

Компьютерные технологии в научных исследованиях и образовании

Юрий Анатольевич Богоявленский, заведующий кафедрой Информатики и математического обеспечения, к.т.н., доцент, [ybgv](http://ybgv.ru)

Ввод и вывод

Простой ввод-вывод файлов

Функция save — запись в файл

Функция load — чтение из файла

Функции fileread и fdisp

Функция save — запись в файл

Функции save и load позволяют записывать данные в файлы на диске и считывать их файлов в различных форматах. Форматом по умолчанию файлов, записанных функцией save, можно управлять с помощью функций save_default_options и save_precision (см. ниже).

Простой пример.

```
ybgv@ybgv-home:~> octave -q -p ~/MyOct
octave:1>
octave:1> cd MyOct/
octave:2> dir
.          avg.m          frmt.m          scr_exm.m
..         count_calls.m  html            scvmax.m
al.m      f.m              newwhos.m       vmax.m
octave:3> A = [pi,e;6.2e23,6]
A =

    3.1416e+00    2.7183e+00
    6.2000e+23    6.0000e+00
```

```
octave:4> save Mtr.A A
octave:5> dir
.          avg.m          html          vmax.m
..         count_calls.m newwhos.m
Mtr.A     f.m           scr_exm.m
al.m     frmt.m        scvmax.m
```

```
octave:6>
```

```
octave:6> exit
```

```
ybgv@ybgv-home:~> octave -q -p ~/MyOct
```

```
octave:1>
```

```
octave:1> A
```

```
error: 'A' undefined near line 1, column 1
```

```
octave:2> cd MyOct/
```

```
octave:3>
```

```
octave:3> load Mtr.A
```

```
octave:4> A
```

```
A =
```

```
3.1416e+00  2.7183e+00
```

```
6.2000e+23  6.0000e+00
```

```
octave:5>
```

Файл на диске представлен в текстовом виде:

```
# Created by Octave 7.1.0, Mon Nov 07 16:28:49 2022 MSK
<ybgv@ybgv-home>
# name: A
# type: matrix
# rows: 2
# columns: 2
3.1415926535897931 2.7182818284590451
```

6.1999999999999994e+23 6

Функция `save` может быть вызвана в следующих формах:

```
save file
save options file
save file v1 v2 . . .
save options file v1 v2 . . .
save ("-", "v1", "v2", . . . )
```

Отметим, что для всех форм кроме последней фактические параметры функции можно указывать без заключения в скобки, разделяя их пробелами. Мы не рассматриваем еще две формы этой функции, связанные со определенными в octave структурами данных. Параметр `options` определяет формат записи.

Функции в форме `save file` и `save options file` запишут в файл все переменные текущей области видимости.

Функции в форме `save file v1 v2 . . .` и `save options file v1 v2 . . .` запишут в файл переменные из списка `v1 v2 . . .`.

Функция в форме `save ("-", "v1", "v2", . . .)` возвращает значение типа строка, содержащее текст, подлежащий записи.

Допустимые значения параметра `options` представлены в следующей таблице. Если параметр опущен, то он, по умолчанию, имеет значение `text` и запись происходит в текстовом формате текущей кодировки ОС. Значение этого параметра, изменяющие формат записи, переопределяют формат, заданный функцией `save_default_options`.

| Значение параметра | Описание |
|--------------------|----------|
|--------------------|----------|

| | |
|----------------|---|
| -append | Добавить данные в конец файла, а не перезаписывать на место предыдущей записи. |
| -ascii | Запись только значений данных без заголовка и любой другой информации. Размерность переменных должна быть не более двух. Для комплексных значений будут записаны только действительные части как числа с плавающей точкой одинарной точности и разделены символами пробела. Для значения <code>-ascii</code> предусмотрены два подзначения: <ul style="list-style-type: none"> -double — записывать значения двойной точности. -tabs — разделять значения символами табуляции. |
| -binary | Запись в двоичном формате <code>octave</code> . |
| -float-binary | Запись в двоичном формате <code>octave</code> , но только значения будут записаны как числа с плавающей точкой одинарной точности. Все значения должны допускать такое представление. |
| -hdf5 | Запись в формате HDF5 Национального центра суперкомпьютерных приложений при Университете Иллинойса. Этот формат доступен только в том случае, если <code>octave</code> была собрана со ссылкой на библиотеки <code>hdf5</code> . |
| -float-hdf5 | Запись в формате HDF5 но только значения будут выведены как числа с плавающей точкой одинарной точности. Все значения должны допускать такое представление. |
| -text | Запись в текстовом формате текущей кодировки ОС (принята по умолчанию). |
| -v7.3 -V7.3 | Не реализованный в текущей версии Octave двоичный формат данных <code>matlab</code> версии 7.3. |

| | |
|--|---|
| -7.3 | |
| -v7 -V7 -7 -mat7-binary | Запись данных в двоичном формате matlab v7. |
| -v6 -V6 -6 -mat -matbinary | Запись данных в двоичном формате matlab v6. |
| -v4 -V4 -4 -mat4-binary | Запись данных в двоичном формате matlab v4. |
| -zip -z | Используется алгоритм gzip для сжатия файла. Это работает с файлами, сжатыми с помощью gzip вне octave. Эта опция доступна только если octave была создана со ссылкой на библиотеки zlib. |

В списке сохраняемых переменных для задания их имен могут использоваться шаблоны поиска, задаваемые следующими специальными символами:

? - совпадение с любым одиночным символом.

* - совпадение с любым, в том числе нулевым количеством символов.

[список_симвлов] - совпадение с любым символом из списка. Если первым символом в списке является «!» или «^», то ищется совпадение с символам, не входящими в список. Например шаблон [a-zA-Z] задает совпадение со всем строчным и прописным символами букв латинского алфавита.

Для всех значений параметра `options` функции `save` кроме значения `-ascii` статус глобальных переменных сохраняется и позже восстанавливается при выполнении функции `load`.

Пример:

Вызов

```
save -binary data a b*
```

запишет в файл `data` значение переменной «а» и всех переменных, имена которых начинаются с символа «b».

Есть три функции, которые изменяют поведение функции `save`:

- `save_default_options`
 - значение по умолчанию - `-text`, допустимые значения параметров см. в таблице выше (значения параметра `option` функции `save`)
- `save_precision`
 - значение по умолчанию — 17, допустимое значение — количество цифр, подлежащие записи в файл в текстовом формате. Для записи без потерь числа двойной точности в формате IEEE-754 минимально необходимое значение этого параметра — 17, одинарной точности — 9. Для больших файлов целесообразно использовать формат `-binary`.
- `save_header_format_string`
 - значение по умолчанию — строка текстового комментария о файле, которая записывается в него вместе с данными. Строка имеет формат:

```
"# Created by Octave VERSION, %a %b %d %H:%M:%S %Y %Z  
<USER@HOST>"
```

и содержит начинающиеся с символа «%» управляющие конструкции функции `strftime`. После обработки этой функцией данные о времени записи и владельце файла получают конкретные значения и строка может иметь, например, такой вид:

```
# Created by Octave 7.1.0, Mon Nov 07 16:28:49 2022  
MSK <ybgv@ybgv-home>.
```

Фактический параметр задает новое значение текстового комментария в записываемом файле. Оно должно иметь строковый тип, первым в строке должен быть символ «#». Это значение передается функции `strftime` и может содержать ее управляющие конструкции. Если значение параметра — пустая строка, то при записи файла текстовый комментарий будет опущен.

Функция `load` — чтение из файла

Функция `load` может быть вызвана в следующих формах:

```
load file
```

```
load options file
```

```
load options file v1 v2 . . .
```

```
S = load ("options", "file", "v1", "v2", . . . )
```

Причем в трех последних формах параметры `file` и `options` можно менять местами. Отметим, что для всех форм кроме последней фактические параметры функции можно указывать без заключения в скобки, разделяя их пробелами.

Функция загружает из файла `file` значения переменных, указанные в списке `v1 v2 . . .`. Для задания их имен могут использоваться те же конструкции шаблонов поиска, что и для функции `save`. Если имена переменных в памяти не определены, то они определяются. Если список не указан, то загружаются значения всех переменных, записанных в файле. Формат данных в файл определяется автоматически, но может быть переопределен с помощью параметра `options`.

Если значение переменной, которая не помечена в файле как глобальная, загружается из файла, и при этом уже существует глобально определенное имя, совпадающее с именем загружаемой переменной, то значение из файла присваивается этой глобальной переменной.

Если же значение переменной, которая помечена в файле как глобальная, загружается из файла, и при этом уже существует локально определенное имя, совпадающее с именем загружаемой переменной, то это локальное имя получает статус глобального.

Таким образом глобальный статус при загрузке из файлов имеет приоритет выше локального.

При вызове вида:

```
S = load ("options", "file", "v1", "v2", . . . )
```

имена переменных в памяти не определяются и значения им не присваиваются. Левая часть получает строковое значение, представляющее данные, содержащиеся в файле.

Функция `load` может загружать данные, хранящиеся в текстовом и двоичном форматах `octave`, а также в двоичном формате `matlab`. Если Octave скомпилирована с поддержкой `zlib`, она также может загружать файлы, сжатые утилитой `gzip`, автоматически определяя тип файла и выполняя преобразование из различных форматов с плавающей запятой (в настоящее время (ноябрь 2022 г.) поддерживаются только IEEE-754 big и little endian).

Допустимые значения параметра `options` представлены в следующей таблице.

| Значение параметра | Описание |
|---------------------|---|
| <code>-force</code> | Включен для обратной совместимости, игнорируется. В текущей версии значения переменных в памяти заменяются на одноименные значения из файлов. |
| <code>-ascii</code> | Активирует предположение, что файл содержит только |

| | |
|--|---|
| | столбцы числовых значений в текстовом формате без заголовка и другой информации. Данные будут загружены в виде матрицы с элементами числового типа, имя которой будет сформировано на основе имени файла. |
| <code>-binary</code> | Активирует предположение, что файл записан в двоичном формате <code>octave</code> . |
| <code>-hdf5</code> | Активирует предположение, что файл записан в формате HDF5 Национального центра суперкомпьютерных приложений при Университете Иллинойса. Отметим, что <code>octave</code> может читать такие файлы только если они записаны ею же с помощью функции <code>save</code> . Этот формат доступен только в том случае, если <code>octave</code> была собрана со ссылкой на библиотеки <code>hdf5</code> . |
| <code>-import</code> | Включен для обратной совместимости, игнорируется. Сейчас <code>octave</code> поддерживает многомерные данные формата HDF5 и автоматически модифицирует имена переменных, если они не являются корректными идентификаторами <code>octave</code> . |
| <code>-text</code> | Активирует предположение, что файл записан в текстовом формате <code>octave</code> . |
| <code>-v7.3</code> <code>-V7.3</code> <code>-7.3</code> | Не реализованный в текущей версии Octave двоичный формат данных <code>matlab</code> версии 7.3. |
| <code>-v7</code> <code>-V7</code> <code>-7</code> <code>-mat7-binary</code> | Активирует предположение, что файл записан в двоичном формате <code>matlab v7</code> . |

| | |
|--|---|
| -v6 -V6 -6 -mat -matbinary | Активирует предположение, что файл записан в двоичном формате matlab v6. |
| -v4 -V4 -4 -mat4-binary | Активирует предположение, что файл записан в двоичном формате matlab v4. |
| -zip -z | Используется алгоритм gzip для сжатия файла. Это работает с файлами, сжатыми с помощью gzip вне octave. Эта опция доступна только если octave была создана со ссылкой на библиотеки zlib. |

Функции `fileread` и `fdisp`

Функция `fileread` (`имя_файла`) возвращает содержимое файла в виде значения типа строка. Файл должен содержать символы текущей кодировки ОС.

Функция `fdisp` позволяет записывает данные в файл аналогично функции `disp`, которая преобразует значения своих параметров в символьную строку используя способы преобразования, заданные функцией `format`, но выводит их на экран. Для этого первым фактическим параметром функции `fdisp` должно быть указано значение дескриптора файла — числа типа `double`. Функция `fopen` автоматически генерирует это число и устанавливает его однозначное соответствие имени файла. По завершению операций с файлом это соответствие должно быть деактивировано функцией `fclose`.

Мы не будем подробно рассматривать функции `fopen` и `fclose`, которые описаны в [Док, с. 292 — 294] в подразделе 14.2.1 Opening and Closing Files раздела 14.2 C-Style I/O Functions. Рассмотрим простой пример.

```
ybgv@ybgv-home:~> octave -q -p ~/MyOct
octave:1>
octave:1> fileread fdsp.m
warning: fopen: '/home/ybgv/MyOct/fdsp.m' found by searching
load path
warning: called from
    fileread at line 42 column 7

ans =
echo on
cd
cd MyOct
rmdir ("IO", "s")
mkdir IO
cd IO
pwd
ls -l
Fdesc = fopen ("File.utf", "w") # w - право записи
fdisp (Fdesc, "Первая запись")
fdisp (Fdesc, "Число 999")
fdisp (Fdesc, 999)
fdisp (Fdesc, "Число pi")
fdisp (Fdesc, pi)
fclose (Fdesc)
ls -l
fileread File.utf

Fdesc = fopen ("File.utf", "w") # w - право записи
```

```
fdisp (Fdesc, "Вторая запись")
fdisp (Fdesc, "Число 777777777")
fdisp (Fdesc, 777777777)
fdisp (Fdesc, "Число e")
fdisp (Fdesc, e)
fclose (Fdesc)
ls -l
fileread File.utf
```

```
octave:2> fdsp
+ echo on
+ cd
+ cd MyOct
+ rmdir ("IO", "s")
remove entire contents of IO? (yes or no) y
Please answer yes or no.
remove entire contents of IO? (yes or no) yes
+ mkdir IO
+ cd IO
+ pwd
ans = /home/ybgv/MyOct/IO
+ ls -l
итого 0
+ Fdesc = fopen ("File.utf", "w") # w - право записи
Fdesc = 11
+ fdisp (Fdesc, "Первая запись")
+ fdisp (Fdesc, "Число 999")
+ fdisp (Fdesc, 999)
+ fdisp (Fdesc, "Число pi")
+ fdisp (Fdesc, pi)
```

```
+ fclose (Fdesc)
ans = 0
+ ls -l
итого 4
-rw-r--r-- 1 ybgv users 66 Nov 10 18:28 File.utf
+ fileread File.utf
ans = Первая запись
Число 999
999
Число pi
3.1416

+
+ Fdesc = fopen ("File.utf", "w") # w - право записи
Fdesc = 11
+ fprintf (Fdesc, "Вторая запись")
+ fprintf (Fdesc, "Число 777777777")
+ fprintf (Fdesc, 777777777)
+ fprintf (Fdesc, "Число e")
+ fprintf (Fdesc, e)
+ fclose (Fdesc)
ans = 0
+ ls -l
итого 4
-rw-r--r-- 1 ybgv users 78 Nov 10 18:28 File.utf
+ fileread File.utf
ans = Вторая запись
Число 777777777
7.7778e+08
Число e
2.7183
```

```
+  
+  
octave:3>
```

Отметим, что каждый вызов функции `fdisp` завершается записью в файл символа перехода на новую строку.

Упомянем также функцию `native_float_format`, которая выводит на экран данные о текущем используемом формате чисел с плавающей запятой, например:

```
octave:3> native_float_format  
ans = ieee-le  
octave:4>
```

что означает — представление по стандарту IEEE-754, порядок байтов `little endian` — т. е. байт младшей цифры целого числа имеет наименьший адрес, адреса следующих по старшинству байтов последовательно возрастают.

Мы не будем рассматривать представленные в [Док, с. 282 — 292] функции:

- `dlmwrite`, `dlmread` — запись/ чтение матриц с элементами числовых типов с заданием разделителя между элементами в строках;
- `csvwrite`, `csvread` — запись/ чтение матриц с элементами числовых типов в формате `csv` (`comma-separated values` - значения, разделенные запятыми);
- `textscan` — чтение данных из текстового файла или строки с грамматическим разбором на основе заданного формата;
- `importdata` — импорт данных из файлов разных форматов;
- `rescale` — сдвиг и/или нормализация данных для дальнейшего анализа;
- группа функций для управления сохранением на диске значений переменных в рабочем пространстве при аварийном завершении `octave`.
 - `crash_dumps_octave_core`;

- `sigchup_dumps_octave_core;`
- `sigquit_dumps_octave_core;`
- `sigterm_dumps_octave_core;`
- `octave_core_file_options;`
- `octave_core_file_limit;`
- `octave_core_file_name.`