

## Глава 1. Первые шаги

### Содержание главы 1:

- Как запустить Scilab?
- Как покинуть сессию Scilab?
- Где находятся сведения об инициализации Scilab?
- Как изменить шрифт в окне Scilab?
- Как поместить часть текста из окна Scilab в текстовый редактор или какой-либо другой пакет?
- Как редактировать командную строку?
- Как синтаксически перенести длинную командную строку на следующую строку?
- Различаются ли программой строчные и заглавные буквы?
- Как посмотреть демонстрационные примеры?
- Как узнать имя текущего каталога в Scilab?
- Как изменить текущий каталог в Scilab?
- Как запустить программу на выполнение?
- Как получить справку?
- Как понимать формулы синтаксиса команд?
- Как узнать версию пакета Scilab?
- Как сделать комментарий?
- Как использовать Scilab в качестве простого калькулятора?
- Как получить список текущих имен переменных ?
- Как записать результат сессии Scilab в файл?
- Как выполнять операции?
- Как использовать файл сценария?

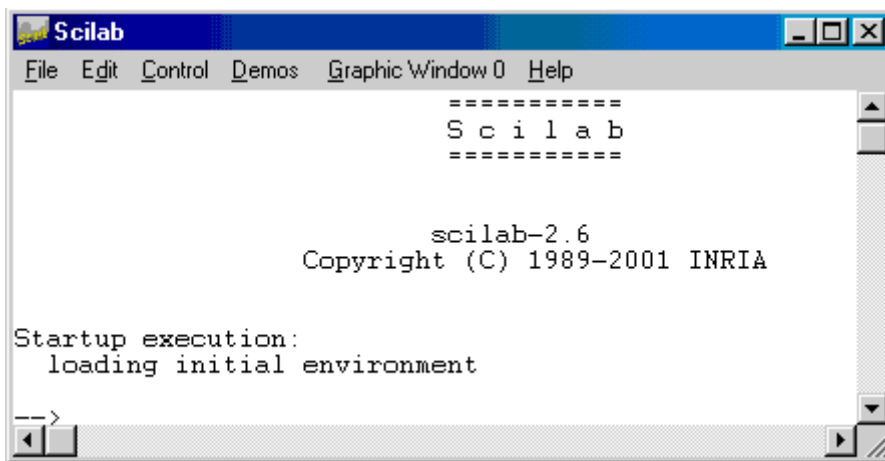
---

## Глава 1. Первые шаги

### Как запустить Scilab?

Из командной строки наберите **scilab\bin\runscilab.exe**

В результате получите главное рабочее окно Scilab с приглашением `-->`:



*Замечание:* В версиях Scilab для операционных систем Linux и Windows могут иметься небольшие отличия в меню и подменю верхней панели управления окна.

---

### Как покинуть сессию Scilab?

Набрать **exit** и нажать **[Enter]**

либо

набрать **quit** и нажать **[Enter]**

либо

выбрать на панели управления окна **File-Exit**

---

### Где находятся сведения об инициализации Scilab?

Начальные установки Scilab находятся в файле `\scilab\scilab.star`. Наверное лучше его не трогать...

---

### Как изменить шрифт в окне Scilab?

С помощью верхней панели окна выполнить **Edit-Choose Font** и далее выберите желаемую гарнитуру и размер шрифта.

---

## Как поместить часть текста из окна Scilab в текстовый редактор или какой-либо другой пакет?

Выделите нужный текст мышью, затем **[Edit][Copy to Clipboard]**.

Затем в Вашем пакете выполните в нужном месте **[Paste]** .

*Примечание:* Привычная замена выбора режим сору как ctrl-c и режима Paste как ctrl-v здесь не срабатывает. Однако, в окне помощи (Help) эти "горячие" кнопки работают.

---

## Как редактировать команду строку?

Посмотрите **Edit-History**, либо можно набрать текст в любом Windows-приложении и перенести его через Clipboard в окно Scilab. ( Примечание: русский язык поддерживается) Рекомендации режима **Edit-History** понадобятся, видимо, только при работе под платформой LINUX. Для набора текста на русском языке выберите в окне scilab режим **Edit-ChooseFont** и далее нужный шрифт, например, Courier.

---

## Как синтаксически перенести длинную командную строку на следующую строку?

В месте переноса добавить две точки.

Пример.

```
plotframe(rect,tics, [%t,%t],..  
["My plot with grids and automatic bounds","x","y"], [0,0.5,0.5,0.5])
```

вместо

```
plotframe(rect,tics, [%t,%t], ["My plot with grids and automatic  
bounds","x","y"], [0,0.5,0.5,0.5])
```

---

## Различаются ли программой строчные и заглавные буквы?

Да, и не только в LINUX, но и в Windows. Если вы определили значение величины "x", значение величины "X" ей не равно!

Пример.

```
->x=5; X=3;
```

```
=>y=x+X+x
```

```
y =
```

```
13. //то есть 5+3+5
```

---

## Как посмотреть демонстрационные примеры?

Выберите в строке меню **Demos** и следуйте дальнейшим указаниям. Это соответствует исполнению в командном режиме

**exec('SCI/demos/alldems.dem');**

Демонстрации будут открываться в новом окне. По окончании просмотра его следует закрыть.

---

## Как узнать имя текущего каталога в Scilab?

С помощью меню **File - GetCurrentDirectory**. Это эквивалентно командной строке **-->getcwd()**

Результат:

```
ans =  
D:\scilab\bin
```

Аналогичный результат дает и команда **pwd()**. Утверждается, что **pwd** печатает текущий каталог Scilab, а **getcwd** возвращает его. (Я не почувствовала разницы).

Синтаксис этих команд:

```
pwd  
x=pwd()  
x=getcwd()
```

Примеры.

```
pwd  
x=pwd()
```

---

## Как изменить текущий каталог в Scilab?

С помощью команды **chdir**.

Синтаксис

```
ierr=chdir('path-name')
```

Параметры

**ierr** : целое число, равно 1, если команда не может изменить каталог ошибка, и 0 во всех остальных случаях.

**path-name** : название нового каталога

Пример.

```
f=chdir("d:\my_scilab")
```

Результат: f=0.

*Замечания:*

1) Если мы хотим указать имя подкаталога пакета Scilab, то место нахождения самого пакета можно не указывать, заменив его на конструкцию **SCI+**.

Пример.

```
chdir(SCI+"/demos")
```

эквивалентно

```
chdir("c:/scilab/demos")
```

2) Если каталог не существует, у меня почему-то появляется сообщение об ошибке и **ierr=240 (Error 998)**. В каком же случае **ierr** будет =1?

---

## Как запустить программу на выполнение?

С помощью меню **File-Exec** либо командой **exec**.

---

## Как получить справку (help)?

Есть три подменю в меню **Help**.

1. Режим **Help-HelpDialog**. Допустим, мы хотим узнать синтаксис операции "exec". Нижняя часть таблицы позволяет выбрать интересующую нас главу. Это будет глава "Scilab Programming". Теперь выберите из верхнего окна операцию "exec" и нажмите кнопку **Show**. В результате получим справочную информацию по этой операции.
2. Режим **Help-Topic** позволяет получить справку прямо по названию команды.
3. Режим **Help-Apropos** позволяет получить справку по задаваемому ключевому слову. Используется, когда Вы забыли название нужной команды.

Еще одна возможность: в командной строке после приглашения `-->` набрать **help <имя команды>**. Эквивалентно **Help-Topic**.

Пример.

```
help exec
```

*Замечание:* Для того, чтобы узнать в каких каталогах присутствуют соответствующие тексты для **help**, служит переменная **%helps**. Значение **%helps** по умолчанию устанавливается в файле **scilab.star**.

---

## Как понимать формулы синтаксиса команд?

Полный синтаксис команд можно получить с помощью команды **help** с параметрами. Параметры, указанные в квадратных скобках не являются обязательными.

Пример.

Пусть мы хотим узнать, как пользоваться командой вычисления синуса **sin**. Выполните **help sin**. В формуле синтаксиса **[t]=sin(x)** параметр **x** - обязателен, а параметр **t** может и не присутствовать. Иногда синтаксис команды не укладывается в одну формулу и представляется несколько вариантов ее применения. В качестве примера посмотрите **help rand**.

---

## Как узнать версию пакета scilab?

Выполните команду **version=getversion()**

Результатом может быть:

```
version = scilab-2.6
```

---

## Как сделать комментарий?

Комментарии должны начинаться также, как и в языке C, с конструкции `//`. Это применяется обычно при выполнении программ в Scilab.

Пример.

```
--> // It is plot of 2 functions --> z=3456; //It is my number
```

---

## Как использовать Scilab в качестве простого калькулятора?

Ввести цифровое выражение и нажать [Enter]. Результатом будет: `ans=<ответ>`

Пример 1.

```
2+3^2
ans =
11.
```

Пример 2.

```
a=1;
b=7;
x=a+b
```

Будет получен ответ

```
x =
8.
```

*Замечание:* Если в конце строки стоит знак "точки с запятой"(`;`), то эта операция выполняется не сразу, а после набора всей последовательности команд до той, после которой уже не стоит точка с запятой.

---

## Как получить список текущих имен переменных ?

Используйте команду **who**. Вместо **who** можно писать **who()**. Результат тот же.

Пример.

```
a=3;
b=5;
who your variables are...
```

```
b a startup ierr demolist
%scicos_display_mode scicos_pal %scicos_menu
%scicos_short %helps MSDOS home PWD TMPDIR
percentlib soundlib xdesslib utillib tdcslib siglib
s2flib roplib optlib metalib elemllib commlib polylib
autolib armalib alglib intlib mtlbllib WSCI SCI
%F %T %z %s %nan %inf $
%t %f %eps %io %i %e
using 5866 elements out of 1000000.
and 48 variables out of 1791
```

Переменные, указанные во второй строке и ниже, являются встроенными. **who('local')** позволяет увидеть все имена встроенных переменных. Например, видно, что существует переменная с именем **%e**. Выполнение команды **%e** [Enter] даст нам ее значение

```
%e =
2.7182818
```

---

## Как записать результат сессии Scilab в файл?

Способ 1.

Мы хотим сохранить полученные в окне Scilab записи в файл. Для этого выполним команду **save('my\_file')**. В простом случае для этого можно использовать меню **File-Save**. В результате файл запишется в текущий каталог. По умолчанию это каталог `scilab\bin\`. Расширение создаваемого файла по умолчанию `*.bin`. Полученный файл в дальнейшем можно загрузить с помощью команды **load**.

Способ 2.

С помощью команды **diary**. Это журнал для записи копии текущей сессии Scilab в текстовый файл.

Синтаксис

**diary('file-name')**

Пример.

```
diary('D:/slon/my_example.txt')
a=5; // Далее идут команды сессии Scilab.
b=6
c=a*b+3
diary(0) //Это признак конца записи сессии
```

Все, что находится в главном окне Scilab между командой **diary('file-name')**, открывающей запись сессии, и командой **diary(0)**, закрывающей запись, будет записано в файл.

*Замечание:* Содержание графических окон при этом не сохраняются.

---

## Как выполнять операции?

Если мы хотим, чтобы каждая операция выполнялась непосредственно сразу, следует после приглашения `->` набрать операцию и клавишу ввода **[Enter]**. Можно набирать операции последовательно, разделяя их знаком `;`, а потом выполнить **[Enter]**. Тогда на экране будет выведен только конечный результат. Можно писать после приглашения в одной командной строке несколько операций, разделяя их точкой с запятой. Тогда результаты тех конструкций, которые заканчиваются знаком `;` не будут выводиться на экран. Мы увидим конечный результат выполнения последовательности нескольких командных строк, последняя из которых не заканчивается точкой с запятой. Результаты промежуточных операций на экран выводиться не будут.

Пример 1.

```
-->a=2
a =
2.
-->b=3
b =
3.
-->c=a+b
c =
5.
```

Пример 2.

```
-->a=2;
```

```
-->b=3;  
-->c=a+b;  
-->d=c*10  
d =  
50.
```

Если мы хотим узнать, чему было равно значение переменной "c", эта возможность нами не утрачена:

```
-->c  
c =  
5.
```

*Замечание по синтаксису:* Если командная строка очень длинная, то ее можно разбить на две, добавляя в месте разбиения две точки (..).

Пример.

Две записи, приводящие к идентичному результату:

1)

```
-->plot2d(x, [sin(x) sin(2*x) sin(3*x)], ..  
-->[1, 2, 3], "111", "L1@L2@L3", [0, -2, 2*%pi, 2], [2, 10, 2, 10])
```

2)

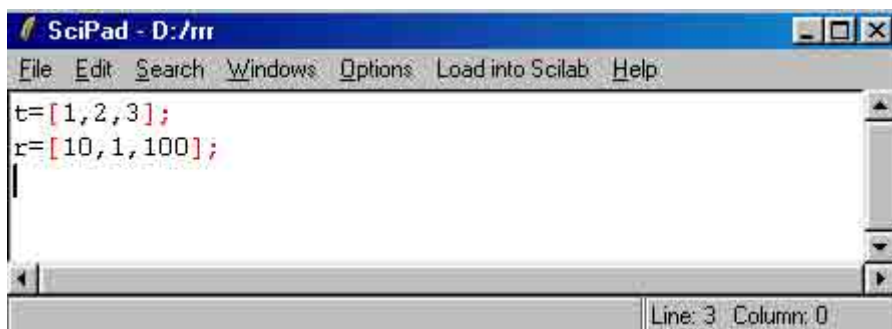
```
-->plot2d(x, [sin(x) sin(2*x) sin(3*x)], [1, 2, 3], "111", "L1@L2@L3", [0, -  
2, 2*%pi, 2], [2, 10, 2, 10])
```

---

## Как использовать файл сценария?

Сценарий является удобным средством ввода команд в пакет Scilab . В файле сценария можно записать ряд последовательных операций, которые сразу будут подгружаться в Scilab блоком . Это альтернатива набору команд непосредственно в главном окне Scilab в режиме командной строки. В версии 2.7 дополнительно создан специальный редактор SciPad для создания файла сценария. Для вызова этого редактора следует выполнить из основного окна Scilab с помощью меню верхней панели **File-Editor**.

В результате получите окно редактора Scipad:



После набора текста сценария возможны два способа действия:

- 1) С помощью меню верхней панели окна **Load into Scilab** текст сценария сразу будет загружен в сессию Scilab. Этот путь возможен только для версии 2.7 и использования редактора SciPad.
- 2) С помощью команд меню верхней панели окна Scipad-редактора **File - Save** записать в файл сценария с расширением **sce**, который может быть загружен позже в сессию Scilab из главного окна с помощью команд меню **File-Exec** для файлов с расширением **sce** или **sci** и для загрузки функций (с расширением **sci**) с помощью команд меню **File-Getf**.

В более ранней версии пакета 2.6 файл сценария можно создать в произвольном текстовом редакторе и использовать способ 2) для ввода его в Scilab.