

# Minichem 0.1

*Руководство пользователя*

А. В. Олейниченко

18 октября 2016 года



# Глава 1

## Общие сведения

Данный документ является подробным руководством пользователя к квантовохимической программе Minichem версии 0.1. Программа является учебной, поэтому может содержать ошибки и работать продолжительное время даже на небольших задачах.

### 1.1 Возможности программы

Доступен расчет энергии в точке методами RHF и UHF.



## Глава 2

# Установка и запуск

### 2.1 Установка

Программа Minichem устанавливается из исходных кодов, доступных на GitHub по адресу <https://github.com/aoleynichenko/minichem>. Для сборки требуются установленные библиотеки LAPACK, BLAS, OpenMP. Детали их установки зависят от дистрибутива и здесь не приводятся. На текущий момент мейкфайлы написаны с учетом специфики ОС Linux. Команды для сборки Minichem:

```
cd \${MINICHEM_HOME}/src
make
make install
```

Исполняемый файл Minichem появится в `MINICHEM_HOME/bin`.

### 2.2 Запуск

Minichem – консольное приложение. Формат команды запуска:

```
minichem <options> <input-files>
```

Удобно добавить директорию `MINICHEM_HOME/bin` в `PATH`.

По умолчанию вывод результатов осуществляется в стандартный поток вывода `stdout`.

### 2.3 Диагностика ошибок. Лог

Кроме основного вывода, Minichem также ведет лог (по умолчанию – `minichem.log`), в который записываются сообщения о ходе исполнения программы. Лог содержит развернутые сообщения о произошедших ошибках

и предупреждениях и может оказаться полезным при отладке как самого Minichem, так и input-скриптов.

## Глава 3

# Описание формата входных файлов

### 3.1 Язык Minichem

Для входных файлов Minichem разрабатывается специальный скриптовый язык программирования, ориентированный на применение в вычислительной химии и наследующий черты языков Psithon, NWChem и JavaScript.

Язык Minichem чувствителен к регистру (кроме названий базисных наборов и символов химических элементов). Символы конца строки и лишние пробельные символы игнорируются, что в принципе позволяет записать весь входной файл в одну строчку. Это возможно благодаря тому, что каждый оператор языка «знает», сколько аргументов должен принять.

### 3.2 Типы данных

**Molecule.** Объекты типа Molecule содержат информацию о заряде молекулы, мультиплетности основного состояния и координатах атомов. Для создания объекта Molecule используется ключевое слово `mol`:

```
mol LiH {  
  mult=1  
  charge=0  
  units=atomic  
  H      0.0000    0.0000    0.0000  
  Li     1.4632    0.0000    0.0000  
}
```

Обязательными в определении молекулы являются только ключевое слово `mol` и координаты атомов в формате хуз. Параметры, указываемые через знак равенства (`mult`, `charge`, `units`...) необязательны. После ключевого слова

`mol` опционально указывается имя молекулы; если оно не указано, объявленная молекула становится текущей (доступной для квантовохимических подпрограмм). Пример объявления безымянной молекулы:

```
mol {  
  C      0.0000    0.0000    0.0000  
  H      0.6281    0.6281   -0.6281  
  H     -0.6281   -0.6281   -0.6281  
  H      0.6281   -0.6281    0.6281  
  H     -0.6281    0.6281    0.6281  
}
```