

# SCILAB

---

## Глава 6. Интерфейс между программами, написанными на языках C и Fortran и пакетом Scilab

### Содержание главы:

- [Как добавить код C и Fortran программ в Scilab?](#)
  - [Динамическое линкование](#)
  - [Особенности интерфейса между C, Fortan и Scilab в операционной системе Windows](#)
- 

Пакет Scilab может быть легко сопряжен с программами на языках C и Fortan. Это важно для получения более быстрого кода, например, в задачах моделирования и оптимизации. Фактически сопряженный (interface) числовой код необходим для большинства нетривиальных приложений. Это может быть осуществлено с помощью динамической связи (dynamic link) или с помощью создания нового исполняемого кода для Scilab. Для выполнения C или Fortran программы, связанной с пакетом Scilab, ее входные параметры должны даны специальным образом передаваться в Scilab, а ее выходные параметры должны трансформироваться в Scilab-переменные. Например, можно также сделать так, что линкуемая программа будет автоматически вызываться с помощью команды высокого уровня: например, команды **ode**, которая позволяет интегрировать дифференциальное уравнение  $dy/dt=f(t,y)$ , используя в правой части функцию **f**, определенную в виде C или Fortran программы и динамически линкованной в пакет Scilab.

---

### Как добавить код C и Fortran программ в Scilab?

---

#### 1) Способ 1.

Самый простой способ вызвать внешнюю программу с помощью команды **link**, которая динамически свяжет программу пользователя с пакетом Scilab и затем вызывать затем вызывать линкованную подпрограмму с помощью команды **call**, которая преаает переменные Scilab линкованной программе и трансформирует назад выходные параметры в переменные Scilab. Такой способ мы будем называть динамическим линкованием. Следует заметить, что команды **ode**, **dae** и команды нелинейной оптимизации могут непосредственно (прямо) динамически связываться использовать с C или Fortran программами пользователя

## 2) Способ 2.

Построение программы интерфейса. Они могут быть написаны пользователем на основе примеров данных в каталогах [scilab/routines/examples/interface-tutorial](#) и [scilab/routines/examples/interfacetour](#). Примеры для Matlab-подобного интерфейса даны в каталоге [scilab/routines/examples/mexfiles](#).

## 3) Способ 3.

Программа интерфейса может быть создана с помощью команды **intersci**. Она использует файл **.desc**, который описывает и C (или Fortran) программу(ы), и имя и параметры соответствующей в Scilab функции(ий).

## 4) Способ 4.

Можно добавить постоянную новую команду в Scilab, составив программу интерфейса, как было сказано выше, и затем создав новый исполняемый код для Scilab. Новая команда должна быть добавлена в файл [scilab/routines/default/fundef](#). В этом случае программе интерфейса будет дано специальное название (например, по умолчанию **matus2**) и номер. Файл `/default/fundef` должен быть обновлен. Новый выполняемый код будет создан после выполнения **"make all"** в главном каталоге Scilab.

---

## Динамическое линкование

---

Пусть мы хотим в пакете Scilab воспользоваться функцией, написанной на языке C. В качестве примера воспользуемся текстом функции `ext1c.c` из каталога [/scilab/examples/link-examples](#). Это подпрограмма выполняет сложение двух матриц. *Замечание:* В каталоге [/scilab/examples/link-examples](#) представлены и другие простые и полезные примеры.

### Последовательность действий для осуществления динамического линкования:

#### Шаг 1.

Набрать в любом текстовом редакторе подпрограмму для вычисления функции на языке C или Fortan. Сохранить ее с расширением **.c** для языка C (соответственно **.f** для языка Fortan).

Содержание файла `ext1c.c`:

```
int ext1c(n, a, b, c)
int *n;
double *a, *b, *c;
{
int k;
for (k = 0; k < *n; ++k)
{
c[k] = a[k] + b[k];
}
return(0);
}
```

## Шаг 2.

Скомпилировать `ext1c.c` в объектный файл.

Для этого на компьютере должен находиться компилятор языка C (Fortran). Компиляторы не являются частью пакета Scilab и не содержатся в нем. Если Вы работаете в операционной системе Windows, то следует пользоваться компиляторами из пакета [cygwin](#).

```
gcc -c ext1c.c
```

В результате будет создан файл `ext1c.o`

## Шаг 3. Запустить пакет Scilab и в нем выполнить:

```
link("ext1c.o", "ext1c", "C")
```

*Замечание:* В данном случае имя подпрограммы совпадает с именем объектного файла. В общем случае в одном объектном файле подпрограмм может быть несколько. Подробно смотрите [help link](#).

Для проверки, правильно ли линкован файл, служит команда `c_link`. Если все корректно, то команда возвращает значение `T` (TRUE).

## Шаг 4.

Теперь мы можем использовать `ext1c` в качестве имени функции внутри пакета Scilab.

Вызов этой функции может осуществляться с помощью команд `call` или `fort`.

```
x_a=[1,2,3];x_b=[4,5,6];x_n=3; // Задание начальных условий  
c=call('ext1c',x_n,1,'i',x_a,2,'d',x_b,3,'d','out',[1,3],4,'d')
```

В результате получим суммарную матрицу

```
c=[5,7,9]
```

В качестве параметров команды `call` выступают имя вызываемой подпрограммы (в данном случае `'ext1c'`) и ее аргументы. В примере приведена длинная форма синтаксиса: после значения каждого аргумента указан номер его порядковый номер в вызываемой процедуре и его тип в кавычках. Например, конструкция `...,x_a, 2,'d',...` в качестве параметров команды `call` означает, что `x_a` является значением второго аргумента для подпрограммы `ext1c` типа `double`.

Возможно применение и короткой формы синтаксиса команды вызова `call`, которая дает более быстрый код и простой синтаксис. См. [help call](#). Однако это потребовало бы написания дополнительного небольшого (C или Fortan) интерфейса.

*Замечания:*

- 1) Для прекращения линкования служит команда `ulink`. Для простоты лучше использовать для линкования форму `q=link("ext1c.o", "ext1c", "C")`. Тогда команда прекращения линкования будет выглядеть как `ulink(q)`. В противном случае вместо `q` следует поставить целое число, отвечающее за соответствующий процесс.
- 2) Для получения информации о линкованных на данный момент библиотеках и функциях служит команда `link("show")`

Пример для языка Fortran.

### Шаг 1.

Содержание файла `ext1f.f`:

```
subroutine summa(n,a,b,c)
double precision a(*),b(*),c(*)
do 1 k=1,n
c(k)=a(k)+b(k)
1 continue
return
end
```

Запишите его в файл `ext1f.f`.

### Шаг 2.

```
g77 -c ext1f.f
```

### Шаг 3.

Перейдите в среду Scilab и выполните:

```
link("ext1f.o","summa","F")
```

Замечание: Для языка fortran флаг "F" можно опустить: `link("ext1f.o","summa")`

### Шаг 4.

Теперь можете использовать подпрограммы из слинкованного файла `ext1f.o`.

```
a=[1,2,3];b=[4,5,6];n=3;
call('summa',n,1,'i',a,2,'d',b,3,'d','out',[1,3],4,'d')
```

Результат:

```
ans =
! 5. 7. 9.!
```

---

## Особенности интерфейса между C, Fortan и Scilab в операционной системе Windows

---

Для работы в операционной системе Windows необходимо дополнительно установить один из двух свободно распространяемых пакетов:

- Cygwin
- [MinGW](#)("Minimalistic GNU for Windows").

Смотрите текст в файлах `\scilab\Win85-util\DllDemo\how-to-build-dlls.txt` и `\scilab\Win85-util\DllDemo\Readme.txt`.

## Об операционной среде Cygwin

Cygwin является операционной UNIX-средой, расширенная Red Hat, для Windows. Она состоит из двух частей:

- 1) DLL (cygwin1.dll), которая работает как UNIX эмуляция
- 2) Из коллекции инструментов, перенесенных из UNIX, позволяющих имитировать присутствие в UNIX/LINUX.

Полезные ссылки о cygwin:

- <http://sources.redhat.com/cygwin/>
- <http://cygwin.com/>
- <http://sources.redhat.com/cygwin/download.html>
- Cygwin Package List <http://sources.redhat.com/cygwin/packages>
- Cygwin User's Guide <http://cygwin.com/cygwin-ug-net/cygwin-ug-net.html>
- <http://cygwin.com/xfree>

*Если Вы хотите пользоваться удаленной машиной с установленным Linux и Scilab, а на Вашем терминале установлена только операционная система Windows, то для полноценного доступа к пакету на удаленной машине Вам будет необходим cygwin с приложением эмуляции терминала [xfree86](#).*

***Xfree86** - это свободно распространяемая система XWindow System, которая работает как под Unix-образными системами, включая Linux, Sun Solaris x86, так и под платформами OS/2 и Cygwin.*

*Замечание: Установка **Cygwin** дело само по себе хлопотное и стоит подумать, не установить ли вместо этого Linux на локальной машине (да и Scilab под Linux тоже). [Здесь](#) можно найти информацию о различных перекомпилированных версиях Scilab.*