

УТВЕРЖДЕН  
07623974.37016-01 90 01-1-ЛУ

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «РАСЧЕТ-ЭМВ»**

**Руководство пользователя**

**07623974.37016-01 90 01-1**

Листов 62

| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

## АННОТАЦИЯ

В данном программном документе приведено руководство пользователя по применению и эксплуатации программного обеспечения ПО «Расчет-ЭМВ», используемого для расчетной оценки результатов воздействия высокочастотных электромагнитных полей на образцы СТО.

В разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» указана последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых пользователь осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

В разделе «Сообщения пользователю» приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия пользователя (действия пользователя в случае сбоя, возможности повторного запуска программы и т.п.).

Оформление программного документа «Руководство пользователя» произведено по требованиям ЕСПД ГОСТ 19.101-77<sup>1)</sup>, ГОСТ 19.103-77<sup>2)</sup>, ГОСТ 19.104-78<sup>3)</sup>, ГОСТ 19.105-78<sup>4)</sup>, ГОСТ 19.106-78<sup>5)</sup>, ГОСТ 19.604-78<sup>6)</sup>.

© Госкорпорация «Росатом», ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина, 2024 г.

---

<sup>1)</sup> ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов

<sup>2)</sup> ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов

<sup>3)</sup> ГОСТ 19.104-78\* ЕСПД. Основные надписи

<sup>4)</sup> ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам

<sup>5)</sup> ГОСТ 19.106-78\* ЕСПД. Требования к программным документам, выполненным печатным способом

<sup>6)</sup> ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| 1. Назначение программы .....                                   | 5  |
| 1.1. Функциональное назначение .....                            | 5  |
| 1.2. Эксплуатационное назначение .....                          | 5  |
| 1.3. Состав функций .....                                       | 5  |
| 2. Условия выполнения программы .....                           | 7  |
| 2.1. Технические и климатические условия .....                  | 7  |
| 2.2. Минимальный состав технических и программных средств ..... | 7  |
| 2.3. Требования к персоналу .....                               | 8  |
| 3. Выполнение программы .....                                   | 9  |
| 3.1. Выполнение программы в среде ОС Linux .....                | 9  |
| 3.1.1. Подготовка к запуску программы .....                     | 9  |
| 3.1.2. Запуск программы .....                                   | 15 |
| 3.1.3. Выполнение программы .....                               | 23 |
| 3.1.4. Завершение программы .....                               | 25 |
| 3.2. Выполнение программы в среде ОС Windows .....              | 26 |
| 3.2.1. Подготовка к запуску программы .....                     | 26 |
| 3.2.2. Запуск программы .....                                   | 35 |
| 3.2.3. Выполнение программы .....                               | 36 |
| 3.2.4. Завершение программы .....                               | 37 |
| 4. Сообщения пользователю .....                                 | 38 |
| 4.1.1. Ошибки, связанные с параметрами .....                    | 41 |
| 4.1.2. Ошибки, связанные с выделением памяти .....              | 43 |
| 4.1.3. Ошибки, связанные с работой с файлами .....              | 43 |
| 4.1.4. Внутренние ошибки .....                                  | 44 |
| Приложение 1 Файл install.sh .....                              | 45 |
| Приложение 2 Файл cmake/inst .....                              | 48 |
| Приложение 3 Файл gcc/inst .....                                | 49 |
| Приложение 4 Файл gcc/inst.nvptx .....                          | 52 |
| Приложение 5 Файл hdf5/inst .....                               | 56 |
| Приложение 6 Файл mvapich/inst .....                            | 57 |
| Приложение 7 Файл qt/inst .....                                 | 59 |

|                                    |    |
|------------------------------------|----|
| Перечень сокращений .....          | 60 |
| Перечень ссылочных документов..... | 61 |

07623974.37016-01 90 01-2

## 1. НАЗНАЧЕНИЕ ПРОГРАММЫ

### 1.1. Функциональное назначение

Функциональным назначением ПО «Расчет-ЭМВ» является расчетная оценка воздействия высокочастотных ЭМП различного происхождения на образцы сложных технических объектов, на аппаратуру, системы и технические устройства на основе конструкторских 3D-моделей на стадиях проектирования, проведения исследовательских, доводочных и предварительных испытаний СТО.

### 1.2. Эксплуатационное назначение

ПО «Расчет-ЭМВ» должна эксплуатироваться в профильных подразделениях на объектах заказчика. Пользователями программы должны являться сотрудники профильных подразделений объектов заказчика.

### 1.3. Состав функций

1.3.1. ПО «Расчет-ЭМВ» обеспечивает возможность выполнения перечисленных ниже функций:

– расчет токов, наведенных при воздействии внешнего высокочастотного электромагнитного поля в проводниках корпуса образца СТО и экранах бортовой кабельной сети (модуль «Расчет-ММ»);

– расчет токов и напряжений во внутренних проводниках и нагрузках бортовой кабельной сети при протекании высокочастотных токов по экранам (оплеткам) БКС (модуль «Расчет-МПЛ»);

– возможность создания расчетных 3D-моделей для моделирования корпуса образца СТО с электрическими неоднородностями в виде люков, стыков, отверстий, элементы внутреннего заполнения (корпуса приборов, перегородки), экранов БКС (модуль «Модель-объект»);

– возможность создания расчетных моделей БКС для моделирования ее топологии, задания параметров нагрузок, характеристик экранов, геометрии поперечного сечения участков БКС (модуль «Модель-БКС»);

– возможность задавать данные, необходимые для запуска расчетов, осуществлять запуск, выводить и визуализировать полученные результаты в графической оболочке.

1.3.2. Функции модуля «Модель-объект», модуля «Модель-БКС» и графической оболочки реализованы в программном модуле RWEditor, описание работы с которым приведено во второй части руководства пользователя [1].

## 2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

### 2.1. Технические и климатические условия

Технические и климатические условия эксплуатации, при которых должна обеспечиваться нормальная работа программы и персонала, должны удовлетворять требованиям, предъявляемым к техническим средствам и рабочим помещениям в части условий их эксплуатации.

### 2.2. Минимальный состав технических и программных средств

2.2.1. В соответствии с [2], с учетом изложенной в этом документе структуры ПО «Расчет-ЭМВ» в состав технических средств должны входить:

1) для клиентской части ПО «Расчет-ЭМВ» (программный модуль RWEditor) ПЭВМ с характеристиками:

- универсальный процессор Intel Core I7 или аналогичный по производительности;
- оперативная память – не менее 16 Гб;
- HDD – не менее 1 Тб;
- видеокарта с производительностью не менее карты NVIDIA GTX 1080 (или аналогичной), поддерживающей OpenGL версии не ниже 3.3;
- монитор;
- мышь и клавиатура.

2) для серверной части ПО «Расчет-ЭМВ» (модули решатели) с учетом возможности массово-параллельных вычислений – вычислительный кластер в следующем техническом составе:

- не менее двух серверов с универсальными процессорами Intel Xeon 6 и выше, либо процессорами AMD EPYC и выше под управлением операционных систем CentOS 7.7 и выше, ПО Супер-ЭМВ 2.0 и выше, Арамид 3.3 и выше;
- оперативная память серверов – не менее 168 Гб;
- HDD сервера – не менее 1 Тб;
- видеоускоритель на сервере – NVIDIA V100 и выше;
- ПЭВМ для доступа к кластеру с вышеуказанными характеристиками.

2.2.2. В состав программных средств должны входить:

- операционная система Windows 10, Astra Linux 1.7, CentOS 7.7 и выше, ПО Супер-ЭВМ 2.0 и выше, Арамид 3.3 и выше;
- компиляторы GNU GCC версии 11.2.0-13.2.0 для серверной части ПО «Расчет-ЭМВ»;
- библиотеки окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019 (либо Microsoft Visual Studio 2019) для клиентской части ПО «Расчет-ЭМВ»;
- библиотека средств распараллеливания MPI – mvarich2 версии 2.3.6 для серверной части ПО «Расчет-ЭМВ»;
- набор библиотек инструментария разработки программного обеспечения для графических ускорителей NVIDIA HPC SDK 23.5 для серверной части ПО «Расчет-ЭМВ»;
- библиотека низкоуровневого распараллеливания для поддержки графических ускорителей nvptx-tools версии 1.0 для серверной части ПО «Расчет-ЭМВ»;
- фреймворк для разработки кроссплатформенных графических приложений Qt версии 5.15 для клиентской и серверной части ПО «Расчет-ЭМВ»;
- клиент удаленного доступа по протоколу SSH – openssh для клиентской и серверной части ПО «Расчет-ЭМВ».

Указанный состав библиотек входит в состав дистрибутива ПО «Расчет-ЭМВ».

### 2.3. Требования к персоналу

Для эксплуатации ПО «Расчет-ЭМВ» пользователь должен иметь опыт работы в среде современных операционных систем семейства Microsoft Windows и Linux, базовые знания в области применения ВВС для решения задач физико-математического моделирования, а также знания в области расчета воздействия электромагнитных полей на аппаратуру, системы и СТО.



### 3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

Программа по своему составу состоит из набора решателей – «Расчет-ММ», «Расчет-МПЛ» и графической оболочки (программного модуля) RWEditor, реализующей следующие функции программных модулей «Модель-объект», модуль «Модель-БКС» и модуля визуализации данных:

- возможность создания расчетных 3D-моделей для моделирования корпуса образца СТО с электрическими неоднородностями в виде люков, стыков, отверстий, элементы внутреннего заполнения (корпуса приборов, перегородки), экранов БКС (модуль «Модель-объект»);

- возможность создания расчетных моделей БКС для моделирования ее топологии, задания параметров нагрузок, характеристик экранов, геометрии поперечного сечения участков БКС (модуль «Модель-БКС»);

- возможность задавать данные, необходимые для запуска расчетов, осуществлять запуск, выводить и визуализировать полученные результаты.

#### 3.1. Выполнение программы в среде ОС Linux

##### 3.1.1. Подготовка к запуску программы

Для установки вспомогательного ПО необходимо выполнить установочный скрипт `thirdparty.sh`. В ходе выполнения этого скрипта будут собраны и установлены пакеты. Список пакетов и их версий приведен в таблице 1.

Таблица 1

| Пакет    | Версия |
|----------|--------|
| gcc      | 11.2.0 |
| cmake    | 3.20.1 |
| hdf5     | 1.10.5 |
| mvapich2 | 2.3.6  |
| qt       | 5.15.2 |
| lib      | 1.0    |

Скрипт `thirdparty.sh` представляет собой самораспаковывающийся архив. Этот архив содержит каталог `thirdparty` соответственно со следующими каталогами и файлами:

- `install.sh`;
- `smake`;
- `gcc`;
- `hdf5`;
- `mvarich`;
- `qt`;
- `lib`.

Файл `install.sh` осуществляет установку всех пакетов. После распаковки содержимого архива происходит выполнение этого файла. Содержимое файла приведено в приложении 1.

Каталог `smake` содержит файл `smake-3.20.1.tar.gz` с исходными текстами пакета `smake` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `smake`. Содержимое файла `inst` приведено в приложении 2.

Каталог `gcc` содержит файлы `binutils-2.38.tar.xz`, `gcc-11.2.0.tar.xz`, `gmp-6.2.1.tar.bz2`, `isl-0.25.tar.gz`, `mpc-1.2.1.tar.gz`, `mpfr-4.1.0.tar.gz`, `newlib-master.tar.gz` и `nvptx-tools-master.tar.gz` с исходными текстами, а также файлы `inst` и `inst.nvptx`. Файл `inst` отвечает за сборку и установку пакета `gcc`. Содержимое файла `inst` приведено в приложении 3. Файл `inst.nvptx` отвечает за сборку и установку пакета `gcc` с поддержкой устройств `Nvidia`. Содержимое файла `inst.nvptx` приведено в приложении 4.

Каталог `hdf5` содержит файл `hdf5-1.10.5.tar.gz` с исходными текстами пакета `hdf5` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `hdf5`. Содержимое файла `inst` приведено в приложении 5.

Каталог `mvarich` содержит файл `mvarich2-2.3.6.tar.gz` с исходными текстами пакета `mvarich` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `mvarich`. Содержимое файла `inst` приведено в приложении 6.

Каталог `qt` содержит файл `qt-everywhere-src-5.15.2.tar.xz` с исходными текстами пакета `qt` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `qt`. Содержимое файла `inst` приведено в приложении 7.

Каталог `lib` содержит набор опциональных для ПО «Расчет-ЭМВ» оптимизирующих библиотек.

Первым собирается компилятор gcc и окружение для его сборки. Далее с помощью этого компилятора собираются остальные пакеты. Возможна сборка компилятора с поддержкой ускорителей от компании Nvidia. В таблице 2 приведены пакеты, которые необходимо собрать для выполнения системных требований gcc.

Таблица 2

| Пакет       | Версия | Примечание                         |
|-------------|--------|------------------------------------|
| binutils    | 2.38   | –                                  |
| gmp         | 6.2.1  | –                                  |
| isl         | 0.25   | –                                  |
| mpc         | 1.2.1  | –                                  |
| mpfr        | 4.1.0  | –                                  |
| newlib      | 4.2.0  | только для gcc с поддержкой nvidia |
| nvptx-tools | 1.0    | только для gcc с поддержкой nvidia |

Пакеты, перечисленные в таблице 2, будут установлены в тот же каталог, что и компилятор gcc.

Сборка остальных, требуемых для системного окружения ПО пакетов, будет происходить в каталоге /tmp, при этом в конце сборки установочные файлы будут удалены.

Если существует необходимость установки дистрибутива ПО в конкретный каталог необходимо скрипту `thirdparty.sh` подать опцию `target`.

Например:

```
./thirdparty.sh --target /opt/thirdparty.
```

В ходе установки необходимо будет ответить на ряд вопросов:

```
- Input install dir [../lib]:
```

В этом случае запрашивается путь для установки вспомогательных библиотек, который может задаваться как относительным, так и как абсолютным. Значение по умолчанию: `../lib`.

Следующим вопросом определяется желаемое количество одновременно выполняемых нитей для сборки:

```
- Input max thread for build packages [16]:
```

В данном случае это параметр, который будет передан команде `make -j <значение>` для выполнения функции сборки всех пакетов. Значение по умолчанию: 16.

Следующим вопросом будет запрос на компиляцию под платформу GPU:

- Build gcc with nvidia support [Y/n]:

В данном случае определяется производить ли сборку компилятора gcc с поддержкой ускорителей nvidia. Значение по умолчанию: yes.

Для выполнения сборки ПО под GPU необходимо указать пути системного окружения:

- Input cuda dir [/opt/nvidia/hpc\_sdk/Linux\_x86\_64/20.9/cuda/]:

Данной строкой задается системный путь до предварительно установленной библиотеки CUDA. На вычислительном узле или ПЭВМ должны быть установлены драйвера для ускорителей Nvidia, соответствующие этой версии библиотеки CUDA. Вопрос будет задан только в случае, если выбрана сборка gcc с поддержкой ускорителей Nvidia. Значение по умолчанию: /opt/nvidia/hpc\_sdk/Linux\_x86\_64/20.9/cuda/.

- Build gcc without nvidia support [Y/n]:

Следующим будет вопрос о необходимости сборки компилятора gcc. Этот вопрос будет задан, если на предыдущем шаге был выбран отказ от сборки gcc с поддержкой ускорителей nvidia. Значение по умолчанию: yes.

Далее идет вопрос о необходимости сборки пакета cmake. Значение по умолчанию: yes.

- Build cmake [Y/n]:

Далее идет вопрос о необходимости сборки пакета hdf5. Значение по умолчанию: yes.

- Build hdf5 [Y/n]:

Далее идет вопрос о необходимости сборки пакета mvapich. Значение по умолчанию: yes.

- Build mvapich [Y/n]:

Далее идет вопрос о необходимости сборки пакета Qt. Значение по умолчанию: yes.

- Build qt (need c++ compiler with c++11 standart) [Y/n]:

Далее идет вопрос о установке опциональных оптимизирующих библиотек. Значение по умолчанию: yes.

- Install extra lib [Y/n]:

Пример вывода установочного скрипта thirdparty.sh приведен на рис. 1.

07623974.37016-01 90 01-1

```

[ocorninayu@ocn1 ntrn]$ ./thirdparty.sh --target /lhome/o37/ocorninayu/src
Creating directory /lhome/o37/ocorninayu/src
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Decrypting and uncompressing Thirdparty...
enter aes-256-cbc decryption password:
100%
Input install dir [../lib]: ../lib
Input max thread for build packages [16]: 8
Build gcc with nvidia support [Y/n]: n
Build gcc without nvidia support [Y/n]:y
===== GMP =====
===== ISL =====
===== MPFR =====
===== MPC =====
===== BINUTILS =====
===== GCC =====
---- GCC BUILD SUCCESS ----
Build cmake [Y/n]: y
===== CMAKE =====
---- CMAKE BUILD SUCCESS ----
Build hdf5 [Y/n]: y
===== HDF5 =====
---- HDF5 BUILD SUCCESS ----
Build mvapich [Y/n]: y
===== MMAPICH =====
---- MMAPICH BUILD SUCCESS ----
Build qt (need c++ compiler with c++11 standart) [Y/n]: y
===== QT5 =====
---- QT BUILD SUCCESS ----
Install extra lib [Y/n]: y
===== EXTRA LIB =====
---- EXTRA LIB INSTALL_SUCCESS ----

```

Рис. 1

Сборка каждого пакета должна завершаться сообщением BUILD\_SUCCESS.

По умолчанию сборка всех пакетов происходит в каталоге /tmp, при этом в конце сборки установочные файлы будут удалены.

Чтобы задать каталог, в котором будет происходить сборка пакетов, необходимо скрипту thirdparty.sh подать опцию target.

Если при сборке пакета произошла ошибка – выдаётся сообщение BUILD\_FAIL. Причину ошибки можно посмотреть в каталоге, где происходила сборка соответствующего пакета. Для просмотра диагностических сообщений при сборке пакетов в каждом подкаталоге исходных текстов создаются файлы журналов с именами: logconf, logstrap, logmake, logmakeinstall.

В результате выполнения установочного скрипта должен быть создан каталог, имя которого было указано на первом шаге установки. Каждый пакет устанавливается в соответствующий подкаталог (cmake, gcc, hdf5, mvapich, qt).

Например, при выборе пользователем в качестве каталога установки вспомогательного ПО /home/user/empscore/lib в его домашнем каталоге будет создана следующая структура, приведенная на рис.2.

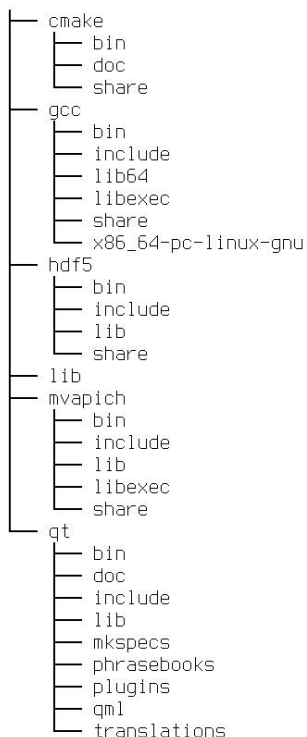


Рис. 2

Установка дистрибутива ПО «Расчет-ЭМВ» осуществляется с помощью установочного скрипта `etalon.sh`.

В процессе установки будет запрошен пароль, сообщаемый разработчиком ПО конкретному заказчику.

Установка дистрибутива состоит из двух процедур:

- установка клиентской части (графическое приложение RWEditor и набор решателей для ПЭВМ под управлением ОС Linux);
- установка серверной части (набор решателей и примеров скриптов для запуска ПО на вычислительной системе).

Диалоговая часть интерфейса установки показана на рис. 3.

```

[ocorninayu@ccf1 Install]$ ./etalon.sh
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Decrypting and uncompressing Etalon...
enter aes-256-cbc decryption password:
100%
Input install dir [/tmp/Etalon]:
Install user gui interface [Y/n]:
---- GUI INSTALL SUCCESS ----
Install cli solver [Y/n]:
---- CLI INSTALL SUCCESS ----

Install project done.
  
```

Рис.3

Полный состав дистрибутива приведен в таблице 3.

Таблица 3

| Имя каталога | Описание  |
|--------------|---|
| RWEditor     | Содержит необходимые файлы для использования ПО «Расчет-ЭМВ» в графическом режиме (клиентская часть).   |
| bin          | Содержит набор решателей для использования в консольном режиме счета. Устанавливается на вычислительный кластер (серверная часть).                      |
| examples     | Содержит набор примеров запуска расчета для системы управления кластером на базе ПО Slurm. Устанавливается на вычислительный кластер (серверная часть). |

После завершения процедуры установки необходимо в окружении пользователя задать пути к библиотекам ранее установленного вспомогательного программного обеспечения. Например, пути можно определить в файле конфигурации командной оболочки BASH - `.bashrc` добавив туда строку следующего содержания:

```
export LD_LIBRARY_PATH=/home/user/empcore/lib:$LD_LIBRARY_PATH
```

В данной строке каталог `/home/user/empcore/lib` – это префикс-каталог, заданный в ходе установки вспомогательного программного обеспечения.

Кроме того, для взаимодействия клиентской части ПО «Расчет-ЭМВ» с лицензионным сервером требуется в системном окружении пользователя установить значения переменных окружения «LLIC\_HOST» и «LLIC\_PORT», в которых задается доменное имя лицензионного сервера и номер TCP-порта лицензионного сервера соответственно.

Лицензионный сервер устанавливается на ПЭВМ пользователя под управлением ОС Windows и входит в состав дистрибутива ПО «Расчет-ЭМВ» для ОС Microsoft Windows.

### 3.1.2. Запуск программы

В соответствии с документом [2], в состав ПО «Расчет-ЭМВ» входит клиентская и серверная части.

Запуск в графическом режиме происходит путем запуска скрипта `runRWE.sh` в каталоге дистрибутива `RWEditor`. Процедура дальнейшей работы с графической оболочкой `RWEditor` подробно расписана во второй части руководства пользователя [1] и является общей для ОС Linux или ОС Windows.

Запуск в консольном режиме можно рассмотреть на примере подкаталога задания расчета модели исследуемого изделия, например, поставляемого в составе дистрибутива и размещаемого в каталоге установки `examples/Model3`.

Структура каталога расчета выглядит следующим образом, представленном на рис. 4.

```

mc [shikhinsky@ccf1]:/ll/Forv/shishinsky/for_for/Model3
/ll/Forv/shishinsky/Forv/Model3
-ВВЕРХ-
Имя                                     Размер  Время  права
EMP.ini                                4648   apr 22 12:54
EMPCore                                6319984 apr 18 15:25
Frequencies.txt                          23   apr 19 14:24
MTL                                     6345928 apr 18 09:48
model30.sio                              972817 май 18 2023
run_slurm                                385   apr 19 14:49
start                                    191   apr 19 14:49
startconsole                             242   apr 22 12:54

-ВВЕРХ-
47T/81T (58%)
Совет: Вы можете выбрать редактор для F4 с помощью переменной оболочки
[shikhinsky@ccf1 Model3]$
Помощь Меню Просмотр Правка Копия

```

Рис. 4

В каталоге находятся два бинарных файла решателей `EMPCore` – «Расчет-ММ» и `MTL` – «Расчет МПЛ», исходный файл для расчета `model3.sio` в формате `SIO`, опциональные файлы параметризованного запуска `EMP.ini` и `Frequencies.txt`, а также скрипты запуска в консольном режиме, в том числе с использованием ПО системы управления `Slurm` [3] для ВВС.

Запуск расчетных модулей в консольном режиме без системы управления осуществляется скриптом `startconsole`, содержимое которого представлено на рис. 5.



```

startconsole  [----] 0 L:[ 1+ 8 9/ 20] *(236 / 248b) 10 0x00A KOI8-R  [*] [X] ^
#!/bin/bash

export HDF5_USE_FILE_LOCKING=FALSE
export LD_LIBRARY_PATH=/opt/gcc/11.2.0/lib64:/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/lib:$LD_LIBRARY_PATH
export PATH=/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/bin:$PATH

mpirun -np 2 ./EMPCore
./MTL
  
```

Рис. 5

В скрипте предварительно формируются переменные окружения, необходимые для запуска решателей (каталоги установки вспомогательного ПО), а также определен порядок запуска решателей – модуль «Расчет-ММ» (EMPCore), затем модуль «Расчет-МПЛ» (MTL).

Модуль «Расчет-ММ» (EMPCore) использует гибридную схему распараллеливания (MPI для частот и OpenMP для матричных решателей), модуль «Расчет-МПЛ» (MTL) реализован с поддержкой функции распараллеливания на общей памяти (OpenMP).

Оптимальная схема запуска (количество MPI-процессов и количество OpenMP потоков) подбирается экспериментально и индивидуально исходя из параметров вычислительного оборудования (количества вычислительных узлов, оперативной памяти, количества ядер, экспериментальной определенной эффективности распараллеливания на потоках) и физико-математической постановки задачи (количество частот, количество ячеек сетки). Для оценки используемого количества памяти можно взять пример сетки с количеством ячеек в 34000 элементов, что соответствует примерно 40 ГБ памяти для решателя «Расчет-ММ» (матричный решатель gauss\_p).

В качестве параметра для запуска на два MPI процесса модуля «Расчет-ММ» в примере указан ключ `-np 2`. При необходимости этот ключ может быть изменен на другое значение, которое определяется количеством одновременно считаемых частот (один MPI процесс ведет расчет одной частоты) и количеством доступных расчетчику вычислительных ресурсов.

Запуск расчетных модулей в консольном режиме с использованием системы управления Slurm осуществляется скриптом `run_slurm`, содержимое которого представлено на рис. 6.

```

mc [rshinshinsyu@ccf1]:/tmp/Model3
run slurm [-M--] 0 L: [ 1+23 24/ 25] *(384 / 385b) 10 0x00A KOI8-R [*] [X]
#!/bin/sh

NP=2 # num. MPI processes
PPN=2

dir='pwd'

NN=$(( $NP / $PPN ))
echo "$NP proc on $NN nodes (PPN=$PPN)"

namevar="2823"

sbatch \
  --job-name=EMRC.000001.Task03 \
  --workdir=$dir \
  --output="$dir/Task0103-$namevar-n${NP}-%j.out" \
  --exclusive \
  -N $NN \
  --ntasks=$NP \
  --ntasks-per-node=$PPN \
  --time=60000 \
  --mem=55000 \
  $dir/start
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перезагрузить 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 6

Данный скрипт определяет набор опций для системы управления Slurm, таких как имя задачи в очереди (--job-name), рабочий каталог (--workdir), имя файла вывода (--output), режим счета (--exclusive), общее количество процессов задачи (для параллельной задачи, --ntasks), количество процессов задачи на узле (--ntasks-per-node), время счета (--time), ограничение по использованию памяти на вычислительном узле (--mem) и имя запускаемого скрипта задачи, в данном случае это start.

В качестве параметра для запуска на два MPI процесса модуля «Расчет-ММ» указаны опции --ntasks=2 и --ntasks-per-node=2. При необходимости значения этих опций может быть изменено в соответствии с количеством одновременно считаемых частот (один MPI процесс ведет расчет одной частоты) и количеством доступных расчетчику вычислительных ресурсов (в данном случае на одном вычислительном узле запускаются два MPI процесса расчетного модуля).

Содержимое запускаемого скрипта start представлено на рис. 7.

```

mc [shikshinskyu@ccf1]:/11/forv/ishikshinskyu/for_forv/Model3
start [----] 0 L: [ 1+ 4 5/ 18] *(149 / 208b) 10 0x00A KOI8-R [*] [X] ^
#!/bin/bash

export HDF5_USE_FILE_LOCKING=FALSE
export LD_LIBRARY_PATH=/opt/gcc/11.2.0/lib64:/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/lib:$LD_LIBRARY_PATH

time srun --mpi=pmi2 ./EMPCore
time srun ./MTL

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перезагрузить 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 7

При необходимости, расчет модулей «Расчет-ММ» и «Расчет-МПЛ», можно производить отдельно, при этом из скриптов запуска убирается ненужная строка с именем бинарного файла соответствующего модуля решателя.

ПО «Расчет-ЭМВ» позволяет использовать оптимизирующие библиотеки стороннего производителя для ускорения процесса расчета. Для этого используются бинарные версии решателей «Расчет-ММ» (EMPCore\_OPT) и «Расчет-МПЛ» (MTL\_OPT), входящие в состав дистрибутива ПО «Расчет-ЭМВ». Необходимо учитывать при расчетах, что данные версии решателей используют удвоенное количество памяти при расчете задачи.

Пример использования оптимизированных решателей в консольном режиме представлен в каталоге examples/Model3\_OPT. Структура каталога расчета выглядит следующим образом, представленном на рис. 8.

```

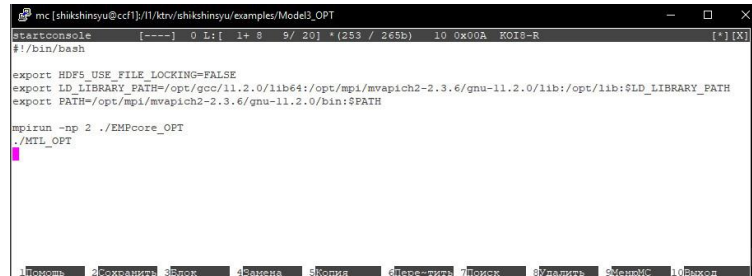
mc [shikshinskyu@ccf1]:/11/rvv/ishikshinskyu/examples/Model3_OPT
Левая панель  Файл  Команда  Настройки  П
<- /11/rvv/ishikshinskyu/examples/Model3_OPT .[*]>
'и      Имя      Размер  Время правки
/..     -ВВЕРХ-
*EMP.ini      4646   апр 18 15:28
*EMPCore_OPT 6328448 апр 18 11:48
*Frequencies.txt 23     апр 19 14:06
*MTL_OPT     6346176 апр 18 09:48
model3.sio   972817 май 18 2023
*run_slurm   386    апр 18 15:26
*start       227    апр 22 15:50
*startconsole 248    апр 22 13:43

-ВВЕРХ-
47T/81T (58%)
Совет: Для отметки каталогов в диалоге выбора, добавьте
[shikshinskyu@ccf1 Model3_OPT]$
1Помощь 2Меню 3Просмотр 4Правка 5Копия

```

Рис. 8

В этом случае запуск расчетных модулей в консольном режиме без системы управления осуществляется скриптом `startconsole`, содержимое которого представлено на рис. 9, а пример запускаемого через систему управления скрипта `start` выглядит так, как представлено на рис. 10.



```

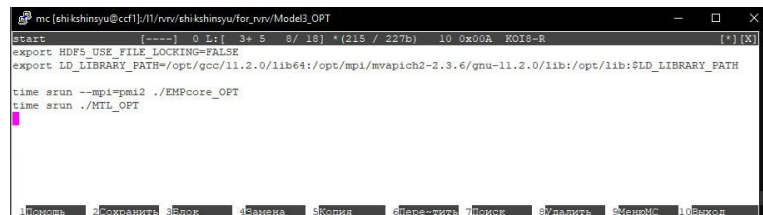
mc [shikhinsky@cdf1] /1/krv/shikhinsky/examples/Model3_OPT
startconsole
#~/bin/bash

export HDF5_USE_FILE_LOCKING=FALSE
export LD_LIBRARY_PATH=/opt/gcc/11.2.0/lib64:/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/lib:/opt/lib:$LD_LIBRARY_PATH
export PATH=/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/bin:$PATH

mpirun -np 2 ./EMPCore_OPT
./MTL_OPT

```

Рис. 9



```

mc [shikhinsky@cdf1] /1/nrv/shikhinsky/for_nrv/Model3_OPT
start
export HDF5_USE_FILE_LOCKING=FALSE
export LD_LIBRARY_PATH=/opt/gcc/11.2.0/lib64:/opt/mpi/mvapich2-2.3.6/gnu-11.2.0/lib:/opt/lib:$LD_LIBRARY_PATH

time srun --mpi=mpi2 ./EMPCore_OPT
time srun ./MTL_OPT

```

Рис. 10

В обоих вышеуказанных скриптах в переменной окружения `LD_LIBRARY_PATH` добавлены пути к установленным сторонним библиотекам оптимизации (`/opt/lib`).

По умолчанию, все параметры для расчета, за исключением входного файла данных, формируются автоматически при запуске бинарного файла модуля решателя. При этом пользователь может задать часть параметров самостоятельно в конфигурационном файле `EMR.ini`, пример которого представлен на рис. 11. Данный файл содержит комментарии по каждой строке конфигурации и может быть сформирован автоматически при запуске, например, бинарного файла решателя `EMPCore`.

Основными, значимыми для пользователя, параметрами данного файла являются исходный файл расчета (параметр `Task`), файл с выводом рассчитанных токов (параметр `SolveCurrents`), выбор матричного решателя (параметр `SolverType`), количество нитей OpenMP для матричных решателей (параметр `OMP_ThreadsQ`), количество нитей `libthreads` для коэффициентов матриц (параметр `ThreadsQ`), задание возможности работы со сторонним форматом HDF (параметр `AllowHDF`) и управление расширенным выводом расчетных модулей (параметр `VerboseMode`). Все остальные параметры являются параметрами по умолчанию, имеют отладочный характер и предназначены для разработчика программы. Изменение их пользователем не допускается.

```

msc [---] 32 L1: [ 1+82 83/ 85] (4666/466b) 10 ОКЛА КОПБ-Р
// файл ввода/вывода
// задача, которая будет решаться в HDF или SIO (то есть описание сцены)
Task //SimRSocket_Task.sio
Results //SimRSocket_Result.sio
// управление выводом
// период выгрузки состояния для перезапуска расчёта
DmpPeriod 0
// сохранять ли все матрицы и вектора в специальный бинарный файл (для настройки решателей)
SaveMatrixToFile false
// выводить ли значения токов на консоль во время расчёта
PrintCurrents false
// загружать ли в текстовые файлы матрицу Z
SaveZToFile true
// выгружать ли в текстовые файлы вектора X и Y
SaveXYToFile false
// раскраска в BIM файлах по амплитуде (Amplitude) или по фазе (Phase) тока
SurfFaceColor Phase
// количество знаков в выводимых файлах с током
PrintPrecision 12
// количество знаков после запятой при сравнении координат узлов в метрах
NodePrecision 7
---
// Управление счетом
// SolverType - тип решателя: 0 - оригинальный метод Гаусса, 1 - gauss_P, 2 - gauss_s на орeMP, 3 - gauss_smg - функциональный решатель
// 5 - gauss_U, 6 - gauss_BK - блочный решатель, 7 - gauss_BK_ORG - блочный решатель с использованием оптимизации, 8 - gauss_ORG - чистый оптимизированный решатель
// 9 - gauss_P_mv - решатель на GPU, 10 - solvef_auto - автоматический выбор решателя
// SolverType gauss_BK_ORG
// доступная ширина интервала для одноо преобуславливателя
// тип интегрирования по поверхности: 0 - по стандартной сетке на треугольнике, 1 - по адаптивной сетке симметричным образом, 2 - нечет Сесилия
// 3 - как в 1, но факторизовано, 4 - как в 2, но факторизовано, 5 - версия по Умолчанию,
// 6 - новейшее интегрирование, 7 - аналитическое интегрирование,
// 8 - аналитическое интегрирование по сетке Дувоана
//
//
// SurfFaceIntegrallType 5
// размер сетки интегрирования внутри треугольников. 0 - автоматический выбор разбиения
TriangleSize 25
// использование сетки Дувоана при интегрировании по треугольникам
UseDuvanovaIntegrallType true
// тип интегрирования по проволокам: 0 - по средним отрезкам, 1 - по адаптивной сетке симметричным образом, 6 - новейшее интегрирование
WireIntegrallType 5
// тип симметризации интеграла SCS1: 0 - среднее арифметическое, -1 - по минимуму, +1 - по максимуму, 2 - двойное интегрирование, -2 - без симметризации
SCS1SymmetrizationType 2
// количество разбиений поверхности проволоки по углу вблизи контакта
WireSubdivisions 10
// количество разбиений поверхности проволоки по длине вблизи контакта
WireSubdivisionsL 20
// симметризация матрицы Z
Symmetrized Z false
// ThreadSQ - количество потоков для многоядерных вычислений
ThreadSQ 32
// OMP_ThreadsQ - количество потоков для OреMP решателей
OMP_ThreadsQ 32
// множитель, на который домножаются выводимые в текстовые таблицы значения тока
ImMultiplier 1
// не производить умножение токов в проволоках на 2*E1*R
NoPIR true
// выводить ли полную таблицу токов (массив PointOfCurrent игнорируется при этом)
OutputCurrentTable false
// выводить ли ток в левые и правые поле в отдельный текстовый файл
SaveLeftRightCurrents true
// записывать ли ближние и дальние поле в текстовые файлы в формате FEKO
SaveFieldToFEKOFiles true
// сохранять ли результаты расчёта в бинарный файл для EKS расчёта
SaveInfoToFile true
// выводить ли состояние расчёта в progress.txt
ProgressBar true
// использовать дополнительные возможности консоли
UseCSI false
// использовать ли результаты предыдущего расчёта при отсутствии изменений во входных данных
PrevResult true
// форсировать использование предыдущих результатов
ForceUsingPrevResult false
// проверять ли сетку на согласованность
CheckSet false
// сохранять ли решение с файлом
SaveSetToFile true
// предельно допустимое относительное отклонение от эталона
MaxRelativeError 0.00000001
// позволять ли считать с HDF входных файлов
AllowHDF true
// расширенный вывод
VerboseMode false

```

Рис. 11

По умолчанию пользователю доступен матричный решатель типа 1, данный решатель (gauss\_p в конфигурационном файле) использует строчную реализацию метода Гаусса решения СЛАУ с несколькими правыми частями с частичным выбором ведущего элемента. Распараллеливание выполнено средствами OpenMP.

В качестве опциональных пользователь может выбрать также для расчетов решатель типа 8 (gauss\_opt), в котором используются функции оптимизированной параллельной библиотеки стороннего производителя для приведения матрицы к LU-разложению и решения СЛАУ с треугольной матрицей, либо авторешатель под номером 10, который определяет оптимальный алгоритм решения СЛАУ, в зависимости от ранга формируемой матрицы.

В качестве опционального файла конфигурации выступает также файл Frequencies.txt, в котором пользователь может задать вручную значения требуемых частот внешнего электромагнитного поля, для которого будет производиться расчет. Пример содержимого файла Frequencies.txt представлен на рис. 12. В данном случае расчет будет проводиться в диапазоне от 5 до 40 МГц с шагом 5 МГц.



Рис. 12

Для решателя «Расчет-МПЛ» (файлы решателей MTL и MTL\_OPT) в конфигурационном файле EMP.ini обязательным параметром является MTL\_input, в котором указывается путь к файлу конфигурации БКС. Опционально этот файл можно указать в командной строке, указав его в качестве параметра для указанных решателей (например, ./MTL mtl\_Cabling.txt).

### 3.1.3. Выполнение программы

Выполнение расчетов с использованием графической оболочки RWEEditor в ОС Linux проводится в соответствии с описанием, изложенном во второй части руководства пользователя [1].

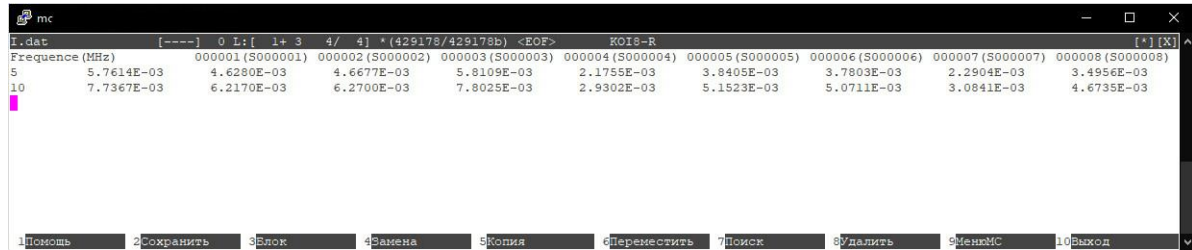
В случае использования консольного режима расчетов ПО «Расчет-ЭМВ» при запуске скрипта, пример см. рис. 5, пользователь получит интерактивный вывод на экран всего процесса расчета. Содержимое вывода практически одинаково как для случая использования ВВС и системы управления Slurm, так и для обычного локального счета на ПЭВМ. Пример вывода расчетного модуля «Расчет-МПЛ» представлен на рис. 13.

В случае использования ПО Slurm вышеуказанный вывод может быть перенаправлен в отдельный файл stdout, путь к которому определяется путем задания опции `-output` в конфигурационном файле скрипта запуска (см. рис. 6).

В случае необходимости вывода дополнительной информации о результатах расчета, например, значений токов, посчитанных в базисных функциях, в конфигурационном файле указывается соответствующий параметр `SolveCurrents` с именем файла вывода. При этом в текущем каталоге расчета будет сформирован файл со значениями частот и рассчитанных токов, пример содержания которого показан на рис. 14.



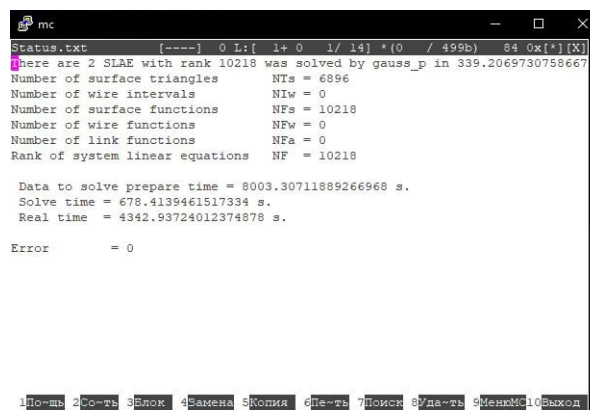




| Frequency (MHz) | 000001 (S000001) | 000002 (S000002) | 000003 (S000003) | 000004 (S000004) | 000005 (S000005) | 000006 (S000006) | 000007 (S000007) | 000008 (S000008) |
|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 5               | 5.7614E-03       | 4.6280E-03       | 4.6677E-03       | 5.8109E-03       | 2.1755E-03       | 3.8405E-03       | 3.7803E-03       | 2.2904E-03       |
| 10              | 7.7367E-03       | 6.2170E-03       | 6.2700E-03       | 7.8025E-03       | 2.9302E-03       | 5.1523E-03       | 5.0711E-03       | 3.0841E-03       |
|                 |                  |                  |                  |                  |                  |                  |                  | 4.6735E-03       |

Рис. 14

При завершении счета ПО «Расчет-ЭМВ» в текущем каталоге расчетов будет сформирован файл с информацией о времени счета и статусе завершения программы Status.txt, пример которого приведен на рис. 15.



```

Status.txt [----] 0 L: [ 1- 0 1/ 14] * (0 / 499b) 84 0x[*] [X]
Here are 2 SLAE with rank 10218 was solved by gauss_p in 339.2069730758667
Number of surface triangles Nfs = 6996
Number of wire intervals NIW = 0
Number of surface functions NFs = 10218
Number of wire functions NFW = 0
Number of link functions NFa = 0
Rank of system linear equations NF = 10218

Data to solve prepare time = 8003.30711889266968 s.
Solve time = 678.4139461517334 s.
Real time = 4342.93724012374878 s.

Error = 0
  
```

Рис. 15

### 3.1.4. Завершение программы

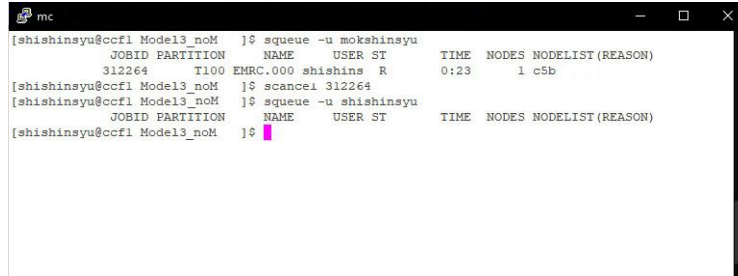
Успешным признаком завершением счета программы является наличие файла Status.txt в расчетном каталоге, в котором содержится запись об отсутствии ошибок Error = 0. При отсутствии такого файла в расчетном каталоге пользователь должен проанализировать либо интерактивную консоль вывода, либо файл вывода системы управления Slurm на отсутствие ошибок.

Завершение счета ПО «Расчет-ЭМВ» с использованием графической оболочки RWEditor проводится в соответствии с описанием, изложенном во второй части руководства пользователя [1].

В случае необходимости принудительного завершения счета при использовании консольного режима без использования ПО Slurm можно использовать сочетание клавиш |Ctrl|+|c|.

Для принудительного завершения счета при использовании консольного режима с использованием ПО Slurm можно использовать набор команд ПО Slurm, например, scancel

<номер задачи>, как показано на рис. 16 (для пакетного режима счета). В случае использования интерактивного режима счета на системе управления ПО Slurm для завершения программы можно также воспользоваться сочетанием клавиш |Ctrl|+|c|.



```

[shishinsyu@ccf1 Model3_noM ]$ squeue -u mokshinsyu
          JOBID PARTITION   NAME       USER  ST       TIME  NODES NODELIST(REASON)
312264      T100    EMRC.000 shishins  R         0:23     1  c5b
[shishinsyu@ccf1 Model3_noM ]$ scancel 312264
[shishinsyu@ccf1 Model3_noM ]$ squeue -u shishinsyu
          JOBID PARTITION   NAME       USER  ST       TIME  NODES NODELIST(REASON)
[shishinsyu@ccf1 Model3_noM ]$

```

Рис. 16

### 3.2. Выполнение программы в среде ОС Windows

#### 3.2.1. Подготовка к запуску программы

Дистрибутив ПО «Расчет-ЭМВ» для ОС Microsoft Windows поставляется в виде RAR архива с именем Расчет-ЭМВ.rar. После распаковки архива структура каталога дистрибутива выглядит так, как показано на рис.17.

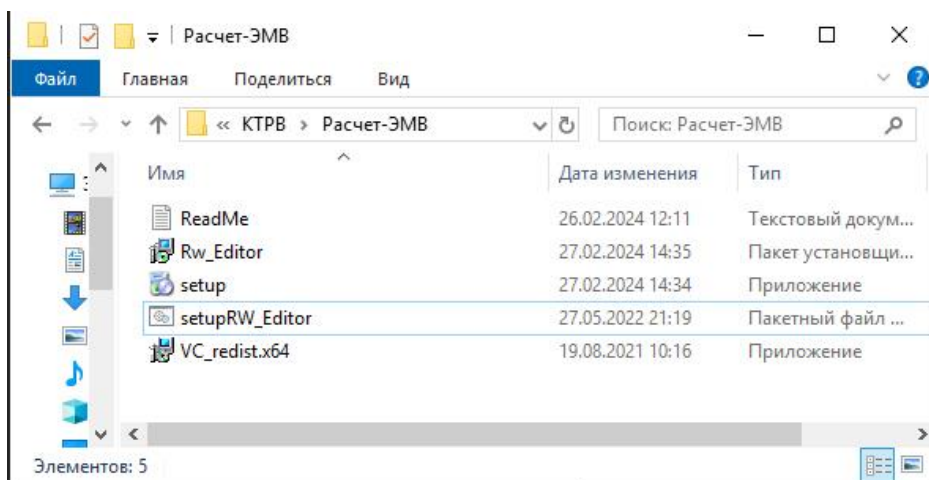


Рис. 17

Для установки ПО «Расчет-ЭМВ» в среде ОС Windows необходимы права администратора операционной системы. От имени учетной записи администратора ПЭВМ необходимо запустить файл setupRw\_Editor.bat и следовать предлагаемым действиям программы установки. При установке предлагается путь по умолчанию C:\Program Files\VNIITF\Rw\_Editor\. При необходимости можно установить свой путь, отредактировав

его в строке ввода или выбрав конкретный путь, как показано на рис. 18 и нажать кнопку «Далее».

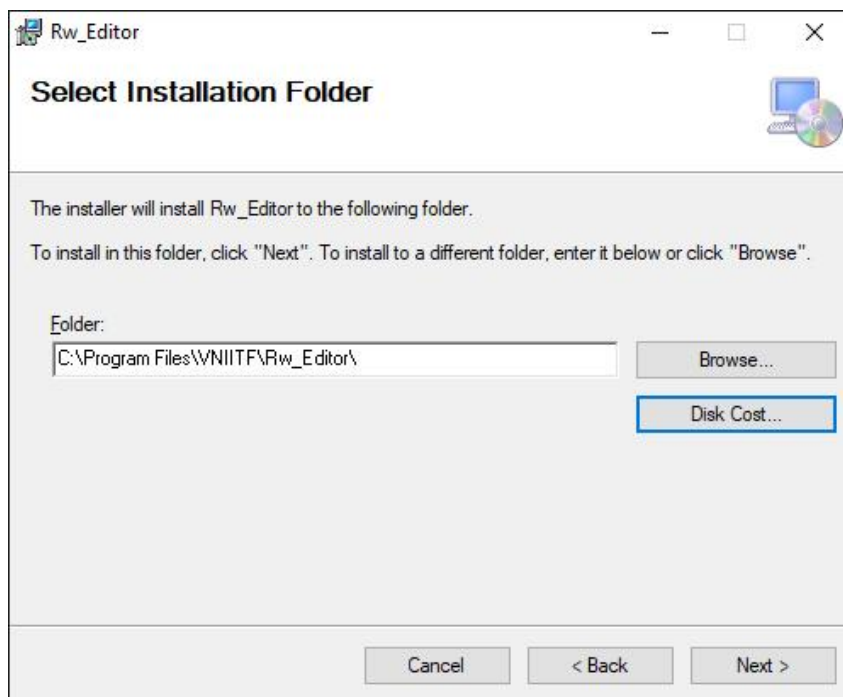


Рис. 18

После установки ПО «Расчет-ЭМВ» автоматически запускается установка библиотек окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019 (либо Microsoft Visual Studio 2019) VC\_redist.x64.exe. В конце установки этого ПО необходимо поставить галочку о согласии.

Кроме того, в процедуру установки ПО «Расчет-ЭМВ» входит установка библиотек MPI: mpich2-1.4.1p1-win-x86-64.msi. Запуск установщика также выполняется автоматически после завершения установки библиотек окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019. При установке выполняется проверка на наличие ПО Microsoft .NET Framework 3.5, которое должно быть предварительно установлено в операционной системе.

В случае отсутствия установленного ПО Microsoft .NET Framework 3.5 оно может быть установлено путем выполнения следующей команды в оболочке PowerShell ОС Windows от имени администратора:

```
dism /online /enable-feature /featurename:NetFX3 /All /source:D:\sources\sxs /limitaccess
```

, где D:\sources\sxs – каталог дистрибутива операционной системы.

В случае, если необходимо провести установку обновлений ПО «Расчет-ЭМВ» права администратора не требуются. Для выполнения этой процедуры пользователь должен запустить полученный от разработчика файл setup.exe, указать путь, куда была установлена предыдущая версия (например, C:\VNIITF\Rw\_Editor\ ) и выполнить установку.

После успешной установки на рабочий стол ПЭВМ пользователя будут добавлены ярлыки для запуска программы RWEditor, которая и будет содержать весь необходимый для использования ПО «Расчет-ЭМВ» в ОС Windows функционал.

В ходе установки ПО «Расчет-ЭМВ» также будет установлен лицензионный сервер.

Для настройки переменных окружения для взаимодействия клиентской части ПО «Расчет-ЭМВ» с лицензионным сервером требуется запустить с правами администратора утилиту control, предоставляющую доступ к классической панели управления ОС Windows, как показано на рис 19, 20.

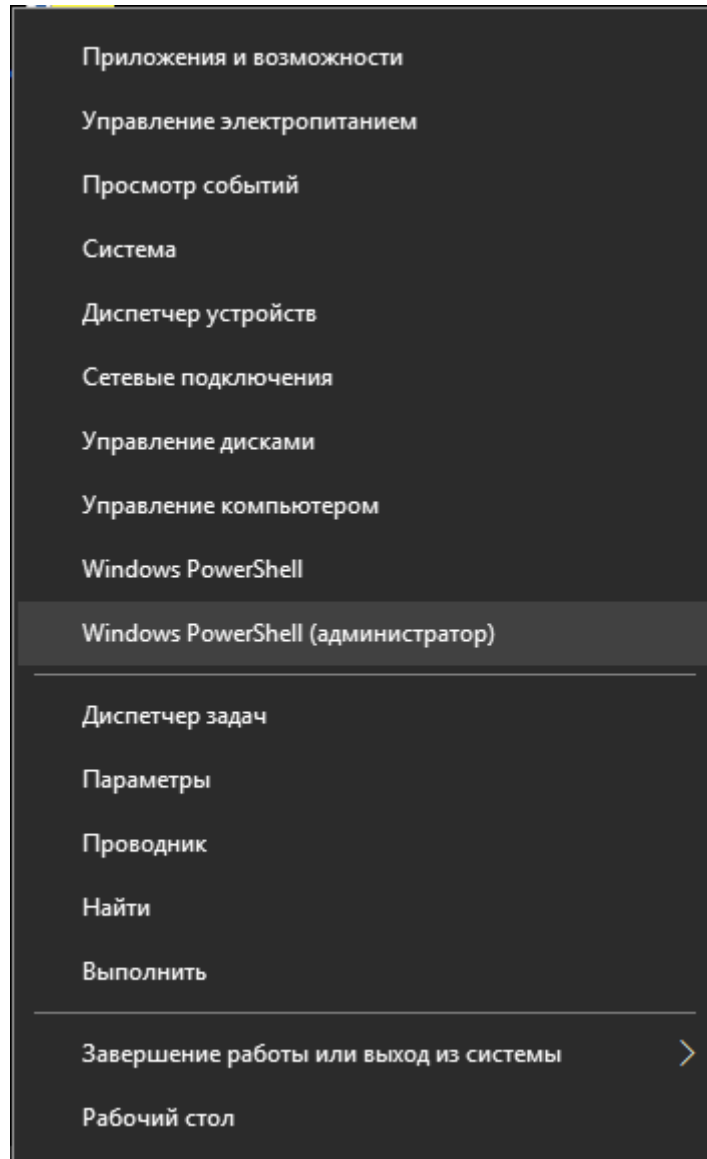


Рис. 19

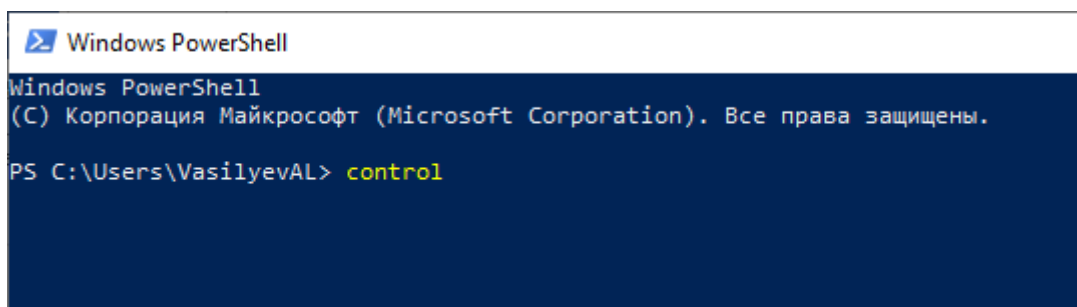


Рис. 20

В появляющихся окнах последовательно выбрать разделы «Система и безопасность»  
-> «Система» -> «Дополнительные параметры системы», как показано на рис. 21, 22, 23.

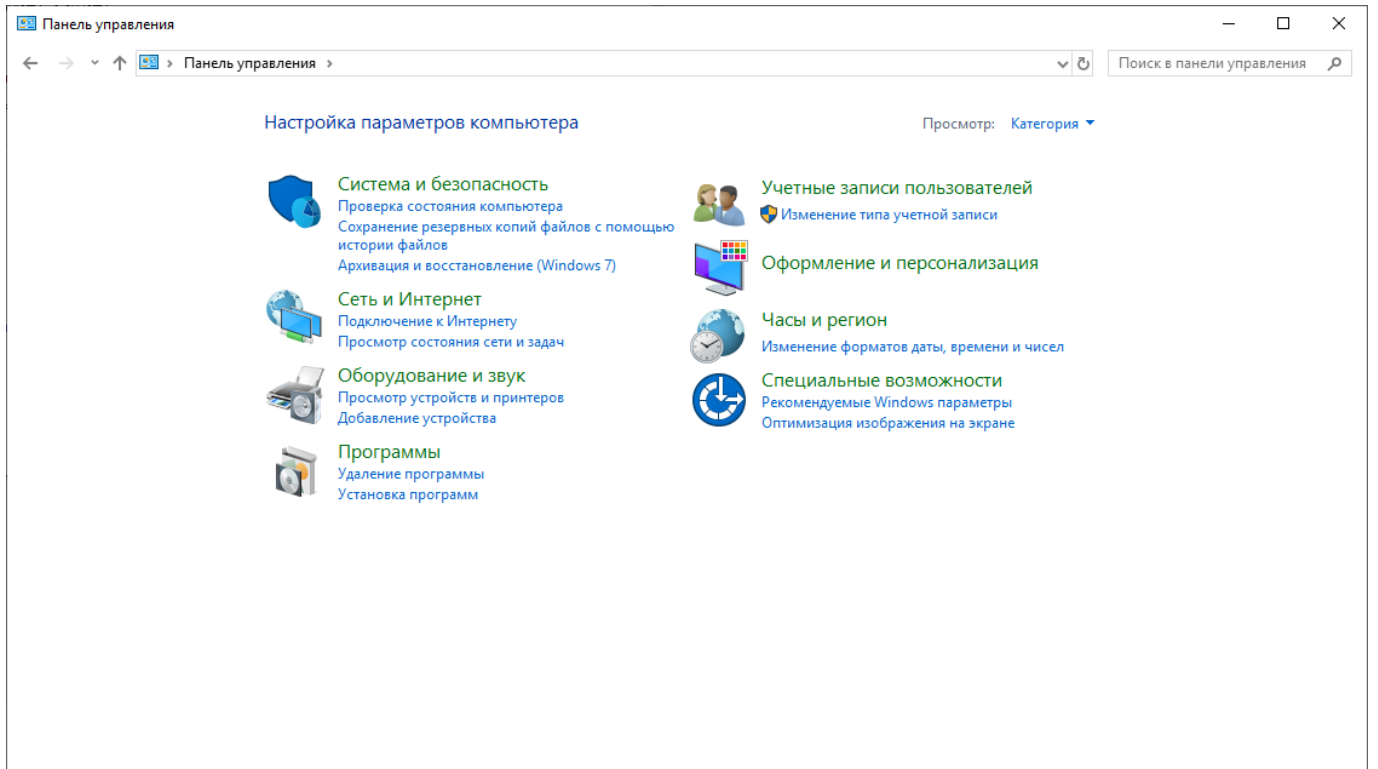


Рис. 21

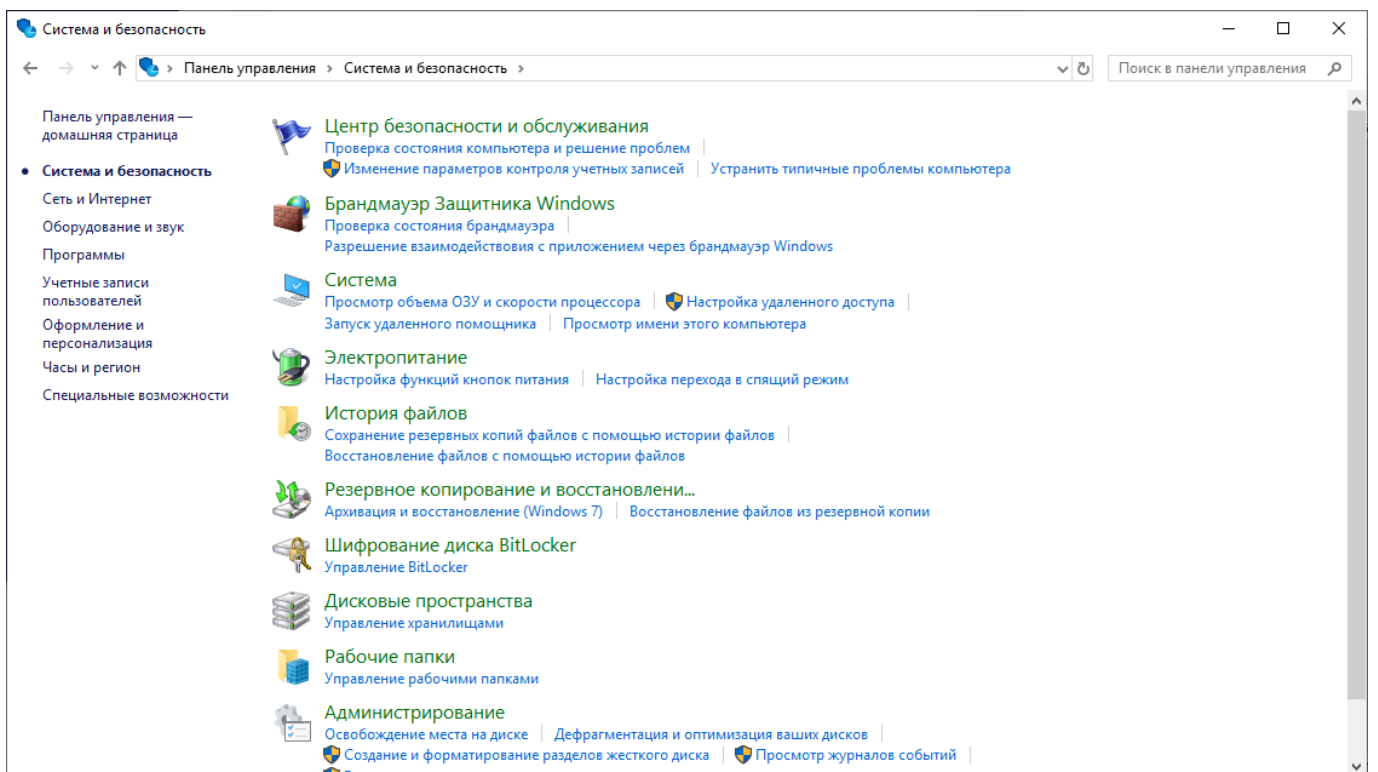


Рис. 22

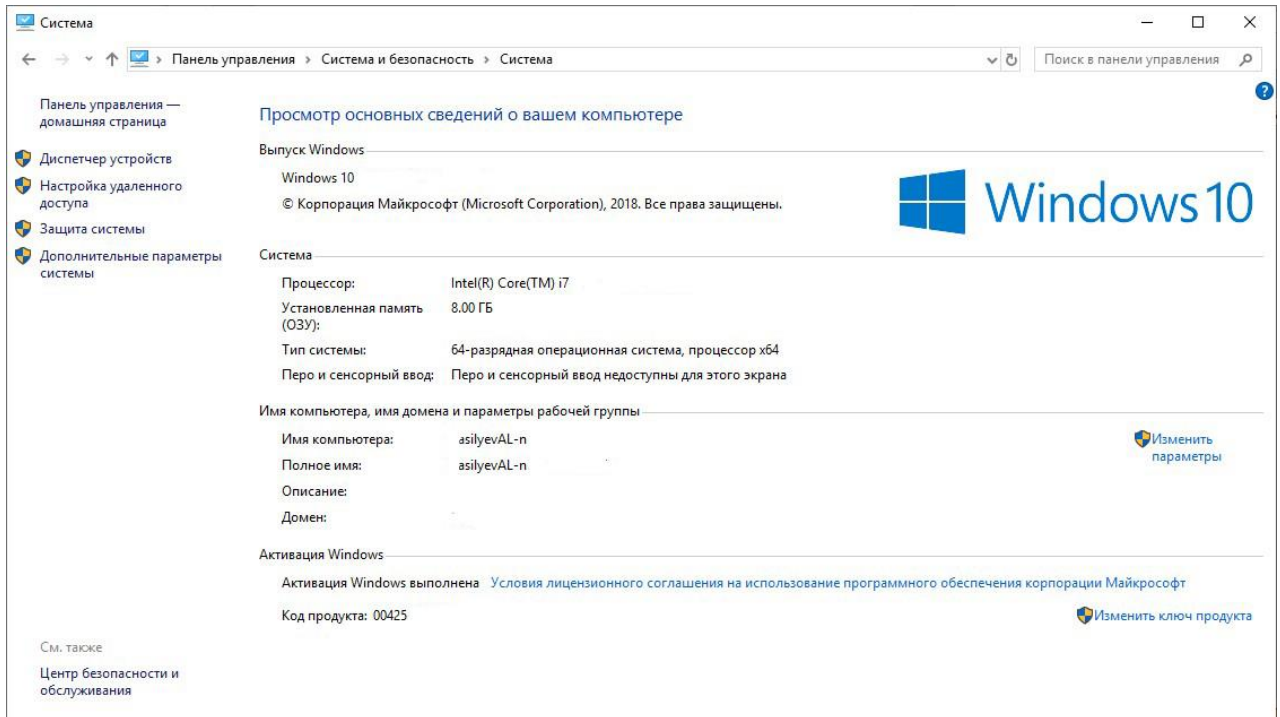


Рис. 23

В окне свойств системы на вкладке «Дополнительно» нажать кнопку «Переменные среды», как показано на рис. 24.

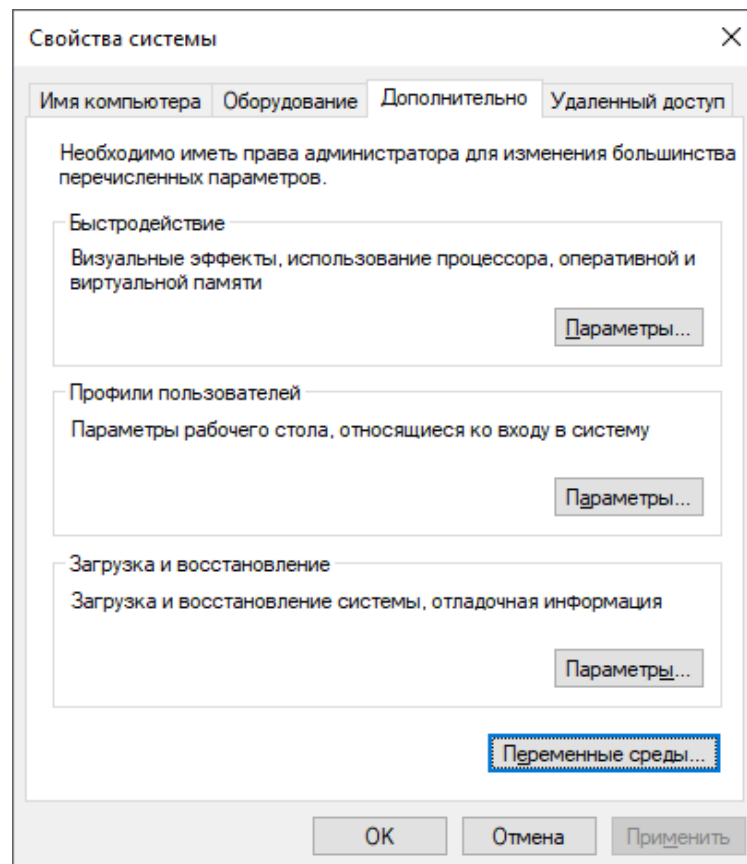


Рис. 24

В окне переменных среды в разделе «Системные переменные» нажать кнопку «Создать», как показано на рис. 25.

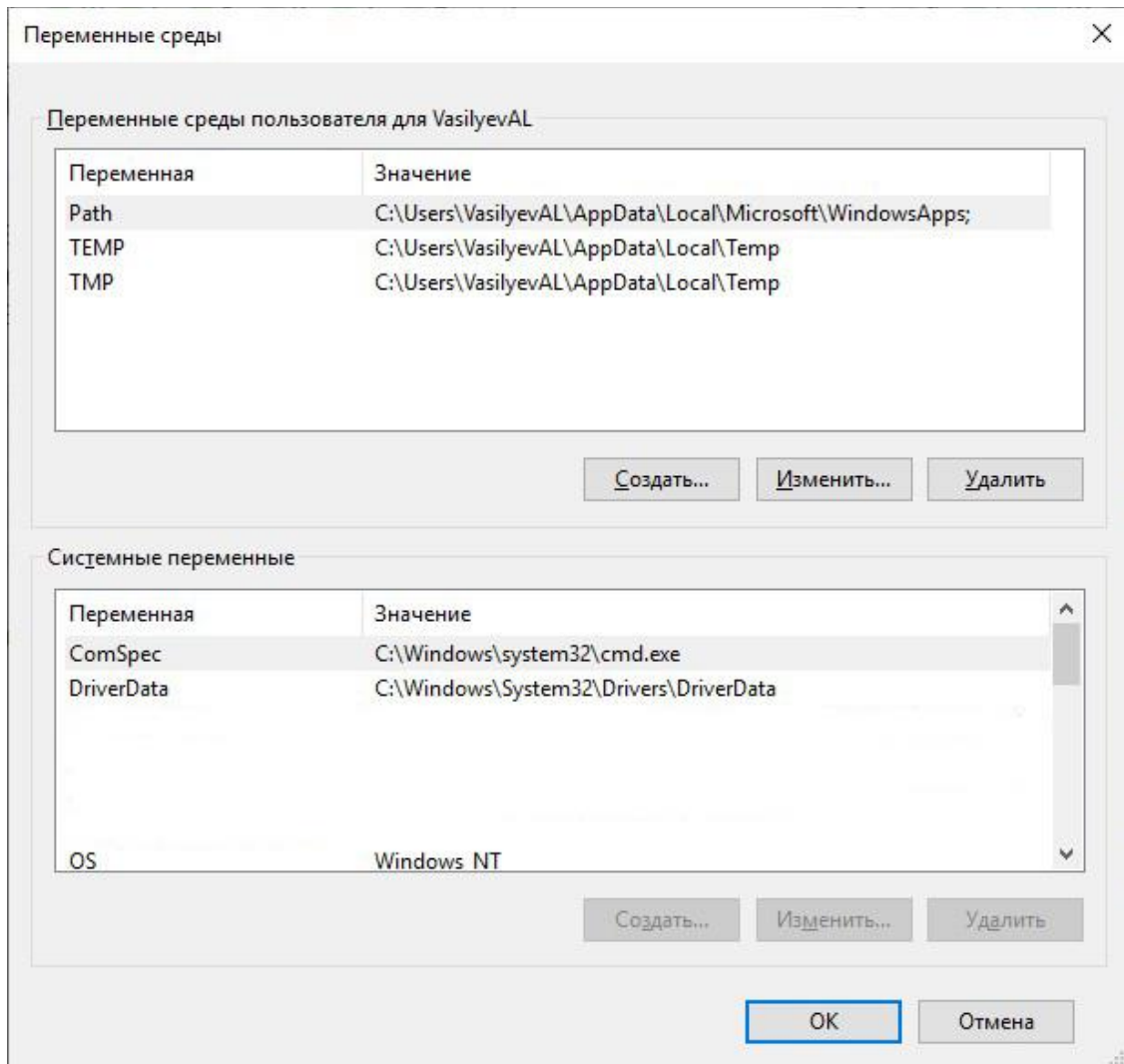


Рис. 25

В окне создания новой системной переменной в поле «Имя переменной» прописать значение «LLIC\_HOST», в поле «Значение переменной» прописать IP-адрес лицензионного сервера, как показано на рис. 26.

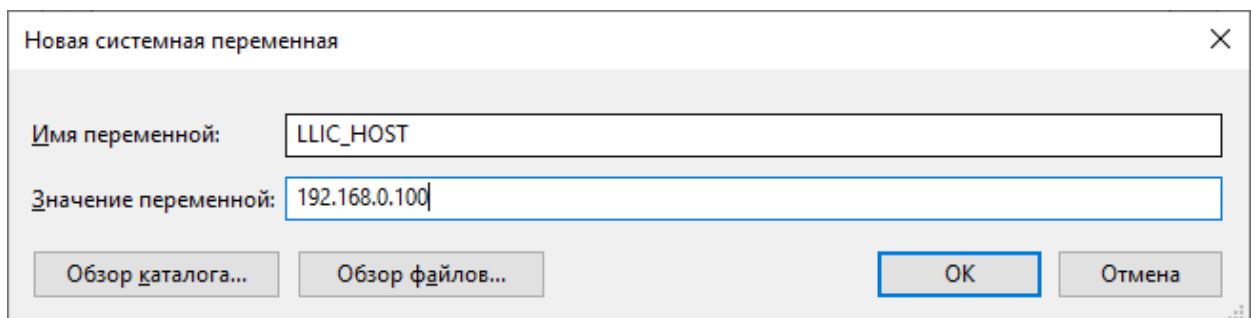


Рис. 26



Аналогичным образом можно создать переменную «LLIC\_PORT», в которой задается номер TCP-порта лицензионного сервера, если в этом есть необходимость.

Следующим шагом, учитывая реализованную в ПО «Расчет-ЭМВ» возможность проведения расчетов на ВВС (взаимодействия с удаленными решателями на ВВС), является процедура настройки аутентификации пользователя, от учетной записи которого будет выполняться расчет на ВВС по публичному ключу.

Для этого необходимо запустить командную оболочку ОС Windows сочетанием клавиш |Win|+г. В появившемся окне набрать имя файла командной оболочки cmd, как показано на рис. 27 и нажать кнопку «ОК».

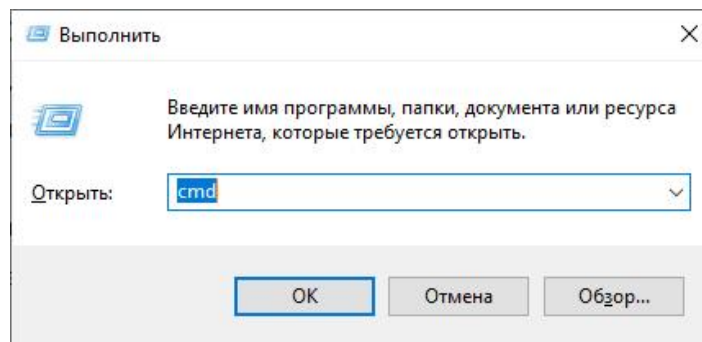


Рис. 27

В появившемся окне консоли нужно выполнить команду смены каталога на каталог установки ПО «Расчет-ЭМВ» (по умолчанию это C:\VNIITF\Rw\_Editor\), как показано на рис. 28.

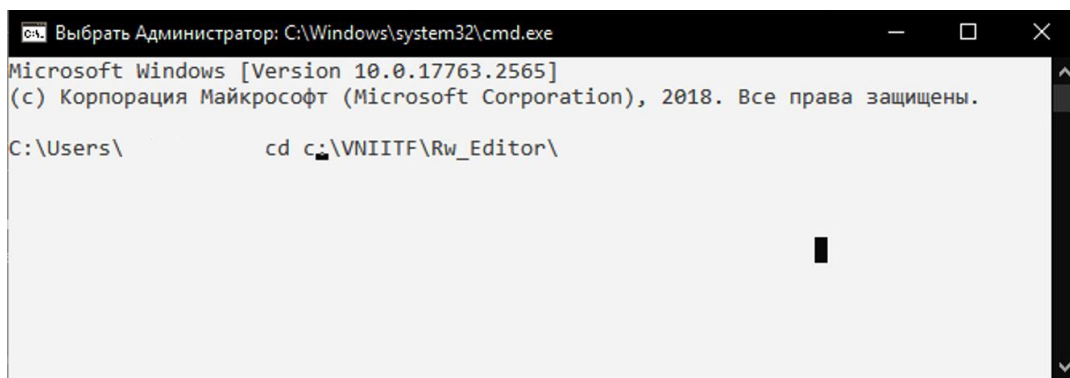


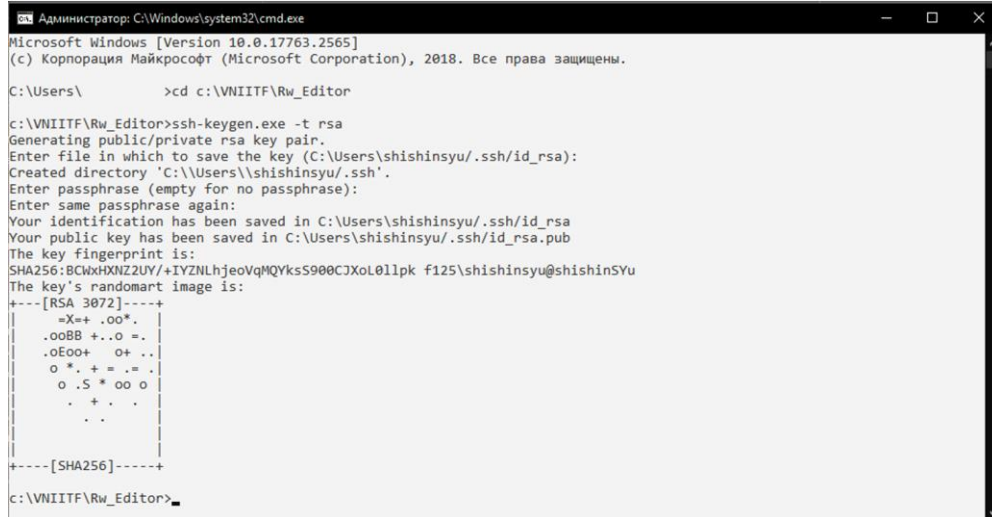
Рис. 28

После этого в командной строке необходимо создать новую пару ключей SSH выполнив команду `ssh-keygen -t rsa`, как показано на рис. 29, при этом на все запросы необходимо нажать *Enter*, после чего система сгенерирует пару ключей, которые по умолчанию сохраняются в:

C:\Users\username\.ssh\id\_rsa (приватный ключ),

C:\Users\username\.ssh\id\_rsa.pub (открытый ключ),

где username – имя пользователя.



```

Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.2565]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\ >cd c:\VNIITF\Rw_Editor

c:\VNIITF\Rw_Editor>ssh-keygen.exe -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\shishinsky\.ssh\id_rsa):
Created directory 'C:\Users\shishinsky\.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\shishinsky\.ssh\id_rsa
Your public key has been saved in C:\Users\shishinsky\.ssh\id_rsa.pub
The key fingerprint is:
SHA256:BCWxHXNZ2UY/+IYZNLhjeoVqMQYksS900CJXol01pk f125\shishinsky@shishinsky
The key's randomart image is:
+--[RSA 3072]-----+
  =X=+ .oo*.
  .ooBB +.o =.
  .oEoo+ o+ ..|
  o * . + = . .|
  o .S * oo o
  . + . .
  . .
+-----[SHA256]-----+
c:\VNIITF\Rw_Editor>

```

Рис. 29

Далее для настройки взаимодействия с удаленным сервером ВВС пользователю необходимо:

– создать на удаленном сервере (например, host) в домашнем каталоге пользователя (например, с учетной записью username) каталог, для этого выполнить команду (на запрос команды, ввести пароль доступа на удаленный сервер):

```
ssh username@host "mkdir -p ~/.ssh && chmod 700 ~/.ssh"
```

– скопировать открытый ключ на удаленный сервер, для этого выполнить команду (на запрос команды, ввести пароль доступа на удаленный сервер):

```
scp c:\Users\username\.ssh\id_rsa.pub username@host:~/.ssh
```

– выполнить команду (на запрос команды, ввести пароль доступа на удаленный сервер):

```
ssh username@host "cat ~/.ssh/id_rsa.pub | cat >> ~/.ssh/authorized_keys && chmod 600
~/.ssh/authorized_keys && rm -f ~/.ssh/id_rsa.pub"
```

– выполнить проверку, выполнив команду:

```
ssh username@host pwd
```

При этом выполнение команды должно пройти без ошибок (должен быть выведен путь к текущему каталогу пользователя на удаленном сервере).

Процедуру копирования публичного ключа в файл `authorized_keys` на удаленном сервере можно также провести с использованием любого графического FTP клиента (например, для того, чтобы не испортить существующий файл при его наличии).

### 3.2.2. Запуск программы

Запуск в графическом режиме происходит путем вызова программы по ярлыку RWEditor на рабочем столе пользователя, либо приложения RWEditor в рабочем каталоге установки. Процедура запуска графической оболочки RWEditor подробно во второй части руководства пользователя [1] и является общей для ОС Linux или ОС Windows.

Запуск в консольном режиме можно рассмотреть на примере подкаталога задания расчета модели исследуемого изделия. Для этого необходимо создать отдельный каталог, например, по имени модели и поместить туда решатели «Расчет-ММ» (EMPCore.exe) и «Расчет-МПЛ» (MTL.exe) вместе с исходным файлом задания расчета в формате sio как показано на рис. 30.

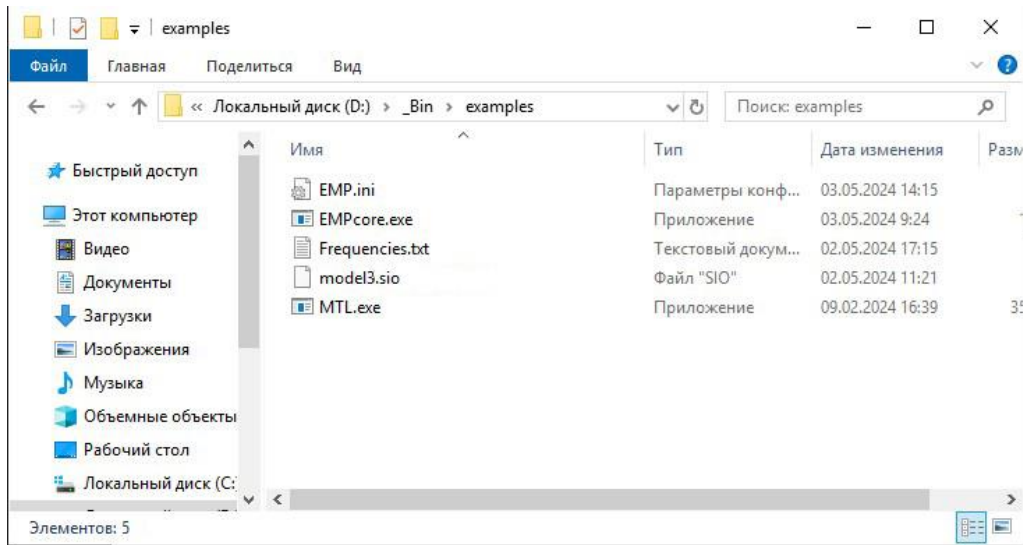


Рис. 30

Запуск расчетных модулей «Расчет-ММ» и «Расчет-МПЛ» в консольном режиме можно проводить как в окне проводника, так и из командной оболочки `cmd`, как по отдельности, так и последовательно, сформировав соответствующий `bat` файл. Пример содержимого указанного файла представлено на рис. 31.

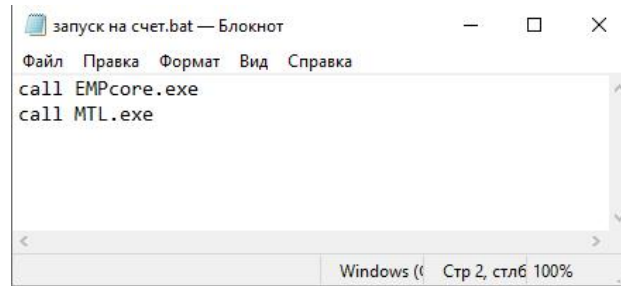


Рис. 31

ПО «Расчет-ЭМВ» позволяет использовать оптимизирующие библиотеки стороннего производителя для ускорения процесса расчета. Для этого используются бинарные версии решателей «Расчет-ММ» (EMPCore\_OPT) и «Расчет-МПЛ» (MTL\_OPT), входящие в состав дистрибутива ПО «Расчет-ЭМВ».

Пример использования оптимизированных решателей в консольном режиме представлен в каталоге examples/Model30\_OPT. Структура каталога расчета см. на рис. 4.

### 3.2.3. Выполнение программы

Выполнение расчетов с использованием графической оболочки RWEEditor в ОС Windows проводится в соответствии с описанием, изложенном во второй части руководства пользователя [1].

Порядок выполнения ПО «Расчет-ЭМВ» в ОС Windows в консольном режиме практически ничем не отличается от порядка ее выполнения в ОС на базе Linux. Пример окна вывода запущенного расчетного модуля «Расчет-ММ» представлен на рис. 32.

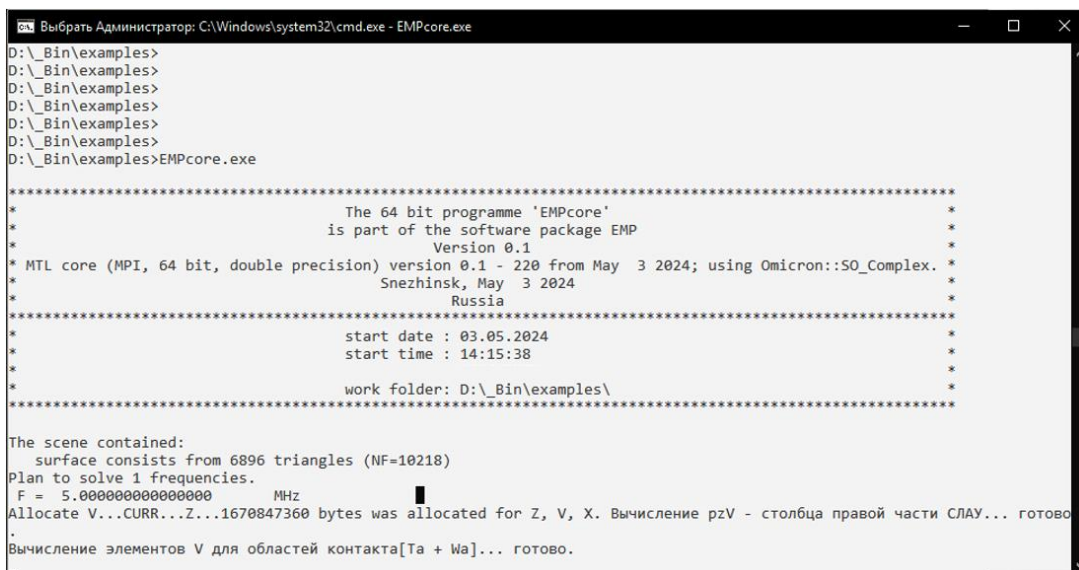


Рис. 32

#### 3.2.4. Завершение программы

Успешным признаком завершением счета программы является наличие файла Status.txt в расчетном каталоге, в котором содержится запись об отсутствии ошибок Error = 0. При отсутствии такого файла в расчетном каталоге пользователь должен проанализировать интерактивную консоль вывода на отсутствие ошибок.

В случае необходимости принудительного завершения счета при использовании консольного режима можно использовать сочетание клавиш |Ctrl|+|c|.

Завершение счета ПО «Расчет-ЭМВ» с использованием графической оболочки RWEEditor в ОС Windows проводится в соответствии с описанием, изложенном во второй части руководства пользователя [1].

#### 4. СООБЩЕНИЯ ПОЛЬЗОВАТЕЛЮ

В процессе выполнения ПО «Расчет-ЭМВ» может формировать несколько типов сообщений пользователю:

- сообщения, формируемые в процессе работы с графической оболочкой RWEEditor;
- сообщения, формируемые в ходе выполнения расчетов с использованием модулей «Расчет-МПЛ» и «Расчет-ММ».

Данные сообщения могут быть информационного и диагностического характера.

Сообщения, формируемые в процессе работы с графической оболочкой RWEEditor подробно описаны во второй части руководства пользователя [1].

Сообщения, формируемые в ходе выполнения расчетов с использованием модулей «Расчет-МПЛ», «Расчет-ММ», RWEEditor в свою очередь могут быть разделены на диагностические сообщения связанные с ошибками формирования расчетных данных, ошибками в вычислениях, ошибками формата чтения/записи формата SIO, ошибками при работе с лицензией.

##### 4.1. Ошибки формирования расчетных данных

Существует категория ошибок, которые могут быть вызваны неправильной подготовкой данных для расчета.

Примером ошибки такого рода является ошибка работы сеточного генератора «MeshRocket::Generate: Ошибка расчета сетки!» (рис. 33), которая возникает из-за нарушения логики работы его внутренних алгоритмов, к которой приводят ошибки геометрии тел, например, при наличии пересечений ребер грани, как показано на рис. 34.

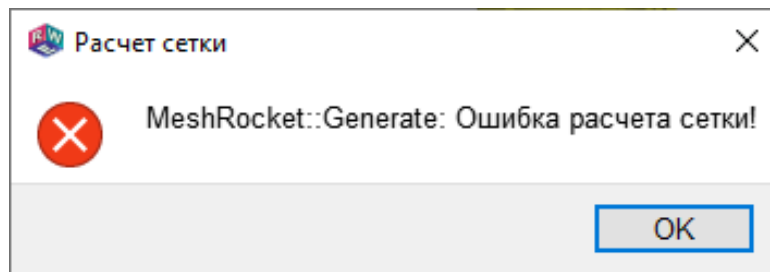


Рис. 33

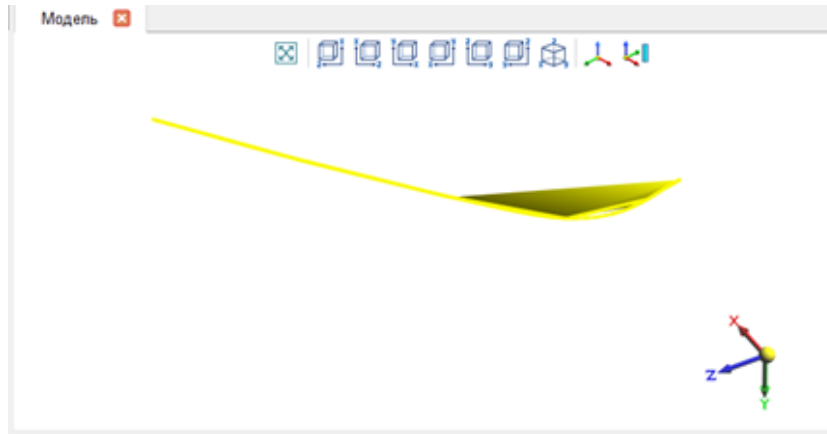


Рис. 34

Для отслеживания ошибок формирования геометрии тел используется инструмент выявления таких ошибок и оповещения пользователя. В инструменте реализованы методы обработки исключений сеточного генератора с последующей идентификацией геометрического объекта, для которого при формировании сетки выявлена ошибка. Полученные в результате работы инструмента данные (номер грани с ошибкой и номер тела, которому эта грань принадлежит) выводятся в виде сообщения в окне «Информация» программы «RWEditor», как показано на рис. 35.

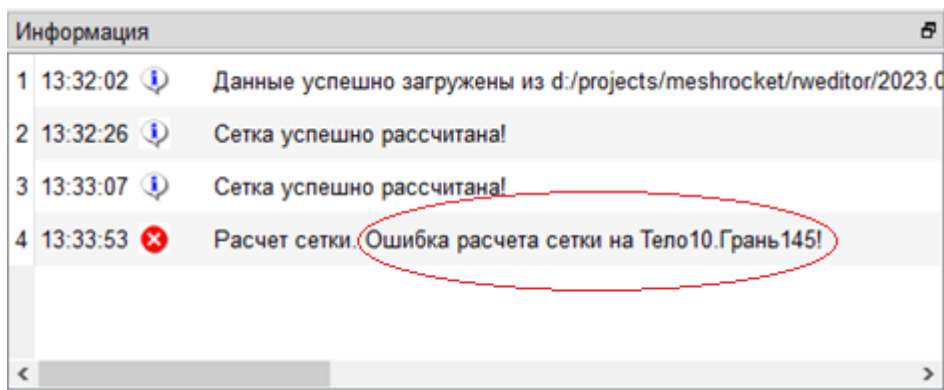


Рис. 35

#### 4.2. Ошибки в вычислениях

Категория ошибок, которые могут относиться к ошибкам вычислений:

- 1) ошибки в задании входных данных к расчету;
- 2) ошибки во входных данных к расчету;
- 3) ошибки при работе с памятью при выполнении расчета;
- 4) ошибки при работе с арифметикой при выполнении расчета;
- 5) ошибки, диагностируемые средой выполнения программы.

К первой категории относятся ошибки, допущенные расчетчиком при формировании начальных данных к расчету: неправильные имена файлов начальных данных, невозможность доступа к файлам начальных данных, неправильные имена расчетных модулей и т.д.

Ко второй категории относятся ошибки, связанные с неправильным содержимым используемых входных файлов: (файл испорчен или содержит неправильно подготовленные данные).

К третьей категории относятся ошибки при обращении к памяти, возникающие во время расчета. Возможными причинами являются использование в задаче количества расчетных элементов, для которых не хватает выделенной задаче на период расчета памяти.

К четвертой категории относятся ошибки, связанные с неправильным выполнением математических операций (деление на 0, вещественное переполнение, корень или экспонента от недопустимого аргумента). Такие ошибки могут возникнуть из-за неправильной подготовки входных данных для расчета, либо из-за наличия внутренних ошибок в программе.

Пятая категория ошибок может быть связана с отсутствием необходимых для выполнения расчета системных и прикладных компонент или неправильно заданными параметрами запуска расчета.

Ошибки первой категории фиксируются в выводных файлах расчета и должны исправляться расчетчиком.

Ошибки второй категории фиксируются в выводных файлах расчета. При их возникновении можно попробовать использовать другие файлы к данными. Однако, если файл с начальными данными готовится расчетчиком и его повторное форсирование не привело к устранению ошибки, необходимо связаться с разработчиком программы формирования данных, возможно в ней имеется ошибка.

Ошибки третьей и четвертой категории фиксируются в выводных файлах расчета, об них необходимо незамедлительно сообщать разработчикам с предъявлением им версии используемой программы и используемых входных данных.

Пятая категория ошибок приводит к невозможности запуска расчета. Для их устранения необходимо связаться с системным администратором.



### 4.3. Ошибки формата SIO

Все ошибки, возникающие при работе с форматом SIO, можно условно разделить на следующие группы:

- ошибки, связанные с параметрами;
- ошибки, связанные с выделением памяти;
- ошибки, связанные с работой с файлами;
- внутренние ошибки.

#### 4.3.1. Ошибки, связанные с параметрами

Эта группа ошибок возникает при неправильно переданных параметрах в функции SIO. Группа состоит из следующего набора ошибок:

- ошибка «передан пустой дескриптор файла» – возникает во всех функциях SIO, использующих дескриптор открытого файла, если вместо него передано пустое значение (например, файл не удалось открыть, но операции с содержимым файла выполняются);
- ошибка «последнее определение не завершено» – может возникнуть при обращении к функциям SIO, если предшествующее создание группы не завершено, т.е. определены не все поля группы;
- ошибка «нет имени в каталоге» – возникает в функциях чтения записей при попытке обращения к блоку данных с несуществующим в каталоге именем;
- ошибка «увеличение не последней записи» – возникает в функции `siorwr` при попытке записать в блок данных информацию большего объема, чем для него отведено в каталоге, и этот блок является не последним в файле;
- ошибка «выход за границу записи» – возникает в функции `siorrd` при попытке прочитать из блока данных информации больше, чем для него отведено в каталоге;
- ошибка «ошибка в формате» – возникает, когда формат запрашиваемой из блока данных информации не соответствует формату, зарегистрированному для этого блока данных в каталоге (может возникать как при чтении, так и при записи, если блок данных уже существует и формат известен);
- ошибка «группа уже определена» – возникает в случае определения группы с уже существующим именем;

- ошибка «группа не определена» – возникает при обращении к группе с именем, отсутствующим в каталоге, или указанное имя принадлежит не группе;
- ошибка «нет такого поля» – возникает при попытке чтения/записи экземпляра группы с указанием имени поля, которого нет в определении группы;
- ошибка «нет такой группы» – возникает при попытке чтения/записи экземпляра группы с указанием имени группы, которой нет в каталоге;
- ошибка «выход за границы поля» – возникает при операциях чтения/записи экземпляра группы, если переданный в операцию размер поля экземпляра группы превышает размер, указанный при определении поля в группе;
- ошибка «экземпляр группы должен быть > 0» – возникает при обращении на чтение с экземпляра группы (на запись в экземпляр группы) с отрицательным номером;
- ошибка «выход за границу записи» – возникает при операциях чтения/записи в случае, если запрашиваемая длина превышает описанную в каталоге длину записи;
- ошибка «массив не существует» – возникает при выполнении операций чтения, записи, запроса атрибутов для массива, имя которого в каталоге отсутствует, т.е. массив не был создан;
- ошибка «размерность вырезки > размерности массива» – возникает при работе с вырезками массива, для которых размерность (число измерений) превышает объявленную размерность массива;
- ошибка «ошибка в вырезке» – возникает при работе с вырезками массива в случае, если начальный индекс в массиве для вырезки по какому-то измерению не соответствует размеру массиву по данному измерению, при этом в параметре <уточнение> указывается номер измерения, заданный индекс и допустимые значения индекса;
- ошибка «ошибка в размерности» – возникает при работе с вырезками массива в случае, если указанная размерность вырезки меньше единицы или больше размерности массива;
- ошибка «Find in cat – Empty Name» – возникает в функциях для работы с блоками данных в случае, если переданное имя блока данных является пустым.

#### 4.3.2. Ошибки, связанные с выделением памяти

Эта группа ошибок возникает при нехватке оперативной памяти при использовании программного модуля или при неправильной работе с памятью в модуле (нарушение списочной структуры выделяемой модулю памяти). Группа состоит из следующего набора ошибок:

- ошибка «нет памяти для HF» – выдается в функции открытия файла в случае невозможности выделения памяти для заголовка файла в оперативной памяти;
- ошибка «нет памяти для каталогов» – выдается в функции открытия файла в случае невозможности выделения памяти для каталогов наборов данных в оперативной памяти.

#### 4.3.3. Ошибки, связанные с работой с файлами

Эта группа ошибок возникает при работе с объектами файловой системы. Группа состоит из следующего набора ошибок:

- ошибка «файл не существует» – выдается при попытке открытия несуществующего файла в режиме чтения, чтения/записи;
- ошибка «ошибка создания файла» – выдается при попытке открыть несуществующий файл в режиме записи, если указано недопустимое имя файла или у пользователя, запускающего программный модуль, обращающийся к функциям SIO, нет прав на создание файла в каталоге, в котором этом файл должен быть создан;
- ошибка «файл на RO не открыт» – выдается при попытке открыть существующий файл на чтение, но режим чтения для этого файла пользователю, запускающему программный модуль, не разрешен;
- ошибка «файл на RW не открыт» – выдается при попытке открыть существующий файл на запись, но режим записи для этого файла пользователю, запускающему программный модуль, не разрешен;
- ошибка «ошибка позиционирования» – возникает при доступе к блокам данных, содержащимся в файле, по адресам, указанным для этих блоков в каталоге файла, и свидетельствует о порче каталога файла;
- ошибка «ошибка fflush» – возникает при невозможности сбросов файловых кэшей на физическое устройство хранения при выполнении операций записи блоков

данных и свидетельствует о неправильной работе операционной системы, либо ее компонент;

– ошибка «ошибка закрытия» – возникает при невозможности закрытия файла, описываемого дескриптором файла (возможно, дескриптор файла испорчен);

– ошибка «старый формат не поддерживается» – в настоящее время чаще всего возникает, если на функцию `siofor` подать файл с нарушенной структурой каталога или файл другой структуры;

– ошибка «ошибка чтения/записи» – возникает при неправильном выполнении операций чтения/записи с содержимым файла и свидетельствует о неправильной работе операций чтения/записи в операционной системе или SIO.

#### 4.3.4. Внутренние ошибки

Существует еще одна категория ошибок, которые несколько раз возникали при проведении расчетов в модулях ПО «Расчет-ЭМВ» – это ошибки, маркируемые в файлах вывода как «internal». Эти ошибки свидетельствуют о неправильной работе SIO, которые требуют вмешательства разработчика ПО «Расчет-ЭМВ».

#### 4.4. Ошибки при работе с лицензией

При отсутствии лицензионного флеш-накопителя в порте ПЭВМ пользователя, при работе с RWEditor возникает ошибка, как показано на рис. 36.

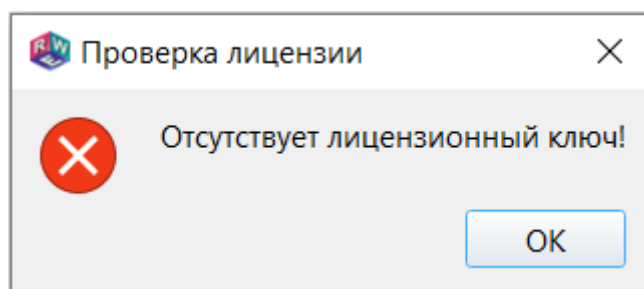


Рис. 36

В этом случае необходимо вставить в USB-порт ПЭВМ пользователя лицензионный флеш-накопитель, либо запросить новый у разработчика в случае его неисправности.

## ФАЙЛ INSTALL.SH

```

- #!/bin/bash
- DEFAULT_DIR=../lib
- NT=16
- CUDA=/opt/nvidia/hpc_sdk/Linux_x86_64/20.9/cuda/
- wd=`pwd`
- echo -n "Input install dir [$DEFAULT_DIR]: "
- read install_dir
- if [ -z $install_dir ]
- then
- install_dir=$DEFAULT_DIR
- fi
-
- mkdir -p $install_dir 2>/dev/null
- prefdir=`realpath $install_dir` # полный путь до каталога куда мы поставим
все пакеты
-
- echo -n "Input max thread for build packages [$NT]: "
- read nt
- if [ -z $nt ]
- then
- nt=$NT
- fi
- echo -n "Build gcc with nvidia support [Y/n]: "
- read gcc_nvptx
- if [ -z $gcc_nvptx ]
- then
- gcc_nvptx="yes"
- fi
- case "$gcc_nvptx" in
- *[yY]*|[yY][eE][sS]*)
- echo -n "Input cuda dir [$CUDA]: "
- read cuda
- if [ -z $cuda ]
- then
- cuda=$CUDA
- fi
- cd gcc
- ./inst.nvptx $nt $prefdir $cuda
- if [ $? -ne 1 ]
- then
- echo "---- GCC BUILD SUCCESS ----"
- gcc_dir="$prefdir/gcc"
- else
- echo "---- GCC BUILD FAIL ----"
- exit 1
-
- fi
- ;;
- *)
- echo -n "Build gcc without nvidia support [Y/n]: "
- read gcc_var
- if [ -z $gcc_var ]

```

07623974.37016-01 90 01-1

```

-     then
-     gcc_var="yes"
-     fi
-     case "$gcc_var" in
-     *[yY]*|[yY][eE][sS]*)
-     cd gcc
-     ./inst $nt $prefdir
-     if [ $? -ne 1 ]
-     then
-         echo "---- GCC BUILD SUCCESS ----"
-         gcc_dir="$prefdir/gcc"
-     else
-         echo "---- GCC BUILD FAIL ----"
-         exit 1
-     fi
-     ;;
-     esac
-     cd $wd
-     echo -n "Build cmake [Y/n]: "
-     read cmake_var
-     if [ -z $cmake_var ]
-     then
-         cmake_var="yes"
-     fi
-     case "$cmake_var" in
-     *[yY]*|[yY][eE][sS]*)
-     cd cmake
-     ./inst $nt $prefdir $gcc_dir
-         if [ $? -ne 1 ]
-         then
-             echo "---- CMAKE BUILD SUCCESS ----"
-         else
-             echo "---- CMAKE BUILD FAIL ----"
-             exit 1
-         fi
-     ;;
-     esac
-     cd $wd
-     echo -n "Build hdf5 [Y/n]: "
-     read hdf_var
-     if [ -z $hdf_var ]
-     then
-         hdf_var="yes"
-     fi
-     case "$hdf_var" in
-     *[yY]*|[yY][eE][sS]*)
-     cd hdf5
-     ./inst $nt $prefdir $gcc_dir
-         if [ $? -ne 1 ]
-         then
-             echo "---- HDF5 BUILD SUCCESS ----"
-         else
-             echo "---- HDF5 BUILD FAIL ----"
-             exit 1

```

07623974.37016-01 90 01-1

```
-     fi
-     ;;
- esac
- cd $wd
- echo -n "Build mvapich [Y/n]: "
- read mvapich_var
- if [ -z $mvapich_var ]
-     then
-         mvapich_var="yes"
-     fi
- case "$mvapich_var" in
-     *[yY]*|[yY][eE][sS]*)
-         cd mvapich
-         ./inst $nt $prefdir $gcc_dir
-         if [ $? -ne 1 ]
-             then
-                 echo "---- MVAPICH BUILD SUCCESS ----"
-             else
-                 echo "---- MVAPICH BUILD FAIL ----"
-                 exit 1
-             fi
-         fi
-     ;;
- esac
- cd $wd
- echo -n "Build qt (need c++ compiler with c++11 standart) [Y/n]: "
- read qt_var
- if [ -z $qt_var ]
-     then
-         qt_var="yes"
-     fi
- case "$qt_var" in
-     *[yY]*|[yY][eE][sS]*)
-         cd qt
-         ./inst $nt $prefdir $gcc_dir
-         if [ $? -ne 1 ]
-             then
-                 echo "---- QT BUILD SUCCESS ----"
-             else
-                 echo "---- QT BUILD FAIL ----"
-                 exit 1
-             fi
-         fi
-     ;;
- esac
```

## ФАЙЛ CMAKE/INST

```
- #!/bin/bash
- wd=`pwd`
- nm=$1
- install_dir=$2
- if [ $3 ]
-     then
-     gcc_dir=$3
-     export PATH=$gcc_dir/bin:$PATH
-     export LD_LIBRARY_PATH=$gcc_dir/lib64:$LD_LIBRARY_PATH
-     export INCLUDE=$INCLUDE:$gcc_dir/include
-     export CC=$gcc_dir/bin/gcc
-     export CXX=$gcc_dir/bin/g++
-     export FC=$gcc_dir/bin/gfortran
-     export F77=$gcc_dir/bin/gfortran
-     export F90=$gcc_dir/bin/gfortran
- fi
- if true
-     then
-     echo "==== CMAKE ====="
-     rm -rf $wd/cmake-3.20.1 2> /dev/null
-     tar xzf cmake-3.20.1.tar.gz
-     cd $wd/cmake-3.20.1
-     ./bootstrap --prefix=$install_dir/cmake > logstrap 2>&1
-     if [ $? -ne 0 ]; then
-     exit 1
-     fi
-     make -j $nm > logmake 2>&1
-     if [ $? -ne 0 ]; then
-     exit 1
-     fi
-     make install > logmakeinstall 2>&1
-     if [ $? -ne 0 ]; then
-     exit 1
-     fi
- fi
- exit 0
```



## ФАЙЛ GCC/INST

```

- #!/bin/sh
- nm=$1
- defdir=$2
- rm -rf $defdir/gcc 2>/dev/null
- mkdir -p $defdir/gcc 2>/dev/null
- PREFIX=$defdir/gcc
- export LD_LIBRARY_PATH=$PREFIX/lib64:$LD_LIBRARY_PATH
- export LDFLAGS=-L$PREFIX/lib64
- export PATH=$PREFIX/bin:$PATH
- wd=`pwd`
- echo "==== GMP ====="
- rm -rf gmp-6.2.1 2>/dev/null
- tar xaf gmp-6.2.1.tar.bz2
- cd gmp-6.2.1
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --enable-cxx --with-pic
> logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== ISL ====="
- rm -rf isl-0.25 2>/dev/null
- tar xaf isl-0.25.tar.gz
- cd isl-0.25
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --with-gmp-
prefix=$PREFIX --with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== MPFR ====="
- rm -rf mpfr-4.1.0 2>/dev/null
- tar xaf mpfr-4.1.0.tar.gz
- cd mpfr-4.1.0
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --with-gmp-
include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 --with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1

```

07623974.37016-01 90 01-1

```

- fi
- cd $wd
- echo "==== MPC ====="
- rm -rf mpc-1.2.1 2>/dev/null
- tar xaf mpc-1.2.1.tar.gz
- cd mpc-1.2.1
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
- --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
- --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 --with-
pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== BINUTILS ====="
- rm -rf binutils-2.38 2>/dev/null
- tar xaf binutils-2.38.tar.xz
- cd binutils-2.38
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
- --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
- --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 \
- --with-mpc-include=$PREFIX/include --with-mpc-lib=$PREFIX/lib64 \
- --with-isl-include=$PREFIX/include --with-isl-lib=$PREFIX/lib64 --with-pic
> logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== GCC ====="
- rm -rf gcc-11.2.0 2>/dev/null
- tar xaf gcc-11.2.0.tar.xz
- rm -rf build-host-gcc 2>/dev/null
- mkdir build-host-gcc 2>/dev/null
- cd build-host-gcc
- ../gcc-11.2.0/configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
- --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
- --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 \
- --with-mpc-include=$PREFIX/include --with-mpc-lib=$PREFIX/lib64 \
- --with-isl-include=$PREFIX/include --with-isl-lib=$PREFIX/lib64 \
- --disable-multilib --enable-languages=c,c++,fortran,lto --enable-lto --
with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then

```

07623974.37016-01 90 01-1

```
- exit 1
- fi
- make -j $nm > logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- exit 0
```

## ФАЙЛ GCC/INST.NVPTX

```

- #!/bin/sh
-
- nm=$1
- defdir=$2
- CUDA=$3
-
- rm -rf $defdir/gcc 2>/dev/null
- mkdir -p $defdir/gcc 2>/dev/null
- PREFIX=$defdir/gcc
-
- export LD_LIBRARY_PATH=$PREFIX/lib64:$LD_LIBRARY_PATH
- export LDFLAGS=-L$PREFIX/lib64
- export PATH=$PREFIX/bin:$PATH
- wd=`pwd`
- echo "==== GMP ====="
- rm -rf gmp-6.2.1 2>/dev/null
- tar xaf gmp-6.2.1.tar.bz2
- cd gmp-6.2.1
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --enable-cxx --with-pic
> logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== ISL ====="
- rm -rf isl-0.25 2>/dev/null
- tar xaf isl-0.25.tar.gz
- cd isl-0.25
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --with-gmp-
prefix=$PREFIX --with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== MPFR ====="
- rm -rf mpfr-4.1.0 2>/dev/null
- tar xaf mpfr-4.1.0.tar.gz
- cd mpfr-4.1.0
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 --with-gmp-
include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 --with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi

```

07623974.37016-01 90 01-1

```

- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== MPC ====="
- rm -rf mpc-1.2.1 2>/dev/null
- tar xaf mpc-1.2.1.tar.gz
- cd mpc-1.2.1
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
-   --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
-   --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 --with-
pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== BINUTILS ====="
- rm -rf binutils-2.38 2>/dev/null
- tar xaf binutils-2.38.tar.xz
- cd binutils-2.38
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
-   --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
-   --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 \
-   --with-mpc-include=$PREFIX/include --with-mpc-lib=$PREFIX/lib64 \
-   --with-isl-include=$PREFIX/include --with-isl-lib=$PREFIX/lib64 --with-pic
> logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== NVPTX-TOOLS ====="
- rm -rf nvptx-tools-master 2>/dev/null
- tar xzf nvptx-tools-master.tar.gz
- cd nvptx-tools-master
- ./configure --prefix=$PREFIX --libdir=$PREFIX/lib64 > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi

```

07623974.37016-01 90 01-1

```

- cd $wd
- echo "==== NEWLIB ====="
- rm -rf newlib-master 2>/dev/null
- tar xzf newlib-master.tar.gz
- cd $wd
- echo "==== GCC-NVPTX ====="
- rm -rf gcc-11.2.0 2>/dev/null
- tar xaf gcc-11.2.0.tar.xz
- cd gcc-11.2.0
- ln -s ../newlib-master/newlib newlib 2>/dev/null
- cd $wd
- rm -rf build-nvptx-gcc 2>/dev/null
- mkdir build-nvptx-gcc 2>/dev/null
- cd build-nvptx-gcc
- export target=$(../gcc-11.2.0/config.guess)
- ../gcc-11.2.0/configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
-   --target=nvptx-none --with-build-time-tools=$PREFIX/nvptx-none/bin \
-   --enable-as-accelerator-for=$target --disable-sjlj-exceptions --enable-
newlib-io-long-long \
-   --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
-   --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 \
-   --with-mpc-include=$PREFIX/include --with-mpc-lib=$PREFIX/lib64 \
-   --with-isl-include=$PREFIX/include --with-isl-lib=$PREFIX/lib64 \
-   --enable-languages=c,c++,fortran,lto --enable-shared > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- cd $wd
- echo "==== GCC ====="
- rm -rf build-host-gcc 2>/dev/null
- mkdir build-host-gcc 2>/dev/null
- cd build-host-gcc
- ../gcc-11.2.0/configure --prefix=$PREFIX --libdir=$PREFIX/lib64 \
-   --with-gmp-include=$PREFIX/include --with-gmp-lib=$PREFIX/lib64 \
-   --with-mpfr-include=$PREFIX/include --with-mpfr-lib=$PREFIX/lib64 \
-   --with-mpc-include=$PREFIX/include --with-mpc-lib=$PREFIX/lib64 \
-   --with-isl-include=$PREFIX/include --with-isl-lib=$PREFIX/lib64 \
-   --enable-offload-targets=nvptx-none --with-cuda-driver-
include=$CUDA/include/ --with-cuda-driver-lib=$CUDA/lib64/ \
-   --disable-multilib --enable-languages=c,c++,fortran,lto --enable-lto --
with-pic > logconf 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- make -j $nm > logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1

```

07623974.37016-01 90 01-1

```
- fi
- make install >> logmake 2>&1
- if [ $? -ne 0 ]; then
-   exit 1
- fi
- exit 0
```

## ФАЙЛ HDF5/INST

```

- #!/bin/bash
- wd=`pwd`
- nm=$1
- install_dir=$2
- if [ $3 ]
-   then
-     gcc_dir=$3
-     export PATH=$gcc_dir/bin:$PATH
-     export LD_LIBRARY_PATH=$gcc_dir/lib64:$LD_LIBRARY_PATH
-     export INCLUDE=$INCLUDE:$gcc_dir/include
-     export CC=$gcc_dir/bin/gcc
-     export CXX=$gcc_dir/bin/g++
-     export FC=$gcc_dir/bin/gfortran
-     export F77=$gcc_dir/bin/gfortran
-     export F90=$gcc_dir/bin/gfortran
-   fi
- if true
-   then
-     echo "==== HDF5 ====="
-     rm -rf hdf5-1.10.5 2>/dev/null
-     tar xzf hdf5-1.10.5.tar.gz
-     cd hdf5-1.10.5
-     ./configure --prefix=$install_dir/hdf5 --enable-cxx --disable-libtool-
lock > logconf 2>&1
-     if [ $? -ne 0 ]; then
-       exit 1
-     fi
-     make -j $nm > logmake 2>&1
-     if [ $? -ne 0 ]; then
-       exit 1
-     fi
-     make install > logmakeinstall 2>&1
-     if [ $? -ne 0 ]; then
-       exit 1
-     fi
-   fi
- fi
- exit 0

```



## ФАЙЛ MVARICH/INST

```

- #!/bin/sh
- wd=`pwd`
- nm=$1
- install_dir=$2
- if [ $3 ]
-     then
-         gcc_dir=$3
-         export PATH=$gcc_dir/bin:$PATH
-         export LD_LIBRARY_PATH=$gcc_dir/lib64:$LD_LIBRARY_PATH
-         export INCLUDE=$INCLUDE:$gcc_dir/include
-         export CC=$gcc_dir/bin/gcc
-         export CXX=$gcc_dir/bin/g++
-         export FC=$gcc_dir/bin/gfortran
-         export F77=$gcc_dir/bin/gfortran
-         export F90=$gcc_dir/bin/gfortran
-         export FCFLAGS="-fallow-argument-mismatch"
-         export FFLAGS="-fallow-argument-mismatch"
-     fi
- CONF_FLAGS="--enable-sharedlibs=gcc \
- --enable-g=dbg \
- --enable-cxx \
- --enable-fortran=yes \
- --enable-wrapper-rpath \
- --enable-header-caching \
- --enable-registration-cache \
- --with-ch3-rank-bits=32 \
- --with-cluster-size=large \
- --enable-threads=multiple \
- --enable-rdma-cm=no \
- --with-ib-include=/ust/include/rdma \
- --enable-romio --with-file-system=lustre --disable-fast"
-
- export OPT_FLAG=""
- CONF_FLAGS="$${CONF_FLAGS} --enable-cuda=no"
- CONF_FLAGS="$CONF_FLAGS --enable-shared --enable-debuginfo"
- export MPICHLIB_CXXFLAGS="$MPICHLIB_CXXFLAGS $OPT_FLAG"
- export MPICHLIB_CFLAGS="$MPICHLIB_CFLAGS $OPT_FLAG"
- export MPICHLIB_FFLAGS="$MPICHLIB_FFLAGS $OPT_FLAG"
- export MPICHLIB_FCFLAGS="$MPICHLIB_FCFLAGS $OPT_FLAG"
- export MPICHLIB_CPPFLAGS="-DENABLE_QOS_SUPPORT"
- unset F90
- if true
-     then
-         echo "==== MVAPICH ====="
-         rm -rf mvapich2-2.3.6 2>/dev/null
-         tar xzf mvapich2-2.3.6.tar.gz
-         cd mvapich2-2.3.6
-         ./configure --prefix=$install_dir/mvapich $CONF_FLAGS > logconf 2>&1
-         if [ $? -ne 0 ]; then
-             exit 1
-         fi
-     make -j $nm > logmake 2>&1

```

07623974.37016-01 90 01-1

```
-   if [ $? -ne 0 ]; then
-   exit 1
-   fi
-   make install > logmakeinstall 2>&1
-   if [ $? -ne 0 ]; then
-   exit 1
-   fi
- fi
- exit 0
```

## ФАЙЛ QT/INST

```
- #!/bin/bash
- wd=`pwd`
- nm=$1
- install_dir=$2
- if [ $3 ]
- then
- gcc_dir=$3
- export PATH=$gcc_dir/bin:$PATH
- export LD_LIBRARY_PATH=$gcc_dir/lib64:$LD_LIBRARY_PATH
- export INCLUDE=$INCLUDE:$gcc_dir/include
- export CC=$gcc_dir/bin/gcc
- export CXX=$gcc_dir/bin/g++
- export FC=$gcc_dir/bin/gfortran
- export F77=$gcc_dir/bin/gfortran
- export F90=$gcc_dir/bin/gfortran
- fi
- if true
- then
- echo "==== QT5 ====="
- rm -rf qt-everywhere-src-5.15.2 2>/dev/null
- tar xaf qt-everywhere-src-5.15.2.tar.xz
- cd $wd/qt-everywhere-src-5.15.2
- ./configure -prefix $install_dir/qt \
- -opensource \
- -release \
- -no-dbus\
- -widgets\
- -gui \
- -no-openssl \
- -confirm-license \
- -nomake tests \
- -nomake examples \
- -skip qtwayland > logconf 2>&1
-
- if [ $? -ne 0 ]; then
- exit 1
- fi
- gmake -j $nm > logmake 2>&1
- if [ $? -ne 0 ]; then
- exit 1
- fi
- gmake install > logmakeinstall 2>&1
- if [ $? -ne 0 ]; then
- exit 1
- fi
- fi
- exit 0
```

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

|      |   |
|------|---|
| БКС  | – бортовая кабельная сеть   |
| ВВС  | – высокопроизводительная вычислительная система   |
| ГОСТ | – государственный стандарт  |
| ЕСПД | – единая система программной документации   |
| ОС   | – операционная система  |
| ПО   | – программное обеспечение   |
| ПЭВМ | – персональная электронно-вычислительная машина   |
| ПО   | – системное программное обеспечение   |
| СТО  | – сложных технических объектов  |
| CPU  | – central processing unit, центральный процессор  |
| GPU  | – graphics processing unit, графический процессор   |
| MPI  | – message passing interface, программный интерфейс для передачи сообщений между процессами одной задачи |

## ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

- 1 07623974.37016-01 90 01-2 Специализированное программное обеспечение «Расчет-ЭМВ». Руководство пользователя. Часть 2.
- 2 07623974.37016-01 13 01 Специализированное программное обеспечение «Расчет-ЭМВ». Описание программы.
- 3 Slurm workload manager. [Электронный ресурс]. – <https://slurm.schedmd.com/documentation.html>.

