профилирования»
 - ◇ Разработка инструмента *TraceKey* началась в 2023 году
- ◆ **Типовые задачи:**
 - ◇ Анализ характеристик реального времени приложений
 - ◇ Исследование процессов взаимодействия между компонентами программных комплексов
 - ◇ Поиск причин аномального поведения (пропущенные прерывания, нарушения в циклограммах и т.д.)

TraceKey — инструмент анализа трасс

Визуальное отображение событий на панели «Timeline»

The screenshot displays the TraceKey application interface. The top window title is `/tmp/proj4/tracebuffer_no_opt.kev`. The left sidebar shows a tree view with the following structure:

- tracebuffer_no_opt.kev
 - Key Header
 - Статистика
 - Timeline
 - Filters
 - Frame Filter
 - Owner Filter
 - Frame Search

The main area is the Timeline, showing a list of threads and processes with their corresponding event bars. The threads listed include:

- Thread 1 (PID 49159)
- undef name (PID 49160)
- proc/boot/devc-pty (PID 65542)
- proc/boot/pipe (PID 65543)
- proc/boot/mqueue (PID 65544)
- proc/boot/random (PID 65545)
- proc/boot/dumper (PID 65546)
- proc/boot/io-hid (PID 65547)
- proc/boot/io-pkt-v4 (PID 65548)
- io-pkt main (PID 65548)
- io-pkt#0x00 (PID 65548)

The timeline visualization shows horizontal bars for each thread, with colors indicating different states or events. A tooltip is visible over a specific event in the `io-pkt#0x00` thread, showing the following details:

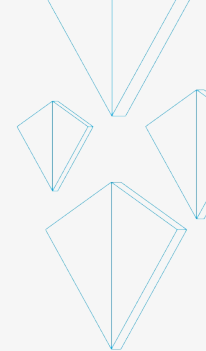
```
EventClass = Thread
EventType = ThreadReceive
Owner = 87
Timestamp = 30232417875
CpuId = 0
pid = 65548
tid = 2
```

The bottom table displays the event details:

Метка времени	CPU_ID	Owner	Тип	Параметры
01:54:42.857 640 689	0	proc/boot/io-pkt-v4 → io-pkt#0x00	thread_running	
01:54:42.857 640 934	0	proc/boot/io-pkt-v4 → io-pkt#0x00	ker_exit_msg_send_v	status = 0; rmsg0 = 0x0
01:54:42.857 641 402	0	proc/boot/io-pkt-v4 → io-pkt#0x00	ker_enter_connect_detach	coid = 0x5
01:54:42.857 641 780	0	proc/boot/io-pkt-v4 → io-pkt#0x00	ker_exit_connect_detach	retVal = 0x0
01:54:42.857 678 820	0	proc/boot/io-pkt-v4 → io-pkt#0x00	ker_enter_msg_send_v	coid = 0x4; msg = 0x200113
01:54:42.857 679 266	0	proc/boot/io-pkt-v4 → io-pkt#0x00	communication send message	rcvid = 0x10005; pid = 3

The bottom left corner shows the current time: `01:54:42.857 651 650`. The bottom right corner has an `Absolute` dropdown menu.

TraceKev — инструмент анализа трасс



◆ Основные функции

- ◆ Визуальное отображение информации о системных событиях, содержащейся в файлах **.kev** (Kernel Events)
- ◆ Группировка, масштабирование, поиск и комплексная фильтрация отображаемой информации
- ◆ Базовая аналитика (временные интервалы, взаимосвязь между событиями)

◆ Основные характеристики

- ◆ Программное ядро и интерфейс собственной разработки (**C++**, **Boost**, **Qt**)
- ◆ Высокая производительность операций визуализации и поиска/фильтрации
 - ◆ Возможность обработки больших системных трасс (**сотни МБ**)

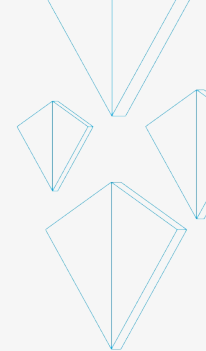
Планы развития

◆ Анализатор системных трасс **TraceKey**

- ◆ Выпуск публичной версии в составе текущей редакции *Комплекта разработчика*
 - ◆ Ориентировочные сроки — **4-й квартал 2024 года**

◆ Дальнейшее расширение функциональности

- ◆ Функции удалённого взаимодействия с целевыми системами (сбор системных трасс с заданными характеристиками по запросу оператора)
- ◆ Дополнительные инструменты сбора статистики
- ◆ Средства настройки пользовательского интерфейса
- ◆ и т.д.



Планы развития

◆ Базовый инструментарий (компиляторы, отладчики)

- ◆ Выпуск новых версий (в т.ч. для поддержки актуальных целевых платформ)
- ◆ Переход на стандартные протоколы удалённой отладки (напр. **gdbserver**)
- ◆ Повышение уровня интеграции со сторонними графическими средами разработки

◆ Расширенный инструментарий

- ◆ Инструменты динамического анализа и «фаззеры» (*в процессе портирования / экспериментального использования*)
 - ◆ **Valgrind** (обновление до актуальной версии) + **Taintgrind** (анализ утечек данных и поиск уязвимостей)
 - ◆ **AFL++** (мутационный фаззер)

Планы развития

◆ Графическая среда разработки

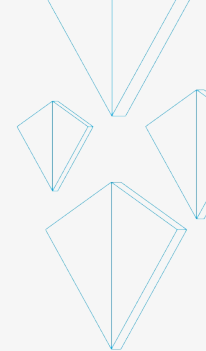
- ◆ Дальнейший выпуск новых версий на базе Qt Creator (ориентир на полноценное использование возможностей базовой платформы с их **адаптацией под особенности разработки для ЗОСРВ «Нейтрино»**)

◆ Компоненты операционной системы «Нейтрино» в составе КР

- ◆ Предоставление возможности использовать встроенные механизмы сборочных систем (напр. **find_package()** в CMake)

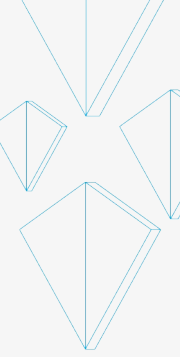
◆ Дополнительная документация

- ◆ Разработка методики перевода рекурсивных проектов на современные сборочные системы



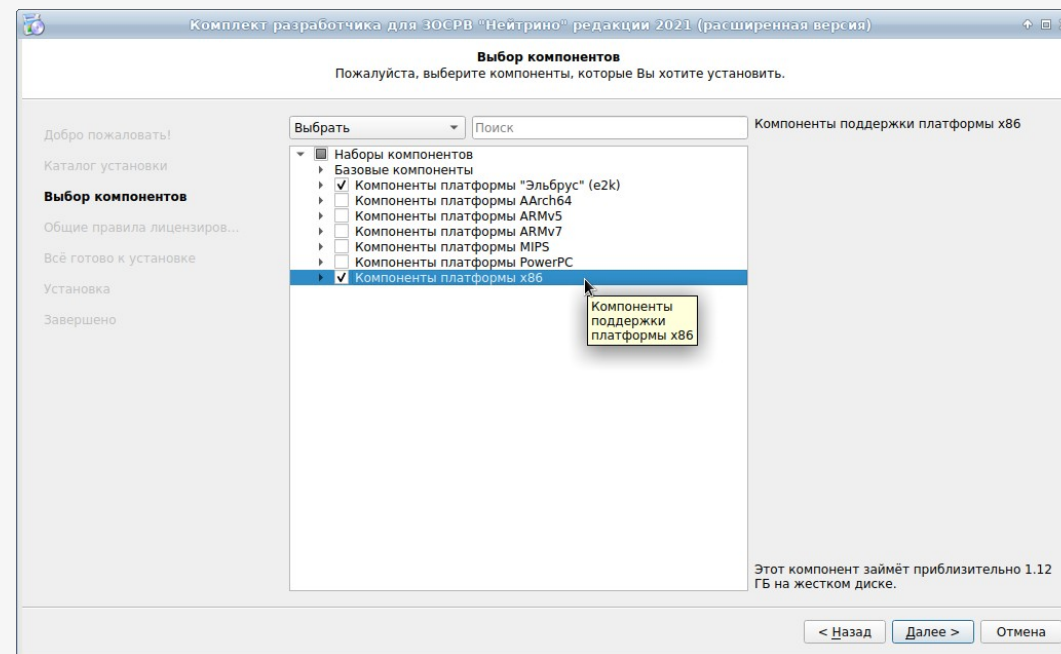
... с точки зрения «администратора»

- ◆ **Установка и настройка инструментария**
- ◆ **Установка обновлений и экспериментальных компонентов**



Начальная установка

- ◆ Комплект разработчика поставляется в виде **ISO**-образа:
 - ◇ Пример: **kpda_neutrino_2021_devkit_extended_20240905_b769.iso**
 - ◇ Официальная поставка на материальном носителе (диск формата DVD)
 - ◇ Возможность скачивания актуальной версии с сайта **dl.kpda.ru** (в рамках действующей подписки на обновления)
- ◆ Образ включает в себя инсталляторы Комплекта разработчика для поддерживаемых операционных систем
- ◆ Не требует установки каких-либо сторонних компонентов (является **автономным**)



Поддерживаемые системы

- ◆ Дистрибутивы **GNU/Linux** (Debian, Ubuntu, Альт, Astra Linux и т. д.)
- ◆ ОС семейства **Microsoft Windows 10** и более поздние
 - ◇ КР до редакции 2021 включительно: исполняемые файлы для Windows (**PE64**)
 - ◇ КР редакции 2024 и далее: **Windows Subsystem for Linux (WSL2)**
- ◆ Установка по умолчанию:
 - ◇ Для **GNU/Linux**: каталог **/opt/kpda<редакция>**
(напр. **/opt/kpda2021**)
 - ◇ Для **Windows**: каталог **C:\KPDA<редакция>**
(напр. **C:\KPDA2021**)
- ◆ Возможна **автоматизированная (неинтерактивная)** установка, в т.ч. для решения задачи установки обновлений, а также создания сборочных окружений (**Docker**)

Версионирование

- ◆ Номер редакции КР соответствует **редакции ЗОСРВ «Нейтрино»**, для которой данный КР предназначен (**2020, 2021** и т. д.) и **не меняется** в ходе её жизненного цикла
- ◆ Возможна установка обновлений, затрагивающих компоненты инструментария и самой операционной системы «Нейтрино»
- ◆ Информация о выпущенных **обновлениях КР** (исправления ошибок в инструментальных компонентах, добавление новых функций и т. д.) доступна в **журналах изменений КР** (т. н. *ChangeLog*), а также косвенно отражена в дате и номере сборки КР
 - ◆ Ревизия и дата выпуска комплекта разработчика указана в файле **devkit_id.txt**, находящемся в каталоге установки (необходимы при обращении в службу *Технической поддержки*)
 - ◆ Пример: `kpda-neutrino 2021 development kit (linux64 generic 20240902_b768)`

Установка обновлений

- ◆ КР редакций **до 2020 включительно** — **монолитный инсталлятор** (обновление через полную переустановку КР; частичная установка или установка «поверх» не поддерживается)
- ◆ КР редакции **2021 (актуальная версия)** — **модульный инсталлятор**
 - ◇ Возможность выборочной установки/удаления компонентов для поддерживаемых платформ (x86, ARM и т.д.)
 - ◇ Возможность установки обновлений без полной переустановки КР (в т.ч. используя средства сетевой установки)

Планы развития

- ◆ КР редакции 2024 (в процессе разработки) — **модульный пакетный инсталлятор**
 - ◇ Переход на использование полноценной **системы управления пакетами** для организации процессов установки, обновления и контроля состояния как инструментальных компонентов (компиляторов, графической среды разработки и т. д.), так и компонентов ЗОСРВ «Нейтрино» в составе КР
 - ◇ Основана на исходном коде менеджера пакетов **«расман»** (проект Arch Linux)

```
$ ntox86-pacman -Q
...
devu+uhci-runtime 1.0-1103
devu+xhci-runtime 1.0-1103
dhcp-dev 4.1.ESV.R16-1103
dhcp-runtime 4.1.ESV.R16-1103
dhcp.client-runtime 1.0-1103
diffutils-runtime 3.8-1103
diskutils-runtime 1.0-1103
dispcnf-runtime 1.0-1103
disputils-runtime 1.0-1103
dumper-runtime 1.0-1103
elvis-runtime 2.2.1-1103
...
$ kpdahost-pacman -Q
...
bindres-runtime 1.0-1
ldrel-runtime 1.0-1
mkxfs-runtime 1.0-1
qt5-runtime 5.15.5-1
qt6-runtime 6.2.7-1
qtcreator-runtime 12.0.0-1
tracekev-runtime 0.99.0-1
...
```

Планы развития

◆ КР редакции 2024 (в процессе разработки)

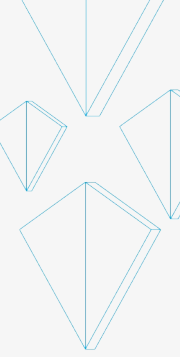
— модульный пакетный инсталлятор

- ◆ Преимущества пакетной системы
 - ◆ Управление зависимостями между пакетами
 - ◆ Контроль целостности установленных пакетов
 - ◆ Возможность установки обновлений и экспериментальных версий компонентов (в т.ч. *по сети*) с последующей корректной деинсталляцией
 - ◆ и т.д.

```
$ ntox86-pacman -Q
...
devu+uhci-runtime 1.0-1103
devu+xhci-runtime 1.0-1103
dhcp-dev 4.1.ESV.R16-1103
dhcp-runtime 4.1.ESV.R16-1103
dhcp.client-runtime 1.0-1103
diffutils-runtime 3.8-1103
diskutils-runtime 1.0-1103
dispcnf-runtime 1.0-1103
disputils-runtime 1.0-1103
dumper-runtime 1.0-1103
elvis-runtime 2.2.1-1103
...
$ kpdahost-pacman -Q
...
bindres-runtime 1.0-1
ldrel-runtime 1.0-1
mkxfs-runtime 1.0-1
qt5-runtime 5.15.5-1
qt6-runtime 6.2.7-1
qtcreator-runtime 12.0.0-1
tracekev-runtime 0.99.0-1
...
```

... с точки зрения «менеджера»

- ◆ **Вопросы приобретения и лицензирования**
- ◆ **Обучение сотрудников**



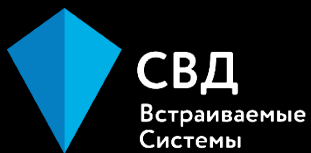
Варианты поставки

- ◆ **Базовая версия** (для коммерческого использования, включает поддержку основных вычислительных архитектур); коммерческое название: *Комплект разработчика для ЗОСРВ "Нейтрино"*
- ◆ **Расширенная версия** (для коммерческого использования, по сравнению с базовой версией дополнительно включает в себя поддержку архитектуры "Эльбрус"); коммерческое название: *Комплект разработчика для ЗОСРВ "Нейтрино" (с поддержкой архитектуры Эльбрус)*
- ◆ **Академическая версия** (для использования в образовательных учреждениях, включает поддержку только платформ x86 и ARMv7); коммерческое название: *Комплект разработчика для ЗОСРВ "Нейтрино" (академическая версия)*
- ◆ Подробная информация: **Отдел продаж ООО «СВД ВС»** (sales@kpda.ru)

Обучение сотрудников

- ◆ Обучение осуществляется в т.ч. **по следующим направлениям:**
 - ◇ разработка программного обеспечения реального времени;
 - ◇ установка, конфигурирование операционных систем реального времени;
 - ◇ разработка программного обеспечения ответственного назначения с учетом требований по функциональной безопасности.
- ◆ **Отдельный курс по инструментальным средствам** находится в процессе разработки
- ◆ Учебный центр реализует **академическую программу**, которая позволяет российским образовательным учреждениям:
 - ◇ получать наше программное обеспечение **для использования в учебном процессе**
 - ◇ бесплатно проходить **курсы повышения квалификации**

Подробная информация: **Учебный центр ООО «СВД ВС»** (education@kpda.ru)



Спасибо за внимание!

Игорь Рондарев

Руководитель отдела
инструментальных средств

ул. Кузнецовская, д. 19,

г. Санкт-Петербург

+7 (812) 346-89-56

www.kpda.ru

support@kpda.ru