

**Министерство образования РФ**  
**Самарский государственный аэрокосмический университет**  
**академика С. П. Королева**

**имени**

## **ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ СИ**

*Методические указания к лабораторным работам*  
*по курсу "Информатика" для студентов 1 курса* *радиотехнического*  
*факультета*

**Самара - 2001**

Составитель **А. А. Тюгашев**

**Использование языка программирования Си:** Метод.указания к лабораторным работам/Самарский аэрокосм.ун-т; Сост. А.А.Тюгашев. Самара, 2001. 14 с.

Рассматривается комплекс проблем, связанных с использованием одного из самых популярных на сегодняшний день языков программирования – Си. Рассматриваются принципы работы в интегрированной среде разработчика компании Borland, типовые программы для работы с массивами и матрицами, сортировки, нахождения сумм и максимумов, работы с файлами и структурами данных.

Предназначены для студентов радиотехнического факультета, выполняющих лабораторные работы по курсу "Информатика". Могут быть использованы в практической деятельности. Подготовлены на кафедре компьютерных систем.

Печатаются по решению Редакционного Совета Самарского государственного аэрокосмического университета

## ВВЕДЕНИЕ

Язык программирования Си является одним из наиболее распространенных в мире средств для разработки профессиональных и сложных программных комплексов, вероятно, самым распространенным. Со времени своего создания в 1972 году язык претерпел ряд существенных дополнений и изменений, став основой, в частности, для современного объектно-ориентированного языка программирования C++.

Язык Си создавался как язык системного программирования, тесно связанный с операционной системой UNIX. Его авторами

являются Брайан Керниган и Деннис Ритчи, работавшие в то время в фирме Bell корпорации AT&T.

В настоящих методических указаниях рассмотрены основы работы в интегрированной среде Borland C++, вопросы создания новых программ и их отладки.

## 1. ОБЩИЕ СВЕДЕНИЯ РАБОТЕ В СРЕДЕ BORLAND C++

### 1.1. Принципы создания и отладки программ.

Принцип работы с интегрированной средой базируется на концепции "редактор - транслятор - отладчик". При программировании тех или иных задач работа программиста представляет собой, как правило, итеративный (циклический) процесс, включающий в себя следующие основные стадии:

- набор исходной программы в текстовом редакторе;
- трансляция (компиляция);
- компоновка (линковка);
- запуск на выполнение;
- отладка.

В обычном варианте при прохождении всех перечисленных стадий необходим последовательный вызов средствами операционной системы соответствующих программ: текстового редактора,

транслятора, компоновщика, отладчика. При работе с интегрированной средой разработчика (IDE, Integrated Development Environment) программист общается только с ее собственным, дружественным интерфейсом, а вызов необходимых служебных программ и модулей осуществляется автоматически. Интерфейс среды разработчика Borland C++ построен на основе меню и окон. Для редактирования каждого текстовых файла открывается одно окно, при этом возможен перенос фрагментов текста из одного файла в другой через буфер оболочки. При трансляции открывается специальное окно трансляции, существует специальное окно для вывода сообщений об ошибках, и т.д. Навигацию по окнам можно осуществлять с помощью нажатия клавиш Alt+<номер требуемого окна>. С помощью нажатия клавиши F5 возможно развернуть активное окно на весь экран. Обратное изменение возможно с помощью еще одного нажатия F5. Alt+F3 приводит к закрытию текущего окна. Ctrl+F5 включает режим изменения размеров и перемещения окон.

Доступ к большинству функций интегрированной системы возможен через главное меню оболочки, которое можно вызвать с помощью клавиши F10.

Перемещаться по меню можно, используя клавиши управления курсором – клавиши со стрелками. При нажатии на клавишу Enter происходит выбор соответствующего пункта в меню. Пункты главного меню имеют связанные с ними так называемые “выпадающие” подменю, например, с пунктом главного меню “File” связано выпадающее меню, в котором перечисляются основные операции с файлами – создание, открытие, сохранение (запись), и т.д.

Для выхода из меню (равно как и для отмены какого-либо действия), можно использовать клавишу Esc. Некоторые пункты выпадающих подменю главного меню могут иметь рядом с названием пункта многоточие (“ . . .”) либо знак треугольника. Это означает, что при выборе данного пункта происходит открытие дополнительного окна диалога или нового подменю. Некоторые пункты меню можно вызвать с помощью так называемых “горячих клавиш” непосредственно, например, в процессе набора текста программы во встроенном в IDE текстовом редакторе. Эти клавиши (или их комбинации) встречаются в соответствующих пунктах

подменю. Например, с помощью нажатия клавиши F3 можно вызвать стандартный диалог открытия файла. При нажатии комбинации "Alt+X" происходит быстрый выход из среды разработчика. Пункты подменю можно вызывать и с помощью комбинации клавиш Alt и буквы, подчеркнутой в названии соответствующего пункта.

Работа с меню, кроме использования клавиатуры, возможна с использованием манипулятора "мышь".

В нижней части экрана при работе с интегрированной оболочкой Borland C++ располагается так называемая строка состояния. Она выполняет следующие функции.

- напоминает о клавишах, которые можно использовать в данный момент.
- позволяет выполнять те или иные действия с помощью манипулятора "мышь". Достаточно указать название действия и щелкнуть на нем левой кнопкой мыши.
- информирует пользователя о действии, выполняемом в настоящий момент (например, сохранение файла).
- предлагает советы и рекомендации по выбранным пунктам меню или диалога.

Система имеет развитую систему так называемой контекстно-зависимой (то есть "понимающей", в каком режиме происходит работа и предлагающей помощь именно по нему) подсказки. Есть соответствующий пункт в главном меню, есть и соответствующая "горячая клавиша" - F1. Есть и еще одна полезная особенность. При подводе курсора в окне текстового редактора к набранному оператору или стандартной функции языка Си и нажатии комбинации клавиш Ctrl+F1 выводится информация о данном ключевом слове.

Отладка (debugging) программ подразумевает процесс выявления и устранения ошибок в написанных программах. Как гласит известная в программистских кругах аксиома, "каждая программа содержит по крайней мере одну ошибку". Еще одно шутливое изречение гласит, что "устраняя одну ошибку в программе, мы вносим три новых". И, хотя в этих утверждениях есть только доля шутки, данный факт не дает нам оснований отказаться от устранения ошибок в программах.

Встроенный в IDE Borland C++ отладчик обладает возможностями пол установки точек останова (Breakpoints) в

тексте программы, просмотра и модификации при необходимости текущих значений переменных.

Для успешной работы с оболочкой необходимо правильное задание так называемых параметров рабочего окружения. В частности, при работе с системой Borland C++ используются стандартные файлы, и необходимо указать их расположение (каталог на диске). Доступ к этому режиму осуществляется через пункт главного меню Options, подпункт Directories. В частности, при программировании на языках C/C++ используются так называемые заголовочные файлы, и строка в тексте программы

```
#include <stdio.h>
```

означает, что в текст необходимо вставить файл `stdio.h`, а искать его надо в некотором стандартном каталоге. Этот каталог надо явно задать в оболочке, в разделе Include Directories.

Еще одна важная группа файлов – служебные библиотеки, включающие наборы стандартных функций, таких например, как сравнение строк, математические и др. Их расположение задается в разделе Library Directories.

Наконец, каталог, в который будет помещаться готовая к исполнению программа с расширением `.exe`, задается в разделе Output Directory.

## 1.2 Написание первой программы на языке Си

Для проверки работоспособности и правильности настройки интегрированной среды разработчика напомним, оттранслируем и запустим на выполнение первую программу на языке Си.

Для этого, во-первых, создадим новое окно текстового редактора путем выбора в главном меню пункта File, подпункта New в выпадающем меню. Откроется новое окно текстового редактора. Наберем в нем нижеследующий текст.

```
#include <stdio.h>
int main()
{
    printf ("Здравствуй, Мир !\n");
    return 0;
}
```

Эта программа выводит на экран компьютера текстовое сообщение "Здравствуй, Мир!".

Разберем текст созданной программы. Первая строка содержит одну из так называемых директив препроцессора. Препроцессор – это специальная программа, обрабатывающая тексты на языке Си до того, как они передаются транслятору языка программирования, и осуществляющая некоторые предварительные действия в соответствии с директивами, которые для отличия от собственно операторов языка Си начинаются с символа "#". Директива `#include` в частности, указывает на необходимость включения в программу стандартного заголовочного файла `stdio.h` для подключения к программе библиотеки стандартных функций ввода-вывода (STANDARD Input - Output). Это необходимо для возможности дальнейшего использования библиотечной функции `printf`.

Следующая строка содержит заголовок идущей далее в тексте функции `main`. Надо сказать, что все программы на языке Си являются наборами функций. Функция определяется как некоторый модуль, имеющий в общем случае некоторые аргументы, записываемые в круглых скобках, и могущий возвращать какое-то значение. Одни функции могут вызывать на выполнение другие, и т.д. При этом система начинает выполнение любой программы на языке Си с функции со стандартным именем – `main`. Соответственно, каждая программа на языке Си должна включать ровно одну функцию с именем `main`. Перед названием функции написано "int", что означает, что функция возвращает целое число. Поскольку функция `main` – основная функция программы, то ее возвращаемое значение подразумевает возвращаемое в операционную систему программой значение. Существует соглашение, что если программа завершается нормально, без возникновения нештатных ситуаций, то она возвращает ноль, в противном случае – ненулевое значение. После названия функции `main` идут пустые круглые скобки, что означает, что наша программа не обрабатывает никаких входных значений, передаваемых через командную строку при вызове программы на выполнение (а при вызове с помощью команды `Run` интегрированной оболочки есть специальный пункт `Arguments` в соответствующем подменю).

В следующих строках после заголовка идет собственно тело функции, в котором содержатся действия, которые она выполняет, то есть операторы языка Си и вызовы функций. Тело функции ограничивается фигурными скобками - { и }.

Наша функция вызывает библиотечную функцию форматного вывода на стандартное устройство (в данном случае экран дисплея) - функцию printf. Аргументом данной функции является текстовая строка, которые в языке Си принято заключать в двойные кавычки. В конце строки можно заметить символы "\n", это специальная последовательность символов (на это указывает знак '\\'), означающая, что после вывода строки на экран надо перейти к новой строке. Точка с запятой - стандартный разделитель языка Си, разграничивающий отдельные действия в тексте функции.

Следом идет оператор return 0. Этот оператор возвращает некоторое значение в качестве результата выполнения функции. В данном случае мы передаем ноль операционной системе как свидетельство того, что функция завершается нормальным образом.

Последняя строка программы - закрывающая фигурная скобка, указывающая на конец функции main.

Теперь необходимо осуществить следующие действия:

1. Сохранение текста программы в некотором файле на диске.
2. Трансляция текста программы на языке Си в машинные коды.
3. Компоновка объектного модуля программы с модулем стандартной библиотеки.
4. Запуск готовой исполняемой программы.

Для выполнения первого этапа нажмем кнопку F3. Откроется стандартное окно сохранения файла. В нем можно видеть содержимое текущего каталога на диске, перемещаться по каталогам путем выбора соответственно ".." для перемещения на уровень вверх по дереву каталогов, и название поддиректории для спуска вниз по дереву, задавать имя сохраняемой программы. Пол умолчанию IDE предлагает для программы название "NONAME00.C". Исправим его, например, на "first.c". Необходимо обратить внимание на то, что в каталоге показываются не все файлы, а только те, которые имеют расширения, задаваемые соответствующим пунктом диалога. И при



сохранении файла предлагается одно из стандартных расширений - .c, .h или .cpp для программ на языке C++.

Указав имя файла и расширение .c, которое покаывает системе, что данный файл содержит текст программы на языке Си, выбираем пункт Ok диалога сохранения.

Этапы 2-4 можно выполнять по отдельности, но мы воспользуемся возможностью отдать интегрированной среде команду "сделай все необходимое и запусти программу" с помощью нажатия комбинации клавиш Ctrl+F9. Система тогда автоматически выполняет трансляцию, компоновку (о чем свидетельствует специальное окно, возникающее на экране), и в случае отсутствия ошибок, запускает программу на выполнение.

Не стоит удивляться, если вы не заметите результатов выполнения программы. Дело в том, что включенная в IDE система исполнения программ имеет свое собственное окно, отличное от основного рабочего окна разработки. Для перехода в окно выполнения необходимо использовать комбинацию Alt+F5. Тогда на экране появится строка "Здравствуй, Мир!", свидетельствующая о том, что наш эксперимент завершился успешно, и программа работает.

## 1. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ СИ

### 2.1. Идентификаторы и типы данных

Важным понятием языка программирования являются идентификаторы - имена переменных, функций и т.д. В языке Си идентификаторы представляют собой последовательность букв латинского алфавита, символов подчеркивания и десятичных цифр. При этом первым символом в идентификаторе должна быть буква. При этом заглавные и строчные буквы в языке Си различаются, то есть NAME, Name и name - три разных идентификатора. Имена объектов в программе не должны совпадать с операторами языка и названиями стандартных функций.

Идентификаторы используются для обозначения переменных и имен функций. Все переменные должны относиться к тому или иному типу данных. Базовыми типами данных в языке Си являются:

`int` - целое число  
`float` - число с дробной частью (с "плавающей точкой")  
`char` - символ  
`void` - не имеющий значения  
`double` - вещественное число двойной точности

Перед названием типа могут встречаться так называемые модификаторы, например `unsigned` для чисел означает, что данное число не может хранить отрицательное значение. Таким образом, если при 16-разрядном целом числе тип `int` может хранить числа от -32767 до 32768, то `unsigned int` - от 0 до 65535.

Имеются также модификаторы `short` и `long`, указывающие соответственно на короткое (занимающее меньше памяти) и длинное (для хранения больших значений) целое. При этом можно просто писать `short` и `long` вместо `short int` и `long int`.

Каждая используемая в программе переменная должна быть предварительно объявлена с указанием типа, и в дальнейшем она может использоваться для хранения значений только указанного типа. Допустимо объявлять переменные одного типа как через запятую в одной строке, так и в разных строках, например:

```

int a,b,c;
int x;
char sym, A21;

```

## 2.2 Операторы и выражения

Выражения широко распространены в языке Си. Они состоят из операндов (переменные, константы, числа), соединенных знаками операций (сложения, умножения и др.). Порядок выполнения операций определяется их приоритетами и круглыми скобками, используемыми при записи выражений.

Одним из главных операторов является оператор присваивания, используемый для задания какого-то значения переменной. В языке Си для записи присваивания используется символ "=", например:

```
x=a+b;
```

Более того, существует удобная сокращенная запись для присваивания одного и того же значения группе переменных, например:

```
f=b=c=d=100;
```

К **арифметическим операциям** в языке Си относятся: +, -, \*, /, % (остаток от деления).

Логические операции в языке Си:

-используемые для записи условий

&& - логическое И

|| - логическое ИЛИ

! - логическое НЕ

< - меньше

> - больше

== - равно (два стоящих рядом знака = )

<= - меньше или равно

>= - больше или равно

- используемые для поразрядных логических операций

& - поразрядное логическое И

| - поразрядное логическое ИЛИ

~ - поразрядное отрицание

^ - поразрядное логическое исключающее ИЛИ

В языке Си используются также и другие операции. К ним относятся:

++ - инкремент (увеличение на единицу)

-- - декремент (уменьшение на единицу)

?: - условная операция

Z= - набор операций, где Z - любая бинарная операция

>> - побитовый сдвиг вправо

<< - побитовый сдвиг влево

Очень удобными являются операции инкремента и декремента, которые можно рассматривать как сокращенные формы записи выражений  $x=x+1$  и  $x=x-1$ . Аналогичным образом, используется сокращенная запись вида  $z+=2$  вместо  $z=z+2$ ,  $y*=5$  вместо  $y=y*5$  и т.д.

Для условной операции ее результатом становится значение второго выражения, использованного в ее записи, в случае, если

первое выражение (условное) истинно, и значение третьего – если значение первого ложно. Например:

```
Y = (a > b) ? a : b;
```

Аналогично присвоению Y значения максимума среди a и b.

В случае, если в арифметическом выражении присутствуют операнды нескольких типов, все они приводятся к одному общему типу по встроенным в язык Си правилам. Кроме того, возможно явное (задаваемое программистом) преобразование типа, выполняемое следующим образом:

```
X = y + (float) z;
```

Здесь перед вычислениями переменная z будет приведена к типу чисел с плавающей точкой.

