

Николай Прохоренок

ЯЗЫК С

САМОЕ
НЕОБХОДИМОЕ

Санкт-Петербург

«БХВ-Петербург»

2020

УДК 004.438 С
ББК 32.973.26-018.1
П84

Прохоренок Н. А.

П84 Язык С. Самое необходимое. — СПб.: БХВ-Петербург, 2020. — 480 с.:
ил. — (Самое необходимое)

ISBN 978-5-9775-4116-9

Описан базовый синтаксис современного языка С: типы данных, операторы, условия, циклы, работа с числами, строками, массивами и указателями, создание пользовательских функций, модулей, статических и динамических библиотек. Рассмотрены основные функции стандартной библиотеки языка С, а также функции, применяемые только в операционной системе Windows. Для написания, компиляции и запуска программ используется редактор Eclipse, а для создания исполняемого файла — компилятор gcc.exe версии 8.2, входящий в состав популярной библиотеки MinGW-W64. Книга содержит большое количество практических примеров, помогающих начать программировать на языке С самостоятельно. Весь материал тщательно подобран, хорошо структурирован и компактно изложен, что позволяет использовать книгу как удобный справочник. Электронный архив с примерами находится на сайте издательства.

Для программистов

УДК 004.438 С
ББК 32.973.26-018.1

Группа подготовки издания:

| | |
|----------------------|--------------------------|
| Руководитель проекта | <i>Евгений Рыбаков</i> |
| Зав. редакцией | <i>Екатерина Сависте</i> |
| Компьютерная верстка | <i>Ольги Сергиенко</i> |
| Дизайн серии | <i>Марины Дамбиевой</i> |
| Оформление обложки | <i>Карины Соловьевой</i> |

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-5-9775-4116-9

© ООО "БХВ", 2020
© Оформление. ООО "БХВ-Петербург", 2020

Оглавление

| | |
|--|-----------|
| Введение | 9 |
| Глава 1. Установка программ под Windows | 11 |
| 1.1. Создание структуры каталогов | 11 |
| 1.2. Добавление пути в переменную <i>PATH</i> | 12 |
| 1.3. Работа с командной строкой | 14 |
| 1.4. Установка MinGW и MSYS | 14 |
| 1.5. Установка MinGW-W64 | 20 |
| 1.6. Установка MSYS2 и MinGW-W64 | 24 |
| 1.7. Установка и настройка редактора Eclipse | 29 |
| 1.8. Создание проектов в редакторе Eclipse | 34 |
| Глава 2. Первые шаги | 44 |
| 2.1. Первая программа | 44 |
| 2.2. Создание пустого проекта в редакторе Eclipse | 47 |
| 2.3. Добавление в проект файла с программой | 49 |
| 2.4. Добавление в проект заголовочного файла | 50 |
| 2.5. Компиляция и запуск программы в редакторе Eclipse | 53 |
| 2.6. Структура программы | 57 |
| 2.7. Комментарии в программе | 63 |
| 2.8. Вывод данных | 65 |
| 2.9. Ввод данных | 71 |
| 2.9.1. Ввод одного символа | 71 |
| 2.9.2. Функция <i>scanf()</i> | 72 |
| 2.9.3. Ввод строки | 77 |
| 2.10. Интерактивный ввод символов | 80 |
| 2.11. Получение данных из командной строки | 82 |
| 2.12. Предотвращение закрытия окна консоли | 84 |
| 2.13. Настройка отображения русских букв в консоли | 86 |
| 2.14. Преждевременное завершение выполнения программы | 89 |
| Глава 3. Переменные и типы данных | 91 |
| 3.1. Объявление переменной | 91 |
| 3.2. Именованые переменных | 92 |

| | |
|---|-----|
| 3.3. Типы данных | 93 |
| 3.4. Целочисленные типы фиксированного размера | 97 |
| 3.5. Оператор <i>sizeof</i> и тип <i>size_t</i> | 99 |
| 3.6. Инициализация переменных..... | 100 |
| 3.7. Оператор <i>typedef</i> | 100 |
| 3.8. Константы | 101 |
| 3.9. Спецификаторы хранения..... | 104 |
| 3.10. Области видимости переменных..... | 105 |
| 3.11. Массивы | 107 |
| 3.12. Строки..... | 110 |
| 3.13. Указатели..... | 111 |
| 3.14. Динамическое выделение памяти | 118 |
| 3.14.1. Функции <i>malloc()</i> и <i>free()</i> | 118 |
| 3.14.2. Функция <i>calloc()</i> | 119 |
| 3.14.3. Функция <i>realloc()</i> | 122 |
| 3.15. Структуры | 123 |
| 3.16. Битовые поля..... | 126 |
| 3.17. Объединения | 127 |
| 3.18. Перечисления | 129 |
| 3.19. Приведение типов..... | 130 |

Глава 4. Операторы и циклы..... 133

| | |
|---|-----|
| 4.1. Математические операторы..... | 133 |
| 4.2. Побитовые операторы..... | 135 |
| 4.3. Операторы присваивания..... | 138 |
| 4.4. Оператор запятая | 138 |
| 4.5. Операторы сравнения..... | 139 |
| 4.6. Приоритет выполнения операторов | 141 |
| 4.7. Оператор ветвления <i>if</i> | 142 |
| 4.8. Оператор <i>?:</i> | 146 |
| 4.9. Оператор выбора <i>switch</i> | 147 |
| 4.10. Цикл <i>for</i> | 149 |
| 4.11. Цикл <i>while</i> | 152 |
| 4.12. Цикл <i>do...while</i> | 152 |
| 4.13. Оператор <i>continue</i> : переход на следующую итерацию цикла..... | 153 |
| 4.14. Оператор <i>break</i> : прерывание цикла..... | 153 |
| 4.15. Оператор <i>goto</i> | 154 |

Глава 5. Числа..... 156

| | |
|--|-----|
| 5.1. Математические константы | 159 |
| 5.2. Основные функции для работы с числами | 160 |
| 5.3. Округление чисел | 164 |
| 5.4. Тригонометрические функции | 165 |
| 5.5. Преобразование строки в число | 165 |
| 5.6. Преобразование числа в строку..... | 174 |
| 5.7. Генерация псевдослучайных чисел..... | 177 |
| 5.8. Бесконечность и значение <i>NAN</i> | 179 |

| | |
|--|------------|
| Глава 6. Массивы | 181 |
| 6.1. Объявление и инициализация массива | 181 |
| 6.2. Определение количества элементов и размера массива..... | 183 |
| 6.3. Получение и изменение значения элемента массива..... | 184 |
| 6.4. Перебор элементов массива..... | 185 |
| 6.5. Доступ к элементам массива с помощью указателя | 186 |
| 6.6. Массивы указателей | 189 |
| 6.7. Динамические массивы | 189 |
| 6.8. Многомерные массивы | 190 |
| 6.9. Поиск минимального и максимального значений | 193 |
| 6.10. Сортировка массива | 195 |
| 6.11. Проверка наличия значения в массиве | 198 |
| 6.12. Копирование элементов из одного массива в другой..... | 201 |
| 6.13. Сравнение массивов | 203 |
| 6.14. Переворачивание массива..... | 204 |
| | |
| Глава 7. Символы и С-строки | 206 |
| 7.1. Объявление и инициализация отдельного символа | 206 |
| 7.2. Настройка локали | 212 |
| 7.3. Изменение регистра символов..... | 216 |
| 7.4. Проверка типа содержимого символа..... | 218 |
| 7.5. Объявление и инициализация С-строки..... | 222 |
| 7.6. Доступ к символам внутри С-строки | 224 |
| 7.7. Определение длины строки | 225 |
| 7.8. Перебор символов С-строки | 226 |
| 7.9. Основные функции для работы с С-строками..... | 227 |
| 7.10. Поиск и замена в С-строке | 232 |
| 7.11. Сравнение С-строк..... | 237 |
| 7.12. Форматирование С-строк..... | 242 |
| | |
| Глава 8. Широкие символы и L-строки | 244 |
| 8.1. Объявление и инициализация широкого символа | 245 |
| 8.2. Вывод и ввод широких символов..... | 247 |
| 8.3. Изменение регистра символов..... | 249 |
| 8.4. Проверка типа содержимого широкого символа | 251 |
| 8.5. Преобразование широких символов в обычные и наоборот..... | 255 |
| 8.6. Объявление и инициализация L-строки..... | 256 |
| 8.7. Доступ к символам внутри L-строки..... | 257 |
| 8.8. Определение длины L-строки..... | 258 |
| 8.9. Перебор символов L-строки | 259 |
| 8.10. Вывод и ввод L-строк..... | 260 |
| 8.11. Преобразование L-строки в L-строку и наоборот..... | 263 |
| 8.12. Преобразование кодировок..... | 265 |
| 8.13. Основные функции для работы с L-строками | 271 |
| 8.14. Поиск и замена в L-строке | 277 |
| 8.15. Сравнение L-строк..... | 282 |
| 8.16. Преобразование L-строки в число..... | 286 |
| 8.17. Преобразование числа в L-строку | 294 |
| 8.18. Типы <i>char16_t</i> и <i>char32_t</i> | 297 |

| | |
|--|------------|
| Глава 9. Работа с датой и временем..... | 300 |
| 9.1. Получение текущей даты и времени..... | 301 |
| 9.2. Форматирование даты и времени..... | 305 |
| 9.3. «Засыпание» программы..... | 309 |
| 9.4. Измерение времени выполнения фрагментов кода..... | 311 |
| Глава 10. Пользовательские функции..... | 312 |
| 10.1. Создание функции и ее вызов..... | 312 |
| 10.2. Расположение объявлений и определений функций..... | 315 |
| 10.3. Способы передачи параметров в функцию..... | 318 |
| 10.4. Передача массивов и строк в функцию..... | 320 |
| 10.5. Переменное количество параметров..... | 324 |
| 10.6. Константные параметры..... | 325 |
| 10.7. Статические переменные и функции..... | 326 |
| 10.8. Способы возврата значения из функции..... | 328 |
| 10.9. Указатели на функции..... | 330 |
| 10.10. Передача в функцию и возврат данных произвольного типа..... | 331 |
| 10.11. Рекурсия..... | 332 |
| 10.12. Встраиваемые функции..... | 333 |
| Глава 11. Обработка ошибок..... | 336 |
| 11.1. Типы ошибок..... | 336 |
| 11.2. Предупреждающие сообщения при компиляции..... | 337 |
| 11.3. Переменная <i>errno</i> и вывод сообщения об ошибке..... | 338 |
| 11.4. Способы поиска ошибок в программе..... | 341 |
| 11.5. Отладка программы в редакторе Eclipse..... | 345 |
| Глава 12. Чтение и запись файлов..... | 351 |
| 12.1. Открытие и закрытие файла..... | 351 |
| 12.2. Указание пути к файлу..... | 354 |
| 12.3. Режимы открытия файла..... | 356 |
| 12.4. Запись в файл..... | 358 |
| 12.5. Чтение из файла..... | 360 |
| 12.6. Чтение и запись двоичных файлов..... | 363 |
| 12.7. Файлы произвольного доступа..... | 365 |
| 12.8. Создание временных файлов..... | 367 |
| 12.9. Перенаправление ввода/вывода..... | 369 |
| 12.10. Работа с буфером ввода и вывода..... | 372 |
| Глава 13. Низкоуровневые потоки ввода и вывода..... | 374 |
| 13.1. Открытие и закрытие файла..... | 374 |
| 13.2. Чтение из файла и запись в файл..... | 377 |
| 13.3. Файлы произвольного доступа..... | 380 |
| 13.4. Создание временных файлов..... | 381 |
| 13.5. Дескрипторы потоков ввода/вывода..... | 382 |
| 13.6. Преобразование низкоуровневого потока в обычный..... | 383 |
| 13.7. Создание копии потока..... | 384 |
| 13.8. Перенаправление потоков..... | 384 |

| | |
|--|------------|
| Глава 14. Работа с файловой системой | 386 |
| 14.1. Преобразование пути к файлу или каталогу..... | 386 |
| 14.2. Переименование, перемещение и удаление файла | 390 |
| 14.3. Проверка прав доступа к файлу и каталогу | 391 |
| 14.4. Изменение прав доступа к файлу | 393 |
| 14.5. Делаем файл скрытым..... | 394 |
| 14.6. Получение информации о файле | 395 |
| 14.7. Функции для работы с дисками..... | 399 |
| 14.8. Функции для работы с каталогами..... | 401 |
| 14.9. Перебор объектов, расположенных в каталоге | 403 |
| Глава 15. Потоки и процессы..... | 407 |
| 15.1. Потоки в WinAPI | 407 |
| 15.1.1. Создание и завершение потока..... | 407 |
| 15.1.2. Синхронизация потоков | 412 |
| 15.2. Функции для работы с потоками, объявленные в файле process.h | 417 |
| 15.3. Потоки POSIX..... | 420 |
| 15.3.1. Создание и завершение потока..... | 420 |
| 15.3.2. Синхронизация потоков | 423 |
| 15.4. Запуск процессов | 426 |
| 15.5. Получение идентификатора процесса..... | 429 |
| Глава 16. Создание библиотек | 430 |
| 16.1. Статические библиотеки | 430 |
| 16.1.1. Создание статической библиотеки из командной строки | 430 |
| 16.1.2. Создание статической библиотеки в редакторе Eclipse | 434 |
| 16.2. Динамические библиотеки..... | 438 |
| 16.2.1. Создание динамической библиотеки из командной строки..... | 438 |
| 16.2.2. Создание динамической библиотеки в редакторе Eclipse..... | 440 |
| 16.2.3. Загрузка динамической библиотеки во время выполнения программы | 443 |
| 16.2.4. Экспортируемые и внутренние функции | 446 |
| 16.2.5. Функция <i>DllMain()</i> | 446 |
| Глава 17. Прочее..... | 448 |
| 17.1. Регистрация функции, выполняемой при завершении программы | 448 |
| 17.2. Выполнение системных команд..... | 449 |
| 17.3. Получение и изменение значений системных переменных | 451 |
| 17.4. Директивы препроцессора | 454 |
| 17.5. Создание значка приложения | 455 |
| Заключение..... | 459 |
| Приложение. Описание электронного архива..... | 461 |
| Предметный указатель | 463 |

Введение

Добро пожаловать в мир языка C!

Язык C — это компилируемый язык программирования высокого уровня, предназначенный для самого широкого круга задач. С его помощью можно обрабатывать различные данные, писать инструкции для микроконтроллеров, создавать драйверы, консольные, нативные и оконные приложения, и даже целые операционные системы. Язык C — настоящая легенда, один из старейших языков программирования в мире, переживший многие другие языки и оказавший влияние на синтаксис современных языков программирования, таких как Java, C#, C++, PHP и др.

C — язык кроссплатформенный, позволяющий создавать программы, которые будут работать во всех операционных системах, но для каждой операционной системы компиляцию нужно выполнять отдельно. В этой книге мы рассмотрим основы языка C применительно к 64-битной операционной системе Windows. Для создания исполняемого файла (EXE-файла) мы воспользуемся компилятором gcc.exe версии 8.2, входящим в состав популярной библиотеки MinGW-W64. Для удобства написания и запуска программы будем пользоваться редактором Eclipse.

Существует несколько стандартов языка C: C90 (ANSI C/ISO C), C99 и C11. Для того чтобы использовать правила конкретного стандарта, нужно в составе команды компиляции указать следующие флаги: `-std=c90`, `-std=c99` или `-std=c11`. Мы не будем указывать эти флаги, т. к. станем изучать современный язык C, который включает возможности стандарта C11. Узнать используемый стандарт языка C внутри программы можно с помощью макроса `__STDC_VERSION__`:

```
printf("%ld\n", __STDC_VERSION__); /* 201710 */
```

При указании флагов `-std=c99` и `-std=c11` результаты будут такими:

```
-std=c99: 199901  
-std=c11: 201112
```

Получить информацию о версии компилятора позволяет макрос `__VERSION__`:

```
printf("%s\n", __VERSION__); /* 8.2.1 20181214 */
```

Давайте рассмотрим структуру книги.

В *главе 1* мы установим необходимое программное обеспечение под Windows: библиотеку MinGW и редактор Eclipse.

Глава 2 является вводной. Мы настроим среду разработки, скомпилируем и запустим первую программу — как из командной строки, так и из редактора Eclipse. Кроме того, вкратце разберемся со структурой программы, а также научимся выводить результат работы программы и получать данные от пользователя.

В *главе 3* мы познакомимся с переменными и типами данных в языке C, а в *главе 4* рассмотрим различные операторы, предназначенные для выполнения определенных действий с данными. Кроме того, в этой главе мы изучим операторы ветвления и циклы, позволяющие изменить порядок выполнения программы.

Глава 5 полностью посвящена работе с числами. Вы узнаете, какие числовые типы данных поддерживает язык C, научитесь применять различные функции, генерировать случайные числа и др.

Глава 6 познакомит с массивами в языке C. Вы научитесь создавать как одномерные массивы, так и многомерные, перебирать элементы массива, сортировать, выполнять поиск значений и др.

Глава 7 полностью посвящена работе с однобайтовыми символами и C-строками. В этой главе вы также научитесь работать с различными кодировками, настраивать локаль, выполнять форматирование строки и осуществлять поиск внутри строки. А в *главе 8* мы рассмотрим работу с широкими символами и L-строками, позволяющими использовать двухбайтовые символы из кодировки Unicode.

Глава 9 познакомит со способами работы с датой и временем.

В *главе 10* вы научитесь создавать пользовательские функции, позволяющие применять код многократно, а также объединять их в модули.

В *главе 11* мы рассмотрим способы поиска ошибок в программе и научимся отлаживать программу в редакторе Eclipse.

Главы 12–14 научат работать с файлами и каталогами, читать и записывать файлы в различных форматах.

Глава 15 познакомит с многопоточными приложениями, позволяющими значительно повысить производительность программы за счет параллельного выполнения задач несколькими потоками управления.

В *главе 16* рассматриваются способы создания статических и динамических библиотек. Статические библиотеки становятся частью программы при компиляции, а динамические библиотеки подгружаются при запуске программы или при ее выполнении.

И наконец, в *главе 17* мы рассмотрим возможность запуска функции при завершении работы приложения, научимся выполнять системные команды и получать значения переменных окружения, изучим директивы препроцессора, а также создадим значок для EXE-файла.

Все листинги из этой книги вы найдете в файле Listings.doc, электронный архив с которым можно загрузить с FTP-сервера издательства "БХВ-Петербург" по ссылке <ftp://ftp.bhv.ru/9785977541169.zip> или со страницы книги на сайте www.bhv.ru (см. приложение).

Желаю приятного изучения и надеюсь, что книга поможет вам реализовать как самые простые, так и самые сложные приложения.

ГЛАВА 1



Установка программ под Windows

Вначале необходимо сделать два замечания.

- ❑ Во-первых, имя пользователя компьютера должно состоять только из латинских букв. Никаких русских букв и пробелов, т. к. многие программы сохраняют различные настройки и временные файлы в каталоге `C:\Users\<Имя пользователя>`. Если имя пользователя содержит русские буквы, то они могут быть искажены до неузнаваемости из-за неправильного преобразования кодировок, и программа не сможет сохранить настройки. Помните, что в разных кодировках русские буквы могут иметь разный код. Разработчики программ в основном работают с английским языком и ничего не знают о проблемах с кодировками, т. к. во всех однобайтовых кодировках и в кодировке UTF-8 коды латинских букв одинаковы. Так что, если хотите без проблем заниматься программированием, то от использования русских букв в имени пользователя лучше отказаться.
- ❑ Во-вторых, имена каталогов и файлов в пути не должны содержать русских букв и пробелов. Допустимы только латинские буквы, цифры, тире, подчеркивание и некоторые другие символы. С русскими буквами та же проблема, что описана в предыдущем пункте. При наличии пробелов в пути обычно требуется дополнительно заключать путь в кавычки. Если этого не сделать, то путь будет обрзан до первого встретившегося пробела. Такая проблема будет возникать при сборке и компиляции программ из командной строки.

Соблюдение этих двух простых правил позволит избежать множества проблем в дальнейшем при сборке и компиляции программ сторонних разработчиков.

1.1. Создание структуры каталогов

Перед установкой программ создадим следующую структуру каталогов:

```
book
  cpp
    eclipse
    projects
    lib
```

Каталоги `book` и `src` лучше разместить в корне какого-либо диска. В моем случае это будет диск `C:`, следовательно, пути к содержимому каталогов — `C:\book` и `C:\src`. Можно создать каталоги в любом другом месте, но в пути не должно быть русских букв и пробелов — только латинские буквы, цифры, тире и подчеркивание. Остальных символов лучше избегать, если не хотите проблем с компиляцией и запуском программ.

В каталоге `C:\book` мы станем размещать наши тестовые программы и тренироваться при изучении работы с файлами и каталогами. Некоторые функции при неправильном действии без проблем могут удалить все дерево каталогов, поэтому для экспериментов мы воспользуемся отдельным каталогом, а не каталогом `C:\src`, в котором у нас будет находиться все сокровенное, — обидно, если по случайности мы удалим все установленные программы и проекты.

Внутри каталога `C:\src` у нас созданы три вложенных каталога:

- `eclipse` — в этот каталог мы установим редактор Eclipse;
- `projects` — в этом каталоге мы станем сохранять проекты из редактора Eclipse;
- `lib` — путь к этому каталогу мы добавим в системную переменную `PATH` и будем размещать в ней различные библиотеки, которые потребуются для наших программ.

1.2. Добавление пути в переменную `PATH`

Когда мы в командной строке вводим название программы без предварительного указания пути к ней, то вначале поиск программы выполняется в текущем рабочем каталоге (обычно это каталог, из которого запускается программа), а затем в путях, указанных в системной переменной `PATH`. Аналогично выполняется поиск библиотек динамической компоновки при запуске программы с помощью двойного щелчка на значке файла, но системные каталоги имеют более высокий приоритет, чем каталоги, указанные в переменной `PATH`. Пути в системной переменной `PATH` просматриваются слева направо до первого нахождения искомого объекта. Так что, если в путях расположено несколько объектов с одинаковыми именами, то мы получим только первый найденный объект. Поэтому, если вдруг запустилась другая программа, следует либо удалить путь, ведущий к другой программе, либо переместить новый путь в самое начало системной переменной `PATH`.

Давайте добавим путь к каталогу `C:\src\lib` в переменную `PATH`. Для того чтобы изменить системную переменную в Windows, переходим в **Параметры | Панель управления | Система и безопасность | Система | Дополнительные параметры системы**. В результате откроется окно **Свойства системы** (рис. 1.1). На вкладке **Дополнительно** нажимаем кнопку **Переменные среды**. В открывшемся окне (рис. 1.2) в списке **Системные переменные** выделяем строку с переменной `Path` и нажимаем кнопку **Изменить**. В следующем окне (рис. 1.3) изменяем значение в поле **Значение переменной** — для этого переходим в конец существующей строки, ставим точку с запятой, а затем вводим путь к каталогу `C:\src\lib`:

```
<Текущее значение>;C:\src\lib
```

Сохраняем изменения.

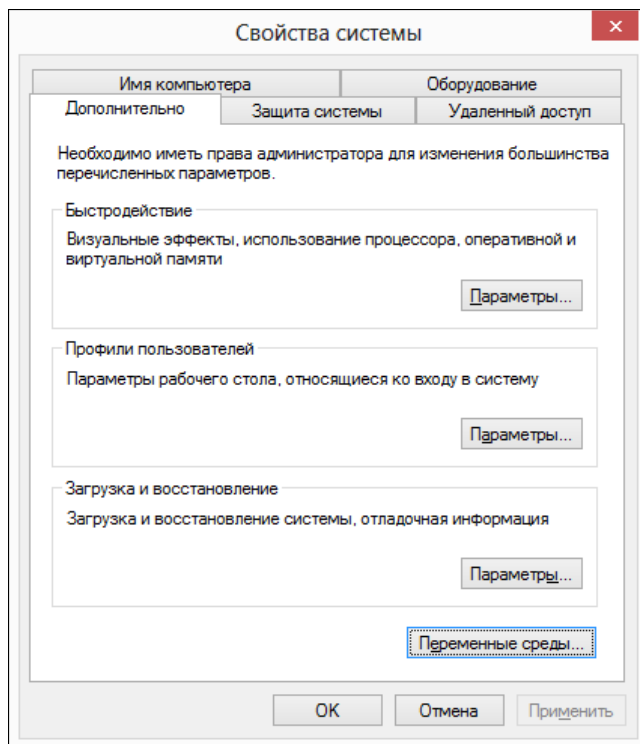


Рис. 1.1. Окно Свойства системы

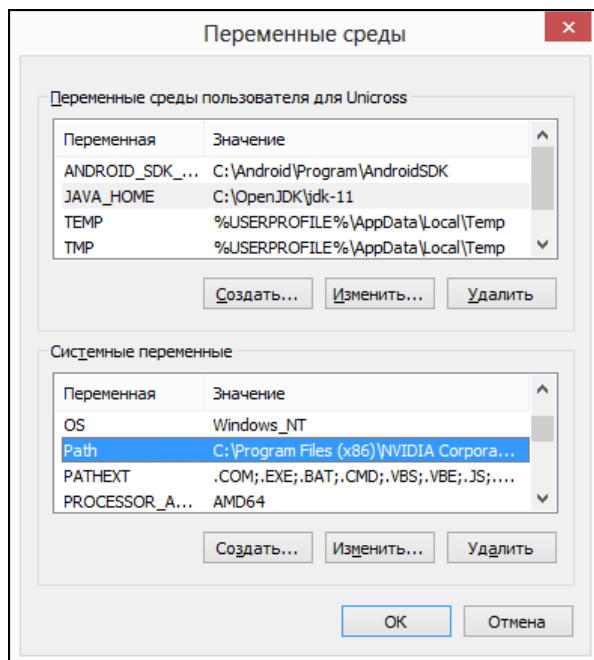


Рис. 1.2. Окно Переменные среды

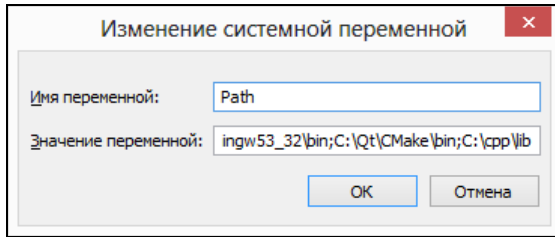


Рис. 1.3. Окно Изменение системной переменной

Добавлять пути в переменную `PATH` мы будем несколько раз, поэтому способ изменения значения этой системной переменной нужно знать наизусть.

ВНИМАНИЕ!

Случайно не удалите существующее значение переменной `PATH`, иначе другие приложения перестанут запускаться.

1.3. Работа с командной строкой

При изучении материала мы часто будем прибегать к помощи приложения Командная строка. Вполне возможно, что вы никогда не пользовались командной строкой и не знаете, как запустить это приложение. Давайте рассмотрим некоторые способы его запуска в Windows:

- через поиск находим приложение Командная строка;
- нажимаем комбинацию клавиш `<Windows>+<R>`. В открывшемся окне вводим `cmd` и нажимаем кнопку **ОК**;
- находим файл `cmd.exe` в каталоге `C:\Windows\System32`;
- в Проводнике щелкаем правой кнопкой мыши на свободном месте списка файлов, удерживая при этом нажатой клавишу `<Shift>`, и из контекстного меню выбираем пункт **Открыть окно команд**;
- в Проводнике в адресной строке вводим `cmd` и нажимаем клавишу `<Enter>`.

В некоторых случаях для выполнения различных команд могут потребоваться права администратора. Для того чтобы запустить командную строку с правами администратора, через поиск находим приложение Командная строка, щелкаем на значке правой кнопкой мыши и затем выбираем пункт **Запуск от имени администратора**.

Запомните способы запуска командной строки. В дальнейшем мы просто будем говорить "запустите командную строку" без уточнения, как это сделать.

1.4. Установка MinGW и MSYS

Язык `C` является компилируемым языком. Для преобразования текстового файла с программой в исполняемый `EXE`-файл потребуется установить на компьютер специальную программу — *компилятор*. Для компиляции примеров из книги мы

воспользуемся бесплатной программой `gcc.exe`, входящей в состав популярной библиотеки MinGW.

Для загрузки библиотеки переходим на сайт <http://www.mingw.org/> и нажимаем кнопку **Download Installer** или переходим по ссылке <http://www.mingw.org/download/installer>. Скачиваем программу установки и запускаем файл `mingw-get-setup.exe`. Обратите внимание: для установки библиотеки потребуется активное подключение к Интернету. После запуска программы установки отобразится окно, показанное на рис. 1.4. Нажимаем кнопку **Install**. На следующем шаге (рис. 1.5) указываем каталог `C:\MinGW` и нажимаем кнопку **Continue**. После скачивания необходимых файлов отобразится окно, показанное на рис. 1.6. Нажимаем кнопку **Continue**. В следующем окне (рис. 1.7) нужно выбрать требуемые компоненты. Устанавливаем флажки **mingw32-base-bin**, **mingw32-gcc-g++-bin** и **msys-base-bin**. Затем в меню **Installation** выбираем пункт **Apply Changes** и в открывшемся окне (рис. 1.8) нажимаем кнопку **Apply**. После установки на последнем шаге (рис. 1.9) нажимаем кнопку **Close**.

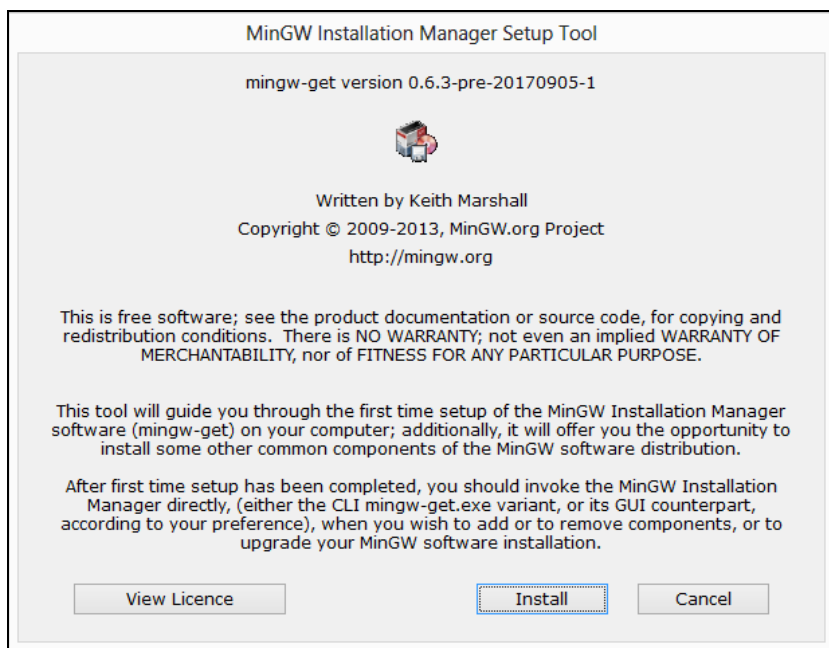


Рис. 1.4. Установка MinGW. Шаг 1

В результате библиотека MinGW будет установлена в каталог `C:\MinGW`. Программу `gcc.exe`, предназначенную для компиляции программ, написанных на языке C, можно найти в каталоге `C:\MinGW\bin`, а заголовочные файлы — в каталоге `C:\MinGW\include`.

Помимо библиотеки MinGW в каталог `C:\MinGW\msys` была установлена библиотека MSYS, которую редакторы используют для сборки проектов.

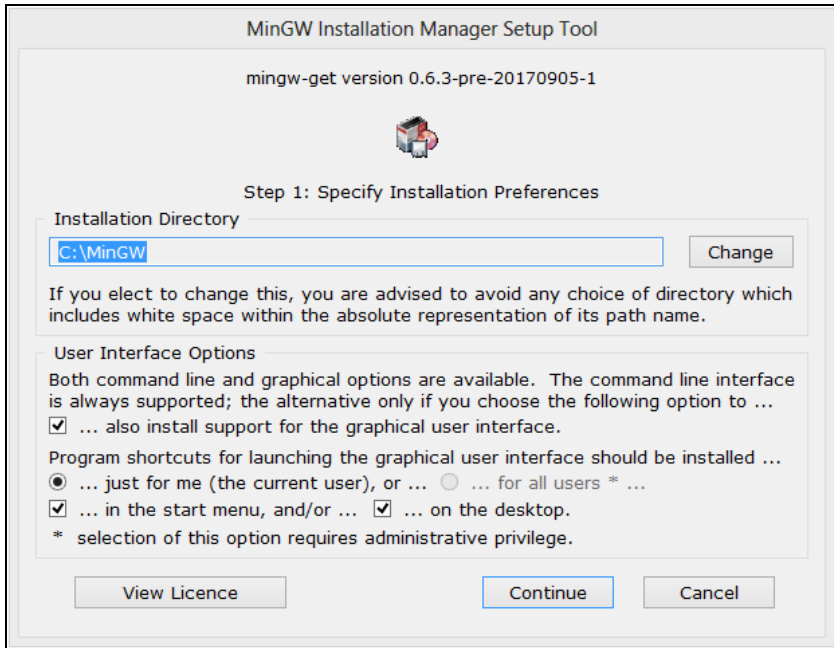


Рис. 1.5. Установка MinGW. Шаг 2

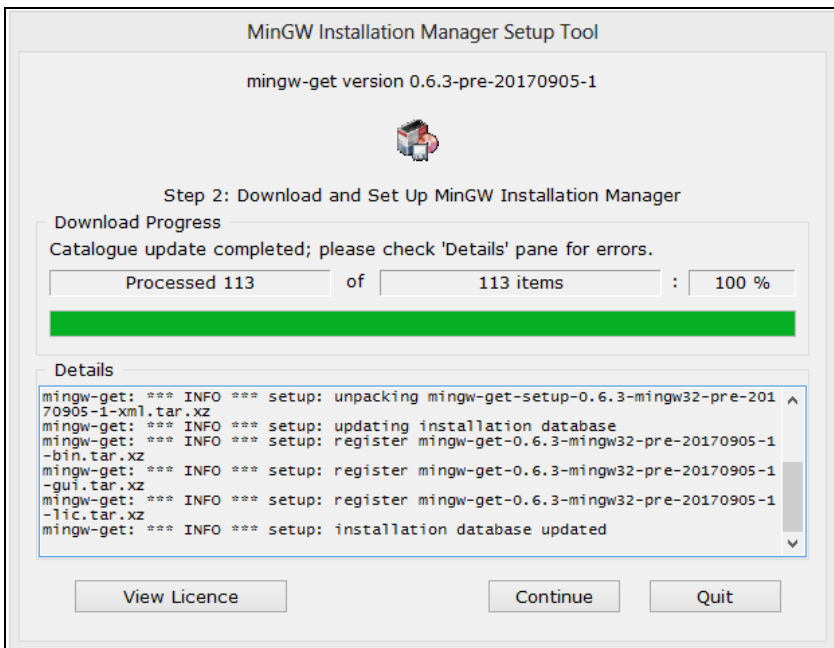


Рис. 1.6. Установка MinGW. Шаг 3

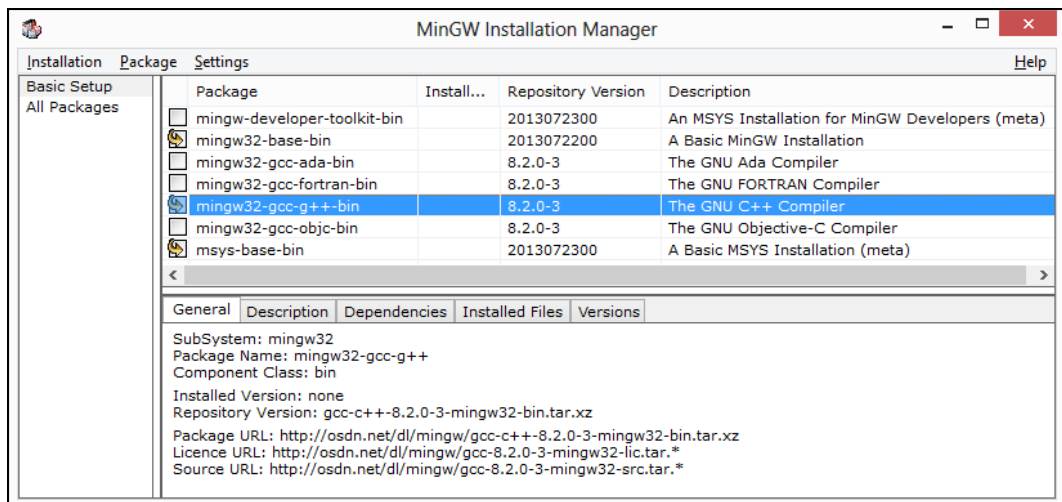


Рис. 1.7. Установка MinGW. Шаг 4

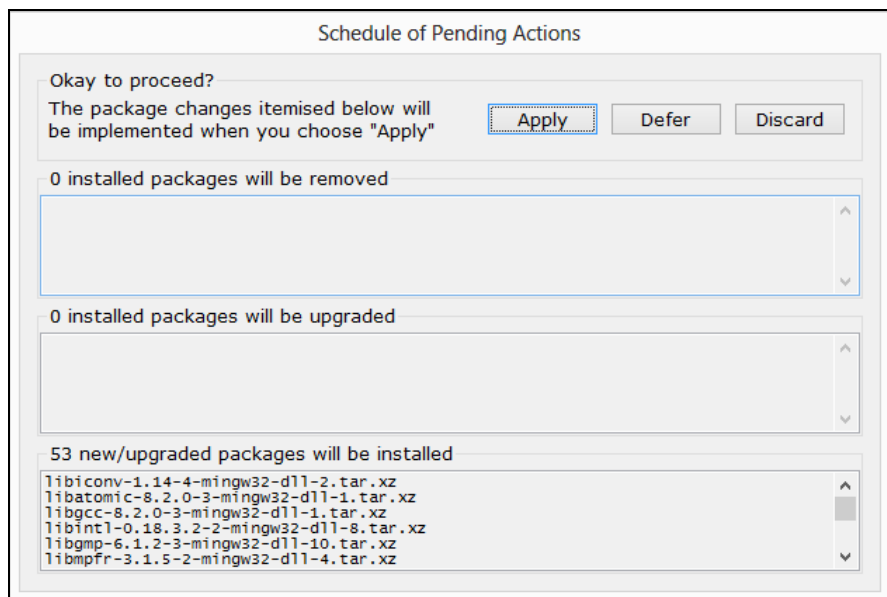


Рис. 1.8. Установка MinGW. Шаг 5

Пути к каталогам `C:\MinGW\bin` и `C:\MinGW\msys\1.0\bin` можно добавить в системную переменную `PATH`. Однако мы этого делать не станем, т. к. установим несколько версий компилятора. Вместо изменения переменной `PATH` на постоянной основе мы будем выполнять изменение в командной строке только для текущего сеанса. Продемонстрируем это на примере, а заодно проверим работоспособность компилятора. Запускаем командную строку и выполняем следующие команды:

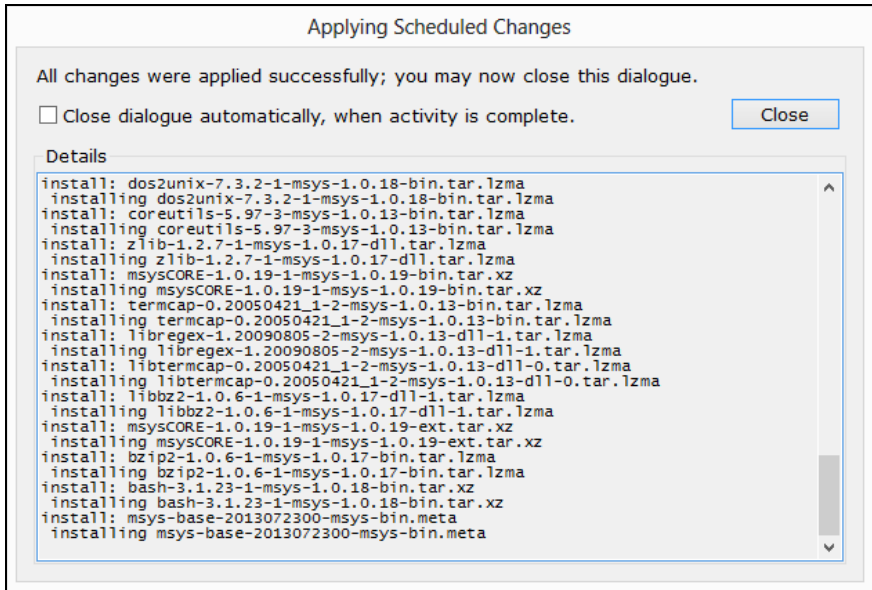


Рис. 1.9. Установка MinGW. Шаг 6

```
C:\Users\Unicross>cd C:\
```

```
C:\>set Path=C:\MinGW\bin;C:\MinGW\msys\1.0\bin;%Path%
```

```
C:\>gcc --version
```

```
gcc (MinGW.org GCC-8.2.0-3) 8.2.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

```
C:\>g++ --version
```

```
g++ (MinGW.org GCC-8.2.0-3) 8.2.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

Первая команда (`cd C:\`) делает текущим корневым каталогом диска C:. Вторая команда (`set Path=C:\MinGW\bin;C:\MinGW\msys\1.0\bin;%Path%`) изменяет значение системной переменной `PATH` для текущего сеанса. Пути к каталогам `C:\MinGW\bin` и `C:\MinGW\msys\1.0\bin` мы добавили в самое начало переменной `PATH`. Третья команда (`gcc --version`) выводит версию программы `gcc.exe`. Эту программу мы будем использовать для компиляции программ, написанных на языке C. Четвертая команда (`g++ --version`) выводит версию программы `g++.exe`. Эту программу можно использовать для компиляции программ, написанных на языке C++. Фрагменты

перед командами означают приглашение для ввода команд. Текст после команд является результатом выполнения этих команд.

Вместо отдельных команд можно написать скрипт, который выполняет сразу несколько команд и отображает результат их работы в отдельном окне. Запускаться такой скрипт будет с помощью двойного щелчка левой кнопкой мыши на значке файла. Для создания файла потребуется текстовый редактор, позволяющий корректно работать с различными кодировками. Советую установить на компьютер редактор Notepad++. Скачать редактор можно абсолютно бесплатно со страницы <https://notepad-plus-plus.org/>. Из двух вариантов (архив и инсталлятор) советую выбрать именно инсталлятор, т. к. при установке можно будет указать язык интерфейса программы. Установка Notepad++ предельно проста и в комментариях не нуждается.

Запускаем Notepad++ и создаем новый документ. Консоль в Windows по умолчанию работает с кодировкой windows-866, поэтому мы должны и файл сохранить в этой кодировке, иначе русские буквы будут искажены. В меню **Кодировки** выбираем пункт **Кодировки | Кириллица | OEM 866**. Вводим текст скрипта (листинг 1.1) и сохраняем под названием script.bat в каталоге C:\book. Запускаем скрипт с помощью двойного щелчка левой кнопкой мыши на значке файла script.bat. Результат выполнения показан на рис. 1.10. Для закрытия окна консоли достаточно нажать любую клавишу.

Листинг 1.1. Содержимое файла script.bat

```
@echo off
title Заголовок окна
cd C:\
echo Текущий каталог: %CD%
@echo.
set Path=C:\MinGW\bin;C:\MinGW\msys\1.0\bin;%Path%
rem Вывод версии gcc.exe
echo gcc --version
@echo.
gcc --version
rem Вывод версии g++.exe
echo g++ --version
@echo.
g++ --version
pause
```

Рассмотрим инструкции из этого скрипта:

- title — позволяет вывести текст в заголовок окна консоли;
- echo — выводит текст в окно консоли;
- @echo. — выводит пустую строку в окно консоли;
- %CD% — переменная, содержащая путь к текущему рабочему каталогу;

```

Текущий каталог: C:\
gcc --version

gcc (MinGW.org GCC-8.2.0-3) 8.2.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

g++ --version

g++ (MinGW.org GCC-8.2.0-3) 8.2.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Для продолжения нажмите любую клавишу . . .

```

Рис. 1.10. Результат выполнения программы из листинга 1.1

- %Path% — переменная, содержащая значение системной переменной PATH;
- rem — вставляет комментарий, поясняющий фрагмент кода;
- pause — ожидает ввода любого символа от пользователя. Если эту инструкцию убрать из скрипта, то программа выполнится, и окно консоли сразу закроется, не дав нам возможности увидеть результат.

1.5. Установка MinGW-W64

Компилятор gcc.exe, установленный в предыдущем разделе в каталог C:\MinGW\bin, позволяет создавать 32-разрядные приложения. Такие приложения будут запускаться и в 32-битной операционной системе, и в 64-битной. Для того чтобы иметь возможность создавать приложения для 64-битной системы, нужно дополнительно установить библиотеку MinGW-W64.

Для загрузки библиотеки MinGW-W64 переходим на страницу <https://sourceforge.net/projects/mingw-w64/> и скачиваем файл mingw-w64-install.exe. После запуска программы установки отобразится окно, показанное на рис. 1.11. Нажимаем кнопку **Next**. В следующем окне (рис. 1.12) из списка **Version** выбираем самую новую версию (в моем случае **8.1.0**), из списка **Architecture** — пункт **x86_64**, из списка **Threads** — пункт **win32**, а из списка **Exception** — пункт **seh**. Нажимаем кнопку **Next**. На следующем шаге (рис. 1.13) задаем путь C:\MinGW64 и нажимаем кнопку **Next**. Начнется установка библиотеки. В следующем окне (рис. 1.14) нажимаем кнопку **Next**, а на последнем шаге (рис. 1.15) — кнопку **Finish**.

В состав MinGW-W64 не входит MSYS. Программа из каталога C:\MinGW\msys настроена на библиотеку MinGW, расположенную в каталоге C:\MinGW, поэтому мы не сможем ее использовать совместно с MinGW-W64. Однако мы можем скопировать каталог C:\MinGW\msys со всем содержимым и вставить его в каталог



Рис. 1.11. Установка MinGW-W64. Шаг 1

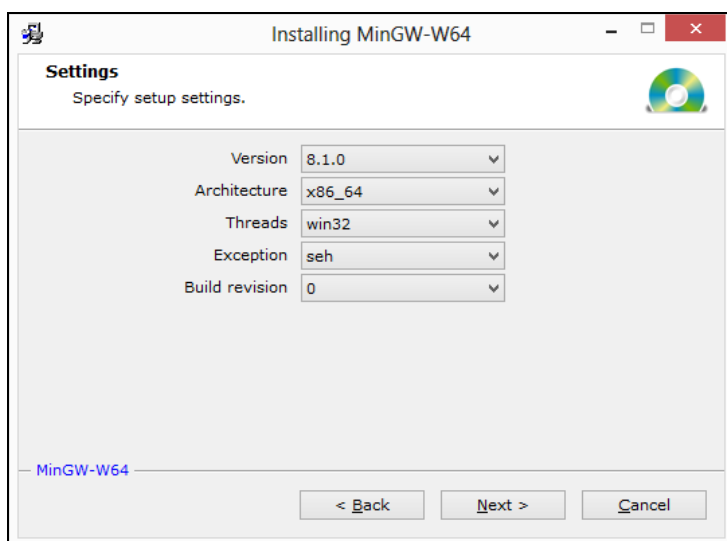


Рис. 1.12. Установка MinGW-W64. Шаг 2

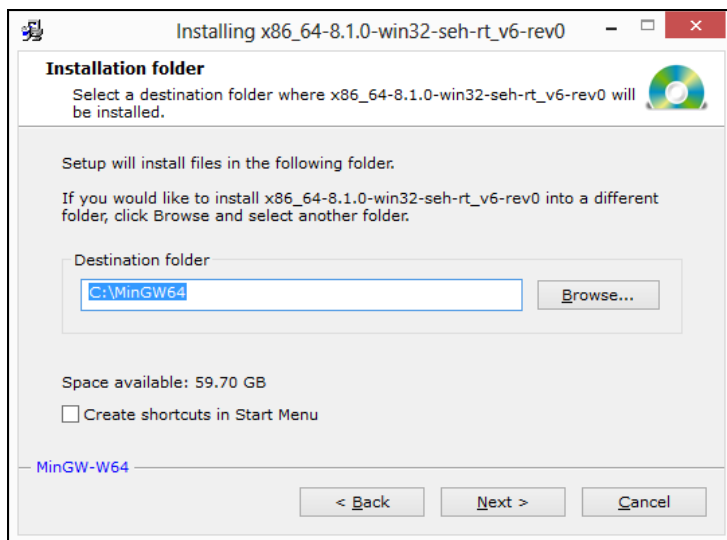


Рис. 1.13. Установка MinGW-W64. Шаг 3

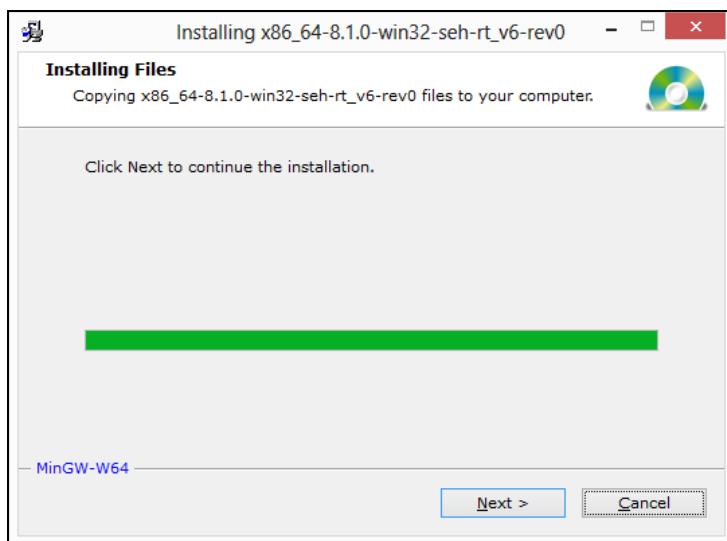


Рис. 1.14. Установка MinGW-W64. Шаг 4

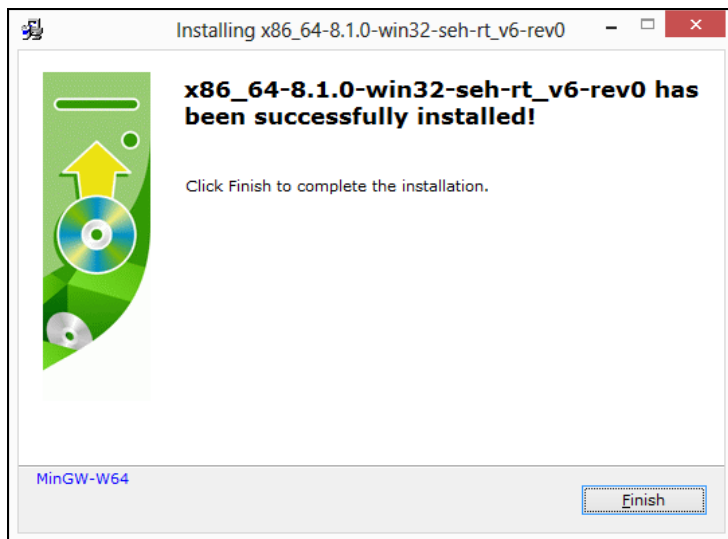


Рис. 1.15. Установка MinGW-W64. Шаг 5

C:\MinGW64\mingw64\msys, а затем указать в конфигурации путь к MinGW-W64. Так мы и сделаем. В результате путь до файла `make.exe` должен быть таким: C:\MinGW64\mingw64\msys\1.0\bin\make.exe. Далее с помощью программы Notepad++ открываем файл C:\MinGW64\mingw64\msys\1.0\etc\fstab и находим строку:

```
C:/MinGW          /mingw
```

Изменяем ее следующим образом и сохраняем файл:

```
C:/MinGW64/mingw64          /mingw
```

Проверим правильность установки MinGW-W64. Открываем командную строку и выполняем следующие команды:

```
C:\Users\Unicross>cd C:\
```

```
C:\>set Path=C:\MinGW64\mingw64\bin;%Path%
```

```
C:\>gcc --version
```

```
gcc (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
C:\>g++ --version
```

```
g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Теперь установим компилятор для создания 32-битных приложений. Запускаем файл `mingw-w64-install.exe` еще раз. На втором шаге (см. рис. 1.12) из списка **Version** выбираем самую новую версию (в моем случае **8.1.0**), из списка **Architecture** — пункт **i686**, из списка **Threads** — пункт **win32**, а из списка **Exception** — пункт **dwarf**. На третьем шаге (см. рис. 1.13) задаем путь `C:\MinGW32`. После установки копируем каталог `msys` со всем содержимым из каталога `C:\MinGW` в каталог `C:\MinGW32\mingw32`. Далее с помощью программы Notepad++ открываем файл `C:\MinGW32\mingw32\msys\1.0\etc\fstab` и находим строку:

```
C:/MinGW                /mingw
```

Изменяем ее следующим образом и сохраняем файл:

```
C:/MinGW32/mingw32    /mingw
```

Проверим правильность установки. Открываем командную строку и выполняем следующие команды:

```
C:\Users\Unicross>cd C:\
```

```
C:\>set Path=C:\MinGW32\mingw32\bin;%Path%
```

```
C:\>gcc --version
```

```
gcc (i686-win32-dwarf-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE.
```

```
C:\>g++ --version
```

```
g++ (i686-win32-dwarf-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE.
```

1.6. Установка MSYS2 и MinGW-W64

Существует еще один способ установки компилятора. Предварительно нам нужно установить библиотеку MSYS2. Переходим на сайт <http://www.msys2.org/>, скачиваем файл `msys2-x86_64-20180531.exe` и запускаем его. В открывшемся окне (рис. 1.16) нажимаем кнопку **Далее**. В следующем окне (рис. 1.17) указываем каталог `C:\msys64` и нажимаем кнопку **Далее**. На следующем шаге (рис. 1.18) нажимаем кнопку **Далее**. После установки нажимаем кнопку **Далее** (рис. 1.19), а затем кнопку **Завершить** (рис. 1.20).

В итоге библиотека MSYS2 будет установлена в каталог `C:\msys64`. В этом каталоге расположены скрипты для запуска: `msys2.exe`, `mingw32.exe` и `mingw64.exe`.

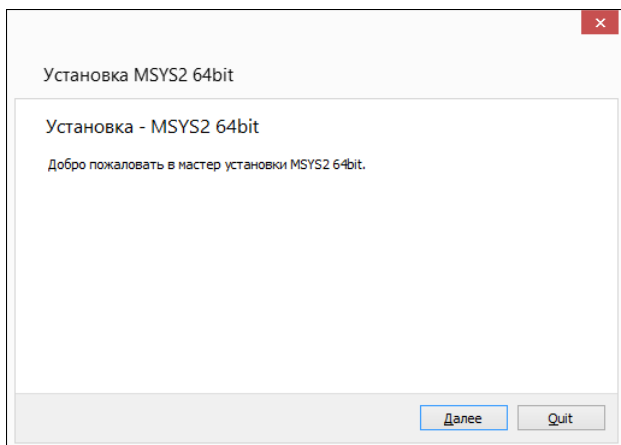


Рис. 1.16. Установка MSYS2. Шаг 1

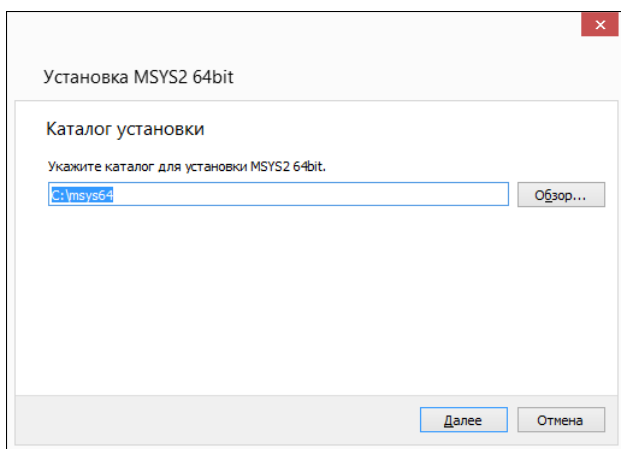


Рис. 1.17. Установка MSYS2. Шаг 2

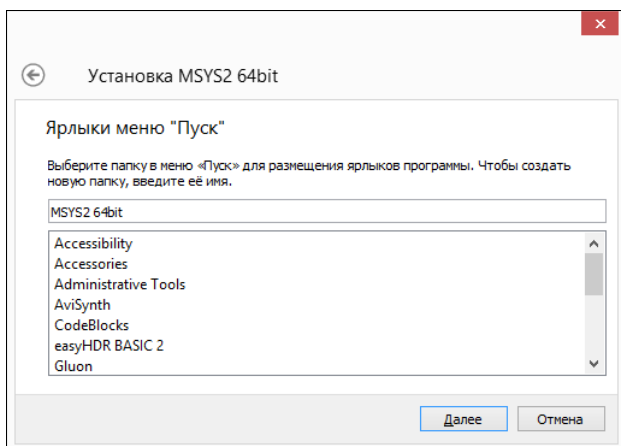


Рис. 1.18. Установка MSYS2. Шаг 3

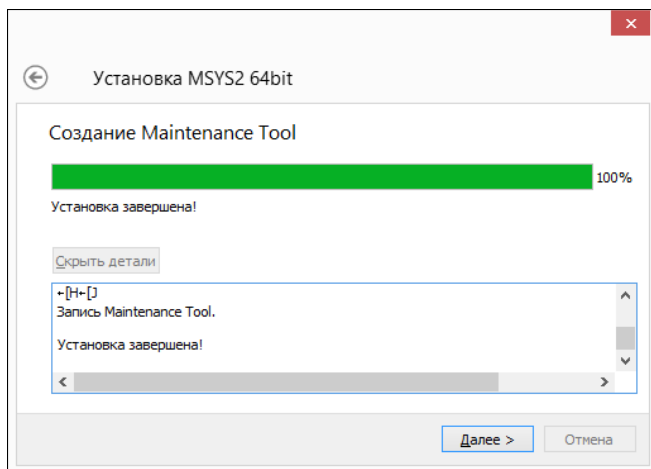


Рис. 1.19. Установка MSYS2. Шаг 4

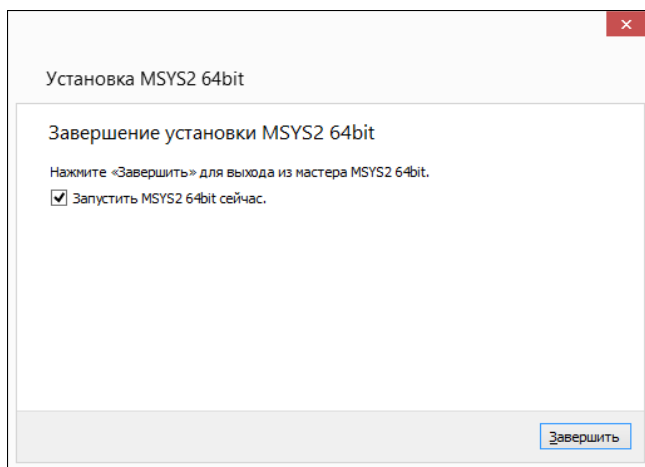


Рис. 1.20. Установка MSYS2. Шаг 5

Файл `msys2.exe` запускает командную строку, в которой мы можем установить различные библиотеки, выполнив специальные команды. Если на последнем шаге при установке вы не сбросили флажок, то это приложение запустится автоматически. Если флажок сбросили, то запускаем программу `msys2.exe` с помощью двойного щелчка левой кнопкой мыши на значке файла. Сначала обновим программу, выполнив команду (рис. 1.21):

```
расман -Su
```

На запрос подтверждения установки вводим букву `Y` и нажимаем клавишу `<Enter>`. После завершения установки закрываем окно, а затем запускаем программу `msys2.exe` снова. Вводим следующую команду:

```
расман -Su
```

На запрос подтверждения установки вводим букву `Y` и нажимаем клавишу `<Enter>`.

```

Unicross@ASUS MSYS ~
$ pacman -Syu
:: Обновление баз данных пакетов...
mingw32                534,8 KiB   824K/s  00:01 [#####] 100%
mingw32.sig            119,0 B     0,00B/s  00:00 [#####] 100%
mingw64                535,8 KiB   282K/s  00:02 [#####] 100%
mingw64.sig            119,0 B     0,00B/s  00:00 [#####] 100%
msys                   178,3 KiB   7,92M/s  00:00 [#####] 100%
msys.sig               119,0 B     0,00B/s  00:00 [#####] 100%
:: Starting core system upgrade...
предупреждение: terminate other MSYS2 programs before proceeding
разрешение зависимостей...
проверка конфликтов...

Пакеты (6) bash-4.4.023-1 filesystem-2018.12-1 mintty-1~2.9.5-1
             msys2-runtime-2.11.2-1  pacman-5.1.2-2  pacman-mirrors-20180604-2

Будет загружено: 19,04 MiB
Будет установлено: 68,25 MiB
Изменение размера: 11,96 MiB

:: Приступить к установке? [Y/n] |

```

Рис. 1.21. Программа msys2.exe

Теперь можно установить библиотеку MinGW-W64. Для этого в окне вводим следующую команду:

```
pacman -S mingw-w64-x86_64-toolchain
```

Для установки всех компонентов нажимаем клавишу <Enter>, а затем на запрос подтверждения установки вводим букву **Y** и нажимаем клавишу <Enter>. Библиотека MinGW-W64 будет установлена в каталог `C:\msys64\mingw64`. Давайте проверим установку, выполнив в командной строке следующие команды:

```
C:\Users\Unicross>cd C:\
```

```
C:\>set Path=C:\msys64\mingw64\bin;%Path%
```

```
C:\>gcc --version
```

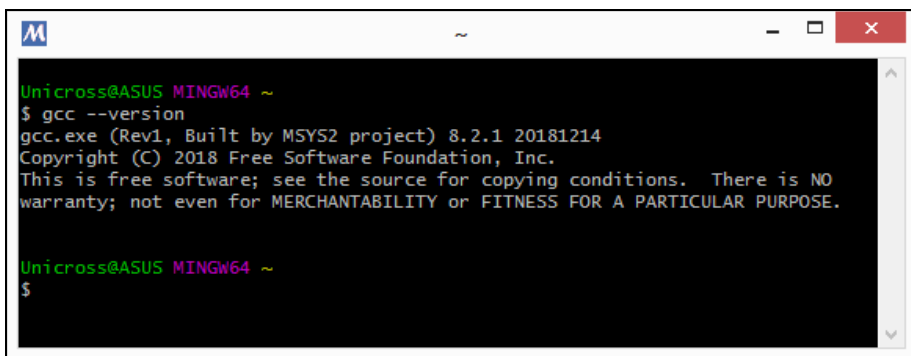
```
gcc (Rev1, Built by MSYS2 project) 8.2.1 20181214
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

```
C:\>g++ --version
```

```
g++ (Rev1, Built by MSYS2 project) 8.2.1 20181214
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

В командной строке мы настраивали переменную окружения `PATH`. Если запустить программу `mingw64.exe`, то переменные окружения дополнительно настраивать не нужно, достаточно сразу выполнить команду (рис. 1.22):

```
Unicross@ASUS MINGW64 ~
$ gcc --version
gcc.exe (Rev1, Built by MSYS2 project) 8.2.1 20181214
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```



```
Unicross@ASUS MINGW64 ~
$ gcc --version
gcc.exe (Rev1, Built by MSYS2 project) 8.2.1 20181214
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Unicross@ASUS MINGW64 ~
$
```

Рис. 1.22. Программа `mingw64.exe`

Удобство использования библиотеки `MSYS2` заключается в том, что помимо установки какой-либо библиотеки, причем всего лишь одной командой, дополнительно устанавливаются все зависимости.

Давайте установим еще несколько компонентов, которые могут пригодиться:

```
pacman -S make
pacman -S mingw-w64-x86_64-cmake
pacman -S mingw-w64-x86_64-gtk3
pacman -S mingw-w64-x86_64-glade
```

Проверим установку библиотеки `GTK+` (она предназначена для разработки оконных приложений на языке C). Запускаем программу `C:\msys64\mingw64.exe` и выполняем следующую команду:

```
pkg-config --libs gtk+-3.0
```

Результат:

```
-LC:/msys64/mingw64/lib -lgtk-3 -lgdk-3 -lgdi32 -limm32 -lshell32 -lole32
-Wl,-luuid -lwinmm -ldwmapi -lsetupapi -lcfgmgr32 -lz -lpangowin32-1.0
-lpangocairo-1.0 -lpango-1.0 -latk-1.0 -lcairo-gobject -lcairo
-lgdk_pixbuf-2.0 -lgio-2.0 -lgobject-2.0 -lglib-2.0 -lintl
```

Все установленные библиотеки скомпилированы под 64-битные операционные системы. Для установки 32-битных версий библиотек нужно в команде заменить

фрагмент x86_64 фрагментом i686. Пример команды для установки 32-битного компилятора:

```
pacman -S mingw-w64-i686-gcc
```

Компилятор будет установлен в каталог `C:\msys64\mingw32`. Для выполнения команд можно воспользоваться программой `C:\msys64\mingw32.exe`. Например, получим версию компилятора:

```
$ gcc --version
gcc.exe (Rev1, Built by MSYS2 project) 7.4.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

Обратите внимание: мы установили только компилятор. Для того чтобы установить все компоненты, нужно воспользоваться командой:

```
pacman -S mingw-w64-i686-toolchain
```

Мы будем учиться создавать приложения на языке C под 64-битные операционные системы, поэтому можно все компоненты для 32-битного компилятора дополнительно не устанавливать, тем более что мы уже установили более новую версию компилятора в *разд. 1.5*.

1.7. Установка и настройка редактора Eclipse

В этой книге для создания и компиляции программ на языке C мы воспользуемся редактором Eclipse. Редактор удобен в использовании, он позволяет автоматически закончить слово, подчеркнуть код с ошибкой, подсветить код программы, вывести список всех функций, отладить программу, а также скомпилировать все файлы проекта всего лишь нажатием одной кнопки без необходимости использования командной строки.

Для загрузки редактора Eclipse переходим на страницу: <https://www.eclipse.org/downloads/packages/> и скачиваем архив с программой из раздела **Eclipse IDE for C/C++ Developers**. В моем случае архив называется `eclipse-cpp-2018-12-R-win32-x86_64.zip`. Распаковываем скачанный архив и копируем каталог `eclipse` с файлами редактора в каталог `C:\cpp`. Редактор не нуждается в установке, поэтому просто переходим в каталог `C:\cpp\eclipse` и запускаем файл `eclipse.exe`. Однако не торопитесь прямо сейчас запускать редактор.

Редактор Eclipse написан на языке Java, поэтому предварительно нужно установить на компьютер либо *Java Runtime Environment (JRE)*, либо *Java Development Kit (JDK)*. JRE используется для запуска приложений, а JDK — как для запуска, так и для разработки приложений на языке Java. Для того чтобы узнать, какая версия Java требуется для работы редактора Eclipse, открываем конфигурационный файл `C:\cpp\eclipse\eclipse.ini` и смотрим значение опции `requiredJavaVersion`. В моем случае опция имеет такое значение:

```
-Dosgi.requiredJavaVersion=1.8
```

Значение 1.8 указывает на то, что нужна Java 8. Устанавливаем на компьютер требуемую версию Java и задаем путь к ней с помощью опции `-vm`. Самый простой способ — создать ярлык для `eclipse.exe`, отобразить окно **Свойства** и в поле **Объект** указать такое значение (замените фрагмент `<Путь к JRE>` реальным значением):

```
C:\cpp\eclipse\eclipse.exe -vm "<Путь к JRE>\bin\server\jvm.dll"
```

Можно использовать и более новую версию, например Java 10, но не выше, т. к. в версии 11 некоторые пакеты были удалены. Пример указания пути к Java 10:

```
C:\cpp\eclipse\eclipse.exe  
-vm "C:\Program Files\Java\jre-10\bin\server\jvm.dll"
```

ПРИМЕЧАНИЕ

Все способы указания значения подробно описаны на странице <http://wiki.eclipse.org/Eclipse.ini>.

При запуске редактор попросит указать каталог с рабочим пространством (рис. 1.23). Указываем `C:\cpp\projects` и нажимаем кнопку **Launch**.

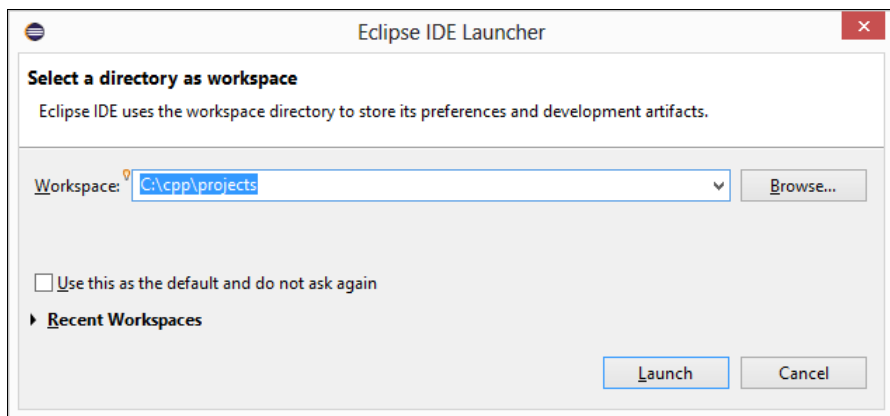


Рис. 1.23. Окно Eclipse IDE Launcher

Прежде чем пользоваться редактором, нужно изменить некоторые настройки по умолчанию. Начнем с кодировки файлов, в которых мы будем вводить текст программы. Для того чтобы указать кодировку по умолчанию для всех файлов, в меню **Window** выбираем пункт **Preferences**. В открывшемся окне переходим в раздел **General | Workspace** (рис. 1.24). В группе **Text file encoding** устанавливаем флажок **Other** и в текстовое поле вводим `cp1251`. Сохраняем изменения. Таким образом, мы будем пользоваться кодировкой `windows-1251`, которая по умолчанию используется в русской версии Windows.

Если необходимо изменить кодировку уже открытого файла, в меню **Edit** выбираем пункт **Set Encoding**. Если же потребуется указать другую кодировку для всех файлов проекта, в меню **Project** выбираем пункт **Properties**. В открывшемся окне

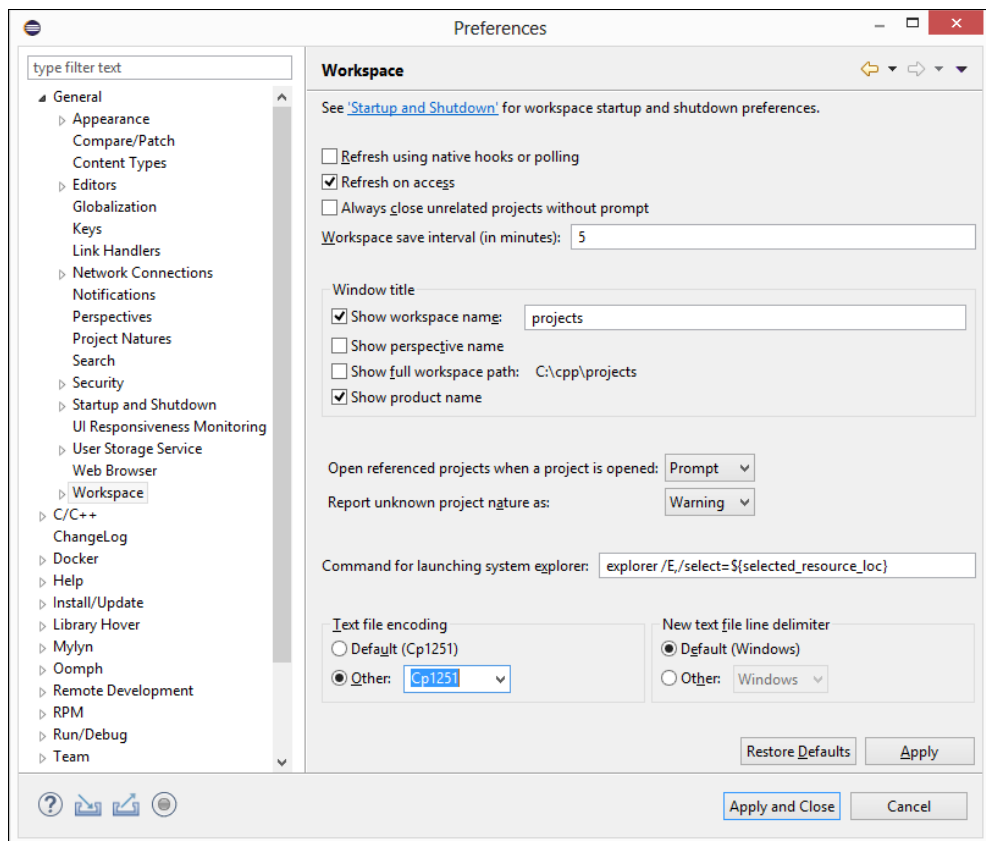


Рис. 1.24. Указание кодировки файлов по умолчанию

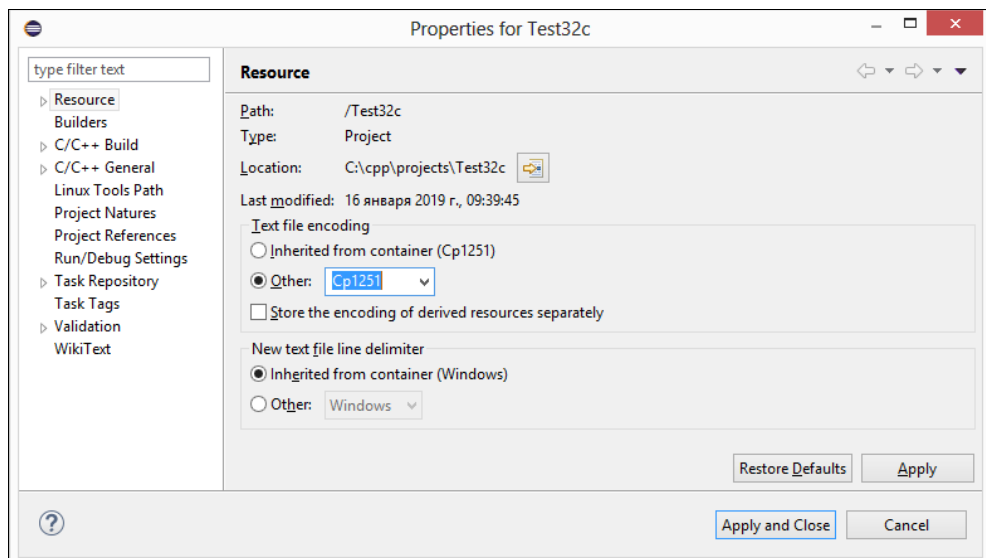


Рис. 1.25. Указание кодировки файлов для проекта

(рис. 1.25) слева в списке выбираем пункт **Resource**, а затем справа в группе **Text file encoding** указываем нужную кодировку.

По умолчанию редактор вместо пробелов вставляет символы табуляции. Нас это не устраивает. Давайте изменим настройку форматирования кода. Для этого в меню **Window** выбираем пункт **Preferences**. В открывшемся окне переходим в раздел **C/C++ | Code Style | Formatter** (рис. 1.26). Нажимаем кнопку **New**. В открывшемся окне (рис. 1.27) в поле **Profile name** вводим название стиля, например **MyStyle**, а из списка выбираем пункт **K&R [built-in]**. Нажимаем кнопку **OK**. Откроется окно (рис. 1.28), в котором можно изменить настройки нашего стиля. На вкладке **Indentation** из списка **Tab policy** выбираем пункт **Spaces only**, а в поля **Indentation size** и **Tab size** вводим число 3. Сохраняем все изменения.

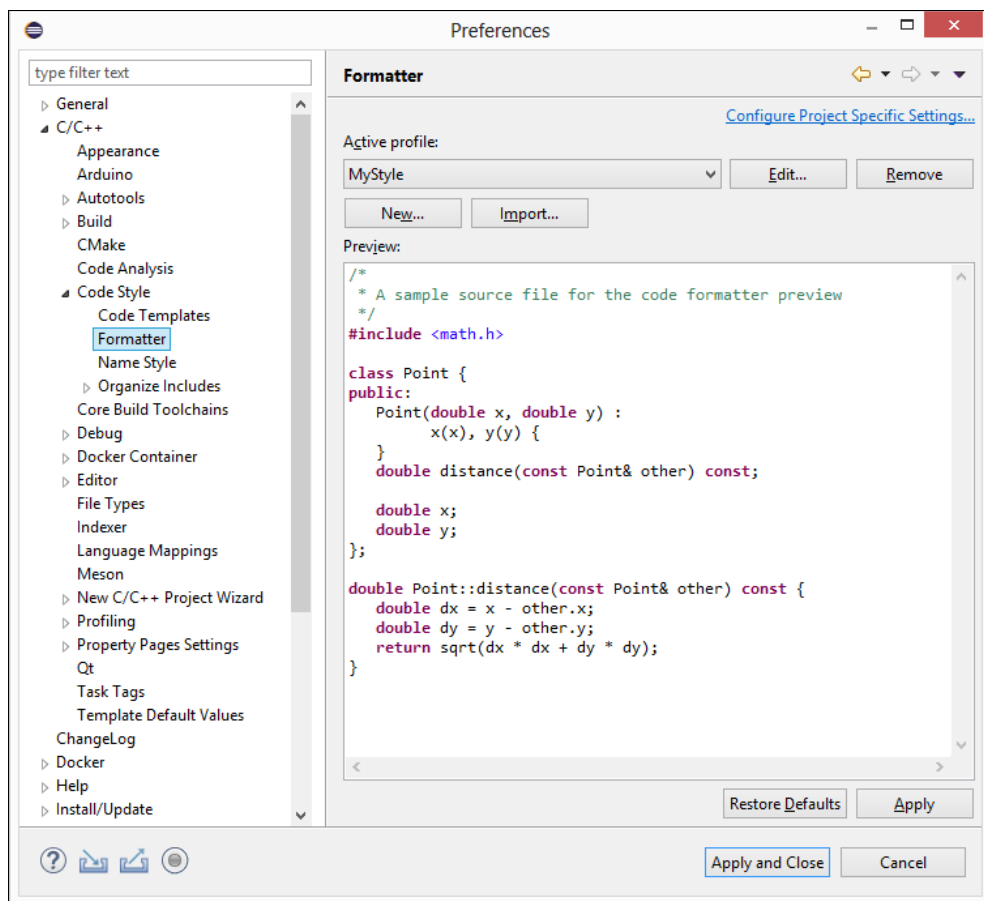


Рис. 1.26. Окно Preferences, раздел Formatter

Если необходимо изменить размер шрифта, то в меню **Window** выбираем пункт **Preferences**. В открывшемся окне переходим в раздел **General | Appearance | Colors and Fonts** (рис. 1.29). Из списка выбираем пункт **C/C++ | Editor | C/C++ Editor Text Font** и нажимаем кнопку **Edit**.

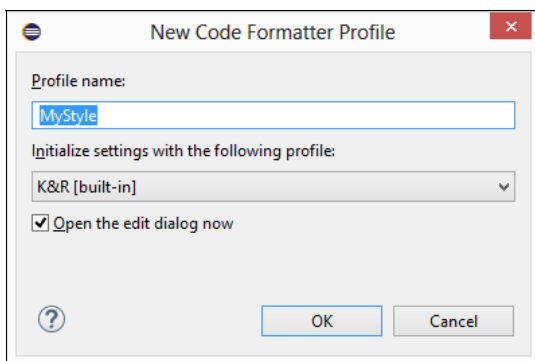


Рис. 1.27. Окно New Code Formatter Profile

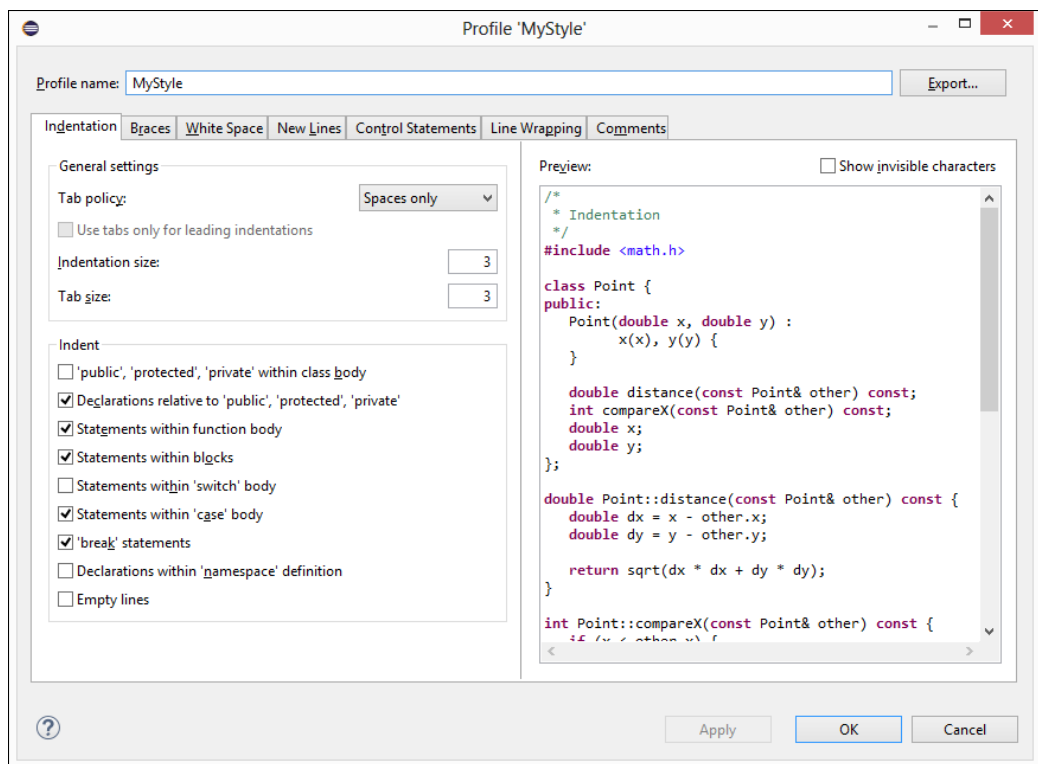


Рис. 1.28. Изменение настроек форматирования

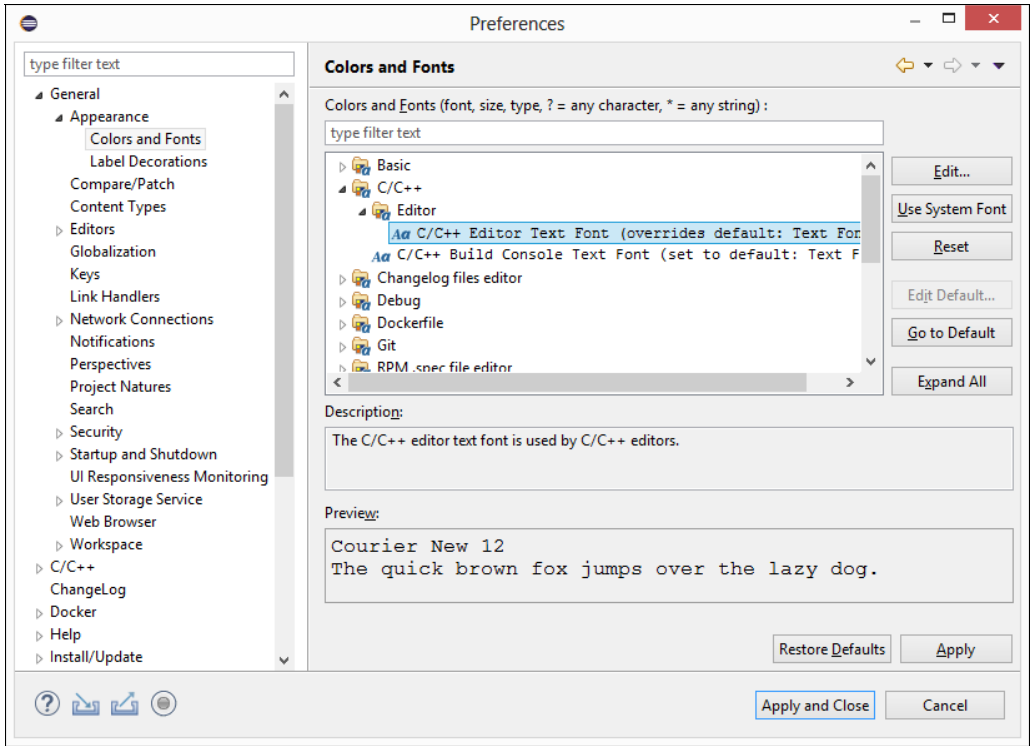


Рис. 1.29. Окно Preferences, раздел Colors and Fonts

1.8. Создание проектов в редакторе Eclipse

Давайте в редакторе Eclipse создадим два проекта. Первый проект (с названием Test32c) мы будем использовать для создания 32-битных программ (компилятор из каталога C:\MinGW32\mingw32\bin), а второй проект (с названием Test64c) — для создания 64-битных программ (компилятор из каталога C:\msys64\mingw64\bin).

Для создания проекта в меню **File** выбираем пункт **New | Project**. В открывшемся окне (рис. 1.30) в списке выбираем пункт **C/C++ | C Project** и нажимаем кнопку **Next**. На следующем шаге (рис. 1.31) в поле **Project name** вводим Test32c, в списке слева выбираем пункт **Executable | Hello World ANSI C Project**, а в списке справа — пункт **MinGW GCC**. Если в списке справа отсутствует пункт **MinGW GCC**, то редактор не смог автоматически обнаружить библиотеку. В этом случае следует сбросить флажок **Show project types and toolchains**, и пункт станет доступен. После нажатия кнопки **Next** отобразится окно (рис. 1.32), в котором можно ввести имя автора, информацию об авторских правах, текст, который будет выведен на консоль, и каталог, в который будут сохраняться файлы. Нажимаем кнопку **Next**. В следующем окне (рис. 1.33) нажимаем кнопку **Finish** для создания проекта.

В результате в каталоге C:\cpp\projects будет создан каталог Test32c, содержащий файлы проекта. В каталоге C:\cpp\projects\Test32c\src можно найти файл