СВД
Встраиваемые
Системы

Практические вопросы использования различных сборочных систем в комплекте разработчика для ЗОСРВ «Нейтрино»

Алексей Васюнин

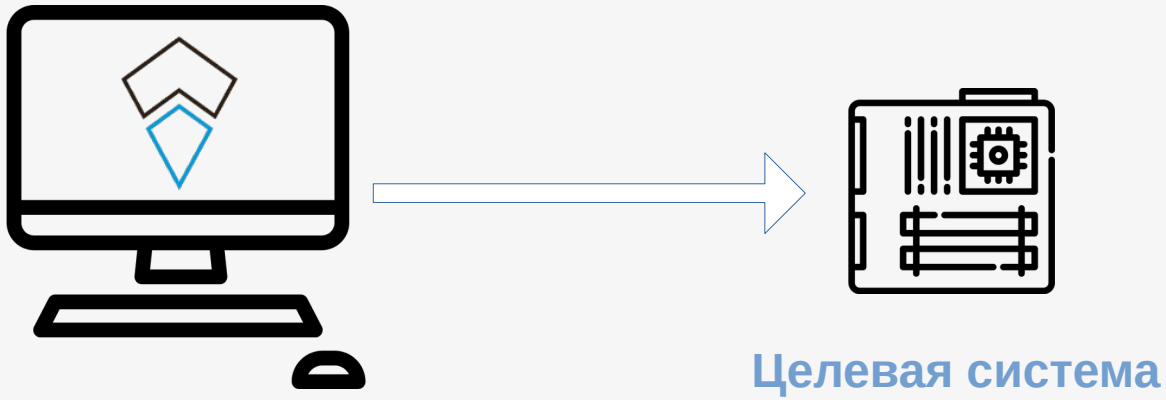
Руководитель отдела
Отдел автоматизации и верификации

Вступление

Комплект разработчика

Включен в Единый реестр российских программ для ЭВМ

Комплект разработчика (КР) для Нейтрино содержит инструментальные средства, необходимые для проведения разработки, отладки и тестирования прикладного и системного программного обеспечения, функционирующего в ОС Нейтрино.

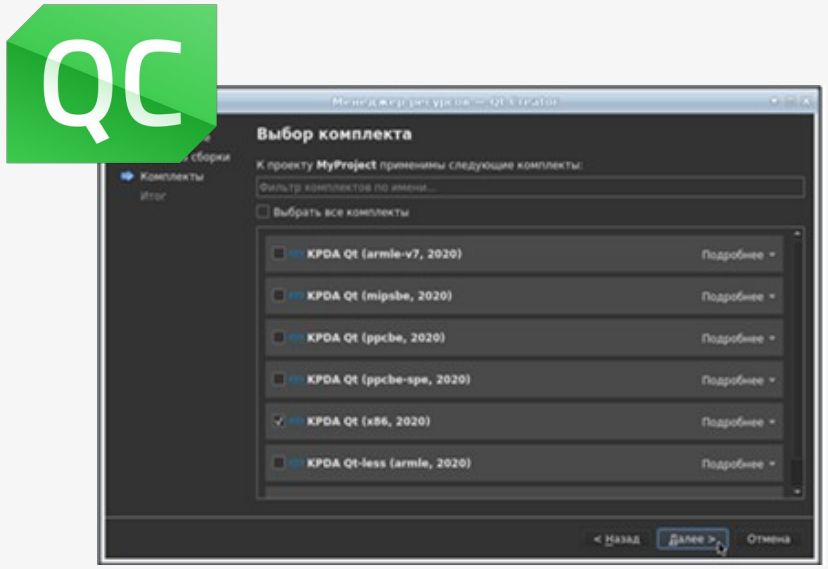


Инструментальная система

Целевая система

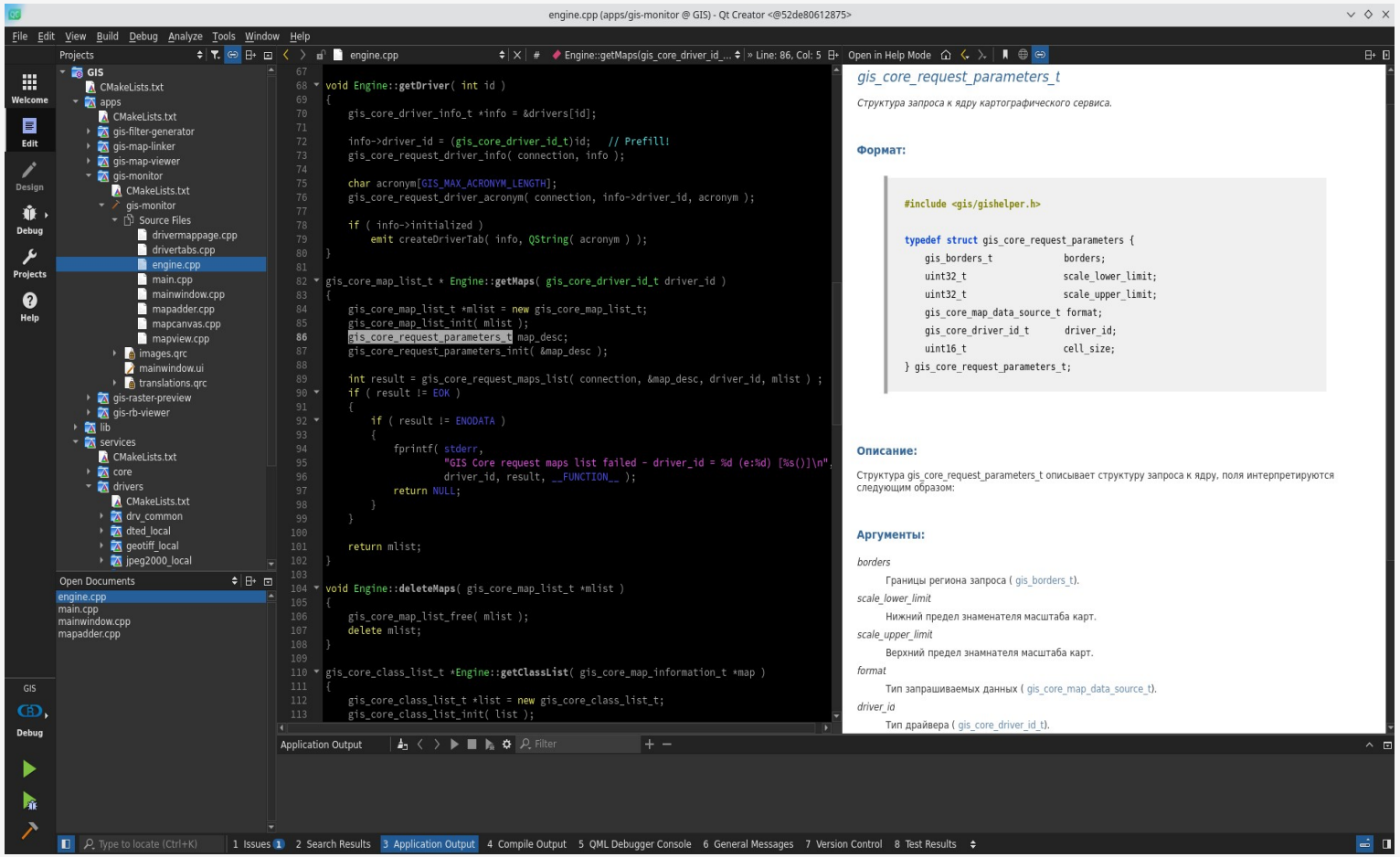
Работа в Qt Creator

Комплект разработчика



- Интеграция с тулчейнами в составе КР
- Сборка и развёртывание на ВМ “в один клик”
- Работа с отладчиком в том же окне
- Интегрированная справочная система
- Подсветка синтаксиса
- Навигация по коду

Интегрированная справочная система

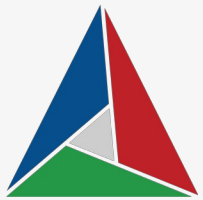


- Аналогична сайту help.kpda.ru
- Навести, нажать F1
- Не надо переключаться между окнами IDE и браузера
- Можно написать свою справку

Доступные сборочные системы

Сборочные системы

- **Рекурсивная сборочная система (GNU Make)**
- **CMake**
- **GNU Autotools**
- **qmake**
- **Meson**



Платформы

- x86
- ARMv5, ARMv7, ARMv8
- PowerPC (BE, BE-SPE)
- MIPS (LE, BE)
- Эльбрус

Рекурсивная сборочная система

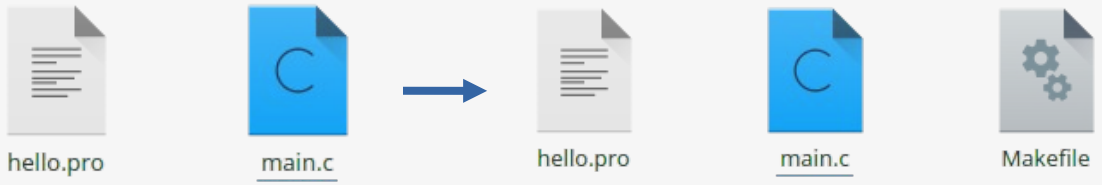

```
make install
```

```
CPULIST=x86 make install
```

```
ifndef QCONFIG
    QCONFIG=qconfig.mk
endif
include $(QCONFIG)

NAME=hello
USEFILE=
INSTALL_ROOT_nto=$(PROJECT_ROOT)/install

include $(MKFILES_ROOT)/qmacros.mk
include $(MKFILES_ROOT)/qtargets.mk
```

+ .mk файл для сторонней системы

qmake




```
TEMPLATE = app
TARGET = hello
SOURCES += main.c

helloworld.path = $$_PRO_FILE_PWD_/install
helloworld.files = hello
INSTALLS += helloworld
```

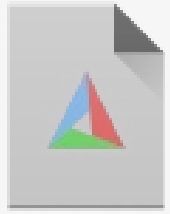
```
$KPDA_HOST/usr/bin/qmake/qmake-x86

make install
```

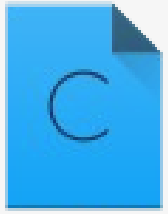
```
ifndef QCONFIG
    QCONFIG=qconfig.mk
endif
include $(QCONFIG)

NAME=hello
USEFILE=
INSTALL_ROOT_nto=$(PROJECT_ROOT)/install

include $(MKFILES_ROOT)/qmake.mk
include $(MKFILES_ROOT)/qtargets.mk
```

Для рекурсивной сборочной системы:

include $(MKFILES_ROOT)/qmake.mk

CMake


CMakeLists.txt


main.c

```
1  cmake_minimum_required( VERSION 3.23 FATAL_ERROR )
2
3  project( HelloWorld LANGUAGES C )
4
5  add_executable( HelloWorld main.c )
6
7  install( TARGETS HelloWorld DESTINATION bin )
```

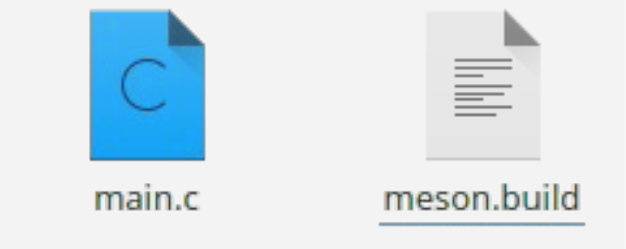
```
cmake --toolchain=$KPDA_HOST/mk/cmake/toolchain-nto-x86.cmake .
cmake --build   .
cmake --install . --prefix /path/to/install
```

```
ifndef QCONFIG
    QCONFIG=qconfig.mk
endif
include $(QCONFIG)

NAME=hello
USEFILE=
INSTALL_ROOT_nto=$(PROJECT_ROOT)/install

include $(MKFILES_ROOT)/cmake.mk
include $(MKFILES_ROOT)/qtargets.mk
```

Для рекурсивной сборочной системы:

include $(MKFILES_ROOT)/cmake.mk

Meson




```
project( 'hello', 'c' )

executable( 'hello', 'main.c', install:true )
```

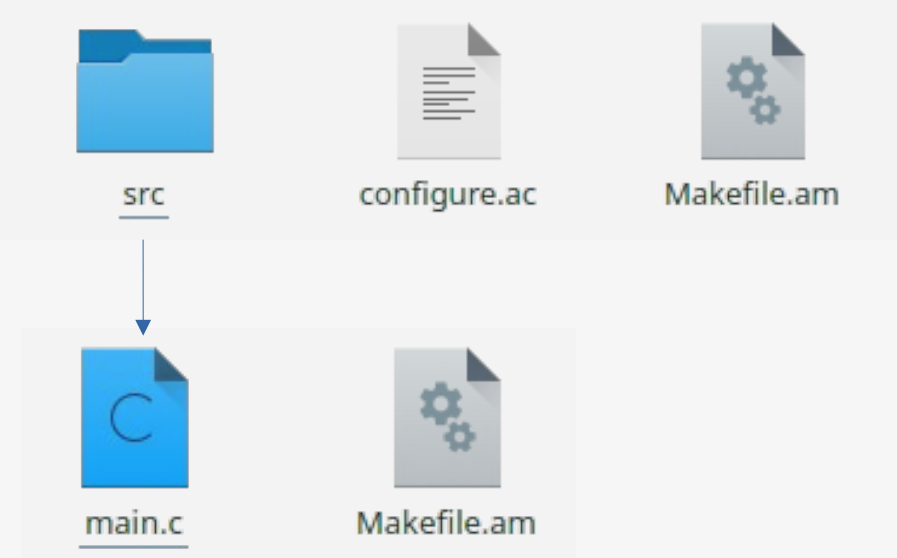
```
meson setup --cross_file=$KPDA_HOST/mk/meson/toolchain-nto-common.meson
            --cross_file=$KPDA_HOST/mk/meson/toolchain-nto-x86.meson   builddir
meson compile -C builddir
meson install -C builddir
```

```
ifndef QCONFIG
    QCONFIG=qconfig.mk
endif
include $(QCONFIG)

NAME=hello
USEFILE=
INSTALL_ROOT_nto=$(PROJECT_ROOT)/install

include $(MKFILES_ROOT)/meson.mk
include $(MKFILES_ROOT)/qtargets.mk
```

Для рекурсивной сборочной системы:

include $(MKFILES_ROOT)/meson.mk

Autotools




```
autoreconf --install
```

ИЛИ

```
aclocal
autoconf
automake --add-missing
```

ЗАТЕМ

```
./configure; make && make install
```

```
ifndef QCONFIG
    QCONFIG=qconfig.mk
endif
include $(QCONFIG)

NAME=hello
USEFILE=
INSTALL_ROOT_nto=$(PROJECT_ROOT)/install

include $(MKFILES_ROOT)/autotools.mk
include $(MKFILES_ROOT)/qtargets.mk
```

Для рекурсивной сборочной системы:

include $(MKFILES_ROOT)/autotools.mk

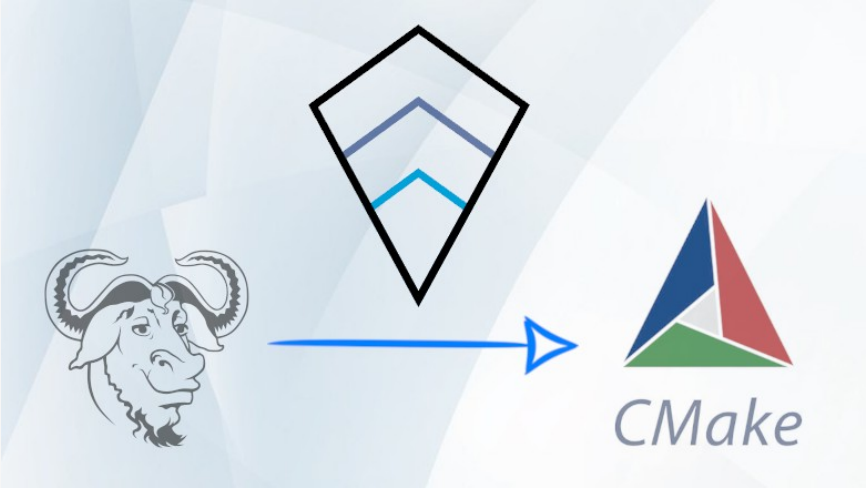
`make && make install` **– без вызова configure**

Сборка open source проектов

Возможность портирования в ОС Нейтрино интересующих Вас open source решений.

- __KPDA__
- __KPDANTO__
- __KPDAVERSION__
- __BIGENDIAN__
- <sys/platform.h>

Смена сборочной системы проекта



- **Автоматизация процессов**
- **Использование IDE**
- **Удобство синтаксиса**
- **Кросс-платформенность**
- **Единая система сборки для всего проекта**

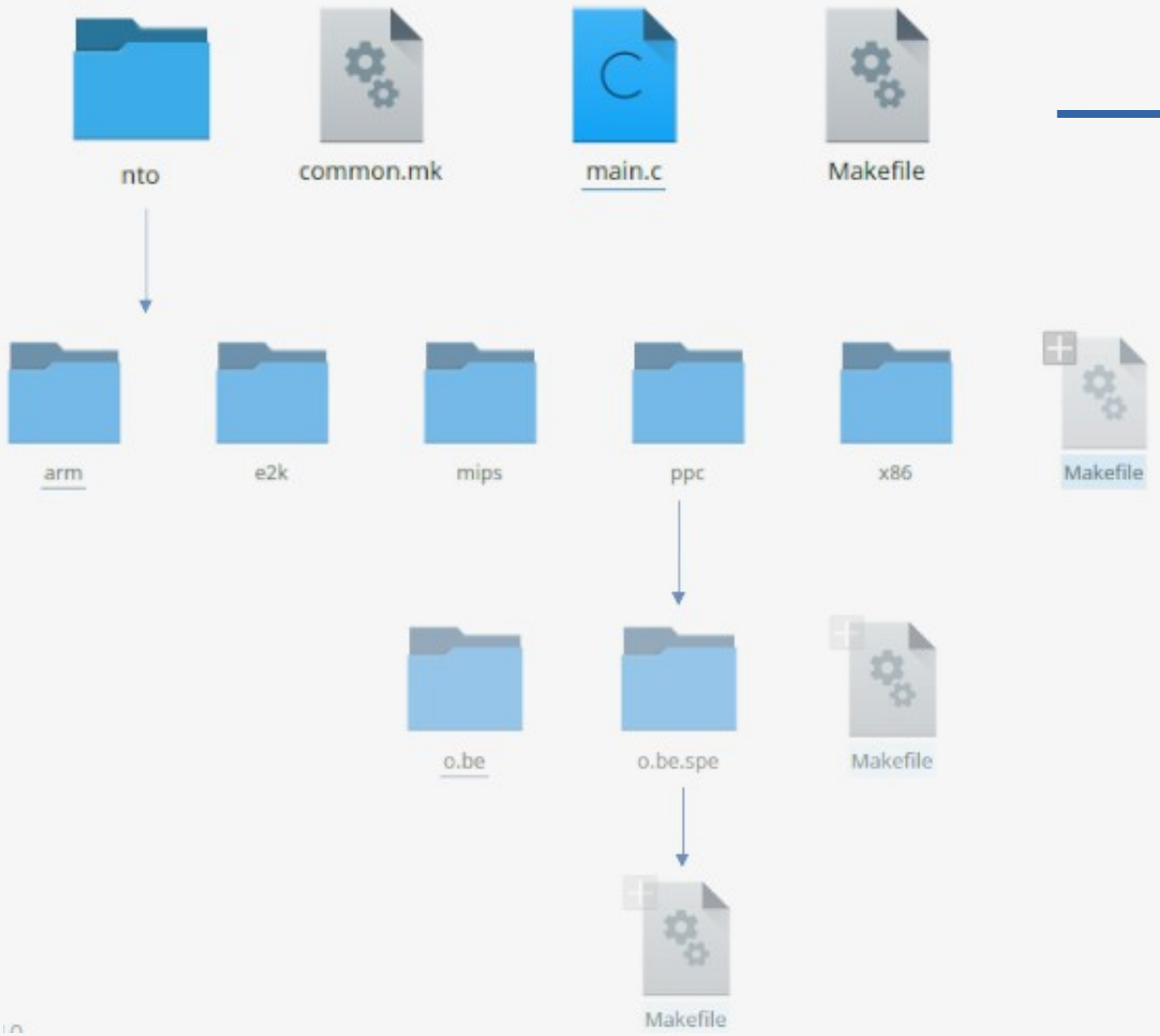
Смена сборочной системы проекта

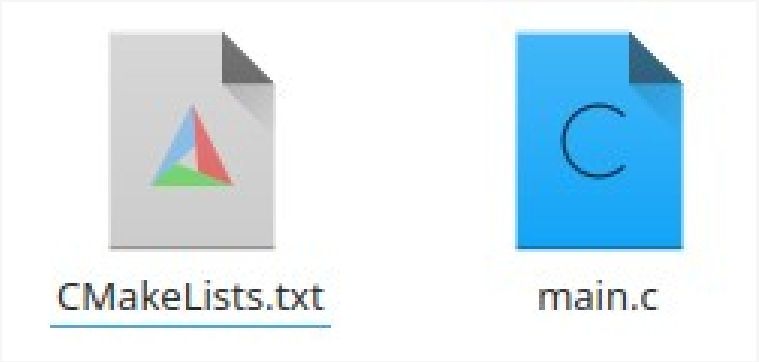
GNU Make

CMake

- **Время на переход на другую систему сборки не ограничено**
- **Возможность вернуться к стабильной сборке при помощи прошлой сборочной системы**
- **Необходимость поддержки обеих сборочных систем на время переходного периода**

Сокращение количества файлов





- Для сборки под шесть архитектур для <u>только одного компонента</u> было удалено:
  - 12 каталогов
  - 22 файла

Для всего проекта удалено несколько сотен файлов.

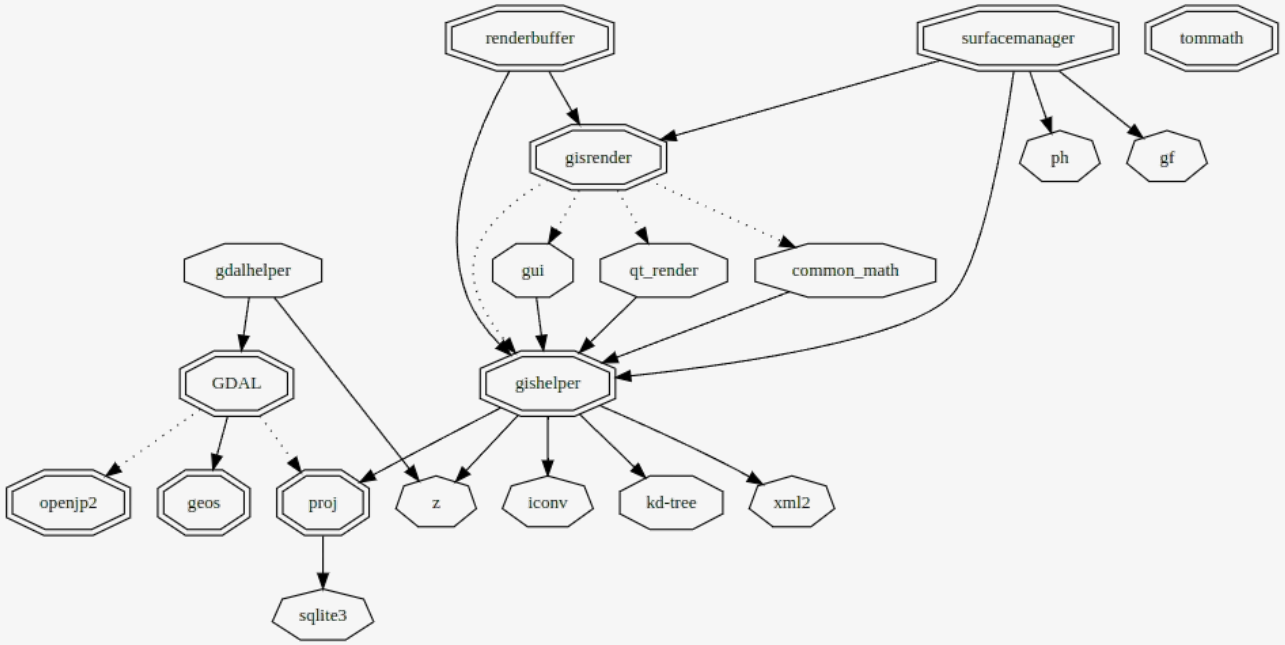
Наглядность сборки

```
[ 98%] Built target gis-map-linker
[ 98%] Automatic MOC and UIC for target gis-map-viewer
[ 98%] Built target gis-map-viewer_autogen
[ 98%] Building CXX object utils/gis-buffer-renderer/CMakeFiles/gis-buffer-renderer.dir/renderer.cpp.o
[ 98%] Building CXX object apps/gis-rb-viewer/CMakeFiles/gis-rb-viewer.dir/main.cpp.o
[ 98%] Generating qrc_translations.cpp
[ 98%] Generating qrc_images.cpp
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/gis-map-viewer_autogen/mocs_compilation.cpp.o
[ 98%] Linking CXX executable gis-monitor
[ 98%] Built target gis-monitor
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/distancemeter.cpp.o
[ 98%] Building CXX object apps/gis-rb-viewer/CMakeFiles/gis-rb-viewer.dir/rbwidget.cpp.o
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/heightscale.cpp.o
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/layerwindow.cpp.o
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/main.cpp.o
[ 98%] Linking CXX executable gis-buffer-renderer
[ 98%] Built target gis-buffer-renderer
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/mainwindow.cpp.o
[ 98%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/mapstylelayerwindow.cpp.o
[ 98%] Linking CXX executable gis-rb-viewer
[ 98%] Built target gis-rb-viewer
[100%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/mapwidget.cpp.o
[100%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/objectinfo.cpp.o
[100%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/objectsearch.cpp.o
[100%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/userdialog.cpp.o
[100%] Building CXX object apps/gis-map-viewer/CMakeFiles/gis-map-viewer.dir/viewparameters.cpp.o
```

```
Installing: /home/   /opt/gis/include/gis/gis_databuffer.h
Installing: /home/   /opt/gis/include/gis/gishelper.h
Installing: /home/   /opt/gis/include/gis/gis_math.h
Installing: /home/   /opt/gis/include/gis/gis_mdp.h
Installing: /home/   /opt/gis/include/gis/gis_objects.h
Installing: /home/   /opt/gis/include/gis/gis_path.h
Installing: /home/   /opt/gis/include/gis/gis_raw.h
Installing: /home/   /opt/gis/include/gis/gis_time.h
Installing: /home/   /opt/gis/include/gis/gis_types.h
Installing: /home/   /opt/gis/include/gis/gis_config.h
```

- **Лог для конкретной архитектуры**
- **Оценка времени сборки**
- **Без излишней информации**

Построение графа зависимостей



- **Анализ зависимостей модулей проекта**
- **Результат в виде графа в формате DOT**
- **Визуализация графа через утилиту dot или любой онлайн-визуализатор (удобно для внесения правок)**

```
cmake --graphviz=graph.dot .

dot -Tsvg graph.dot -o graph.svg
```

Высокоуровневый синтаксис

```
cmake_minimum_required( VERSION 3.23 FATAL_ERROR )

project( hello LANGUAGES CXX )

set( CMAKE_AUTOMOC ON )  # Обработка мета-объектным компилятором (moc)
set( CMAKE_AUTOUIC ON )  # Обработка файлов интерфейса (.ui)
set( CMAKE_AUTORCC ON )  # Обработка файлов ресурсов (.qrc)

add_executable( hello hello.c )

find_package( Qt5
              COMPONENTS
              Core
              REQUIRED )  # Находим установленные библиотеки Qt

target_link_libraries( hello
                       Qt5::Core )  # И линкуем их к приложению

install( TARGETS hello DESTINATION bin )
```

- **Лаконичность языковых конструкций**
- **Меньше требований к синтаксису**
- **Легче в освоении для новых сотрудников**
- **Знаком многим новым сотрудникам**

```
cmake_minimum_required( VERSION 3.23 FATAL_ERROR )

project( hello LANGUAGES C )

add_executable( hello hello.c )

install( TARGETS hello DESTINATION bin )
```

Пример: линковка статической библиотеки

Задача: Прилинковать статическую библиотеку `my_static_lib` к приложению `my_app`.

```
EXTRA_LIBVPATH += $(PROJECT_ROOT)/../my_static_lib/$(OS)/$(CPU)/a.shared$
(subst so,,$(COMPOUND_VARIANT))

LIBS += my_static_lib
```

```
target_link_libraries( my_app my_static_lib )
```

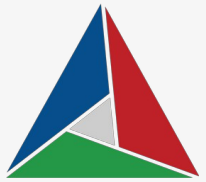
Автоматическое разрешение всех зависимостей.

Пример: линковка файла в библиотеку

Задача: Из множества .csv-файлов сформировать объектный файл и прилинковать его к библиотеке.

Читать файл при загруженной библиотеке можно при помощи макросов _binary_*_start[] и _binary_*_end[]

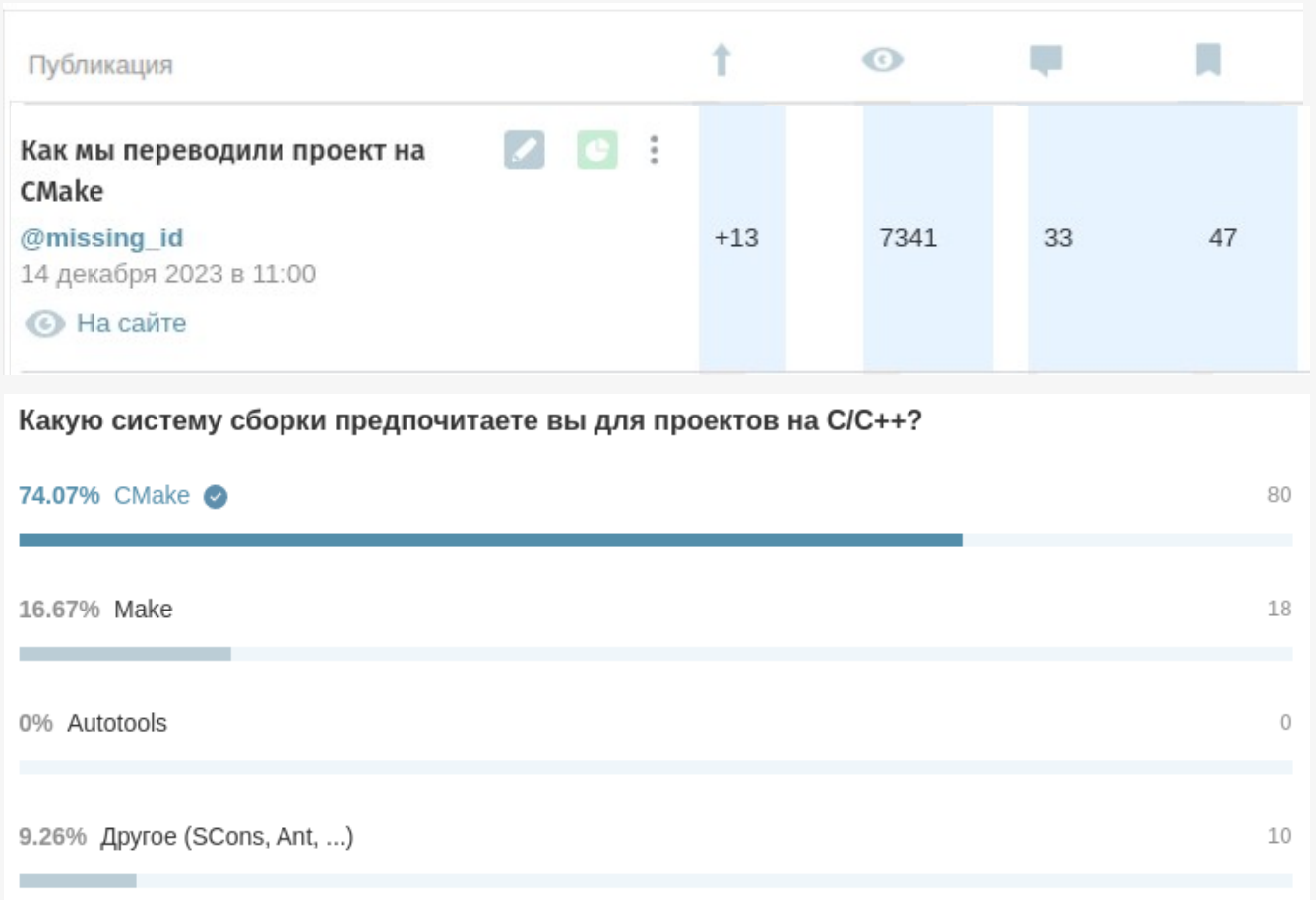
```
RESOURCES:=*.csv
EXTRA_OBJS=$(addsuffix .o,$(RESOURCES))
%.csv.o:
        cp $(subst .o,,$(PROJECT_ROOT)/palettes/$@) .
        $(LDPREF) -nostdlib -Wl,-r -Wl,-b -Wl,binary -o palettes.csv.o $(subst .o,,$@)
        rm $(subst .o,,$@)
```



```
add_custom_command( OUTPUT              ${DSTPATH}/palettes.csv.o
                    WORKING_DIRECTORY   ${SRCPATH}/palettes
                    COMMAND             ${LINKER} -nostdlib -r -b binary -o palettes.csv.o *.csv
                    COMMAND             ${CMAKE_COMMAND} -E copy palettes.csv.o ${DSTPATH}/palettes.csv.o
                    COMMAND             ${CMAKE_COMMAND} -E rm   palettes.csv.o )

add_custom_target( linking_extra_object_files
                   DEPENDS
                   ${CMAKE_BINARY_DIR}/lib/gishelper/palettes.csv.o )

add_dependencies( gishelper linking_extra_object_files )
```

Статья на Хабре

https://habr.com/ru/companies/swd_es/articles/773116



СВД
Встраиваемые
Системы

Спасибо за внимание!

Алексей Васюнин

Руководитель отдела
Отдел автоматизации и верификации

ул. Кузнецовская, д. 19,
г. Санкт-Петербург
+7 (812) 346-89-56
www.kpda.ru
support@kpda.ru