

# 1. ВВЕДЕНИЕ

## 1.1. Что нужно знать

Предполагается, что читатель этой методички должен иметь хотя бы начальные пользовательские навыки при работе с GNU/Linux (умение вводить команды в оболочку и знать основы операционной системы), а также знать язык C хотя бы на начальном уровне. Я посчитал бессмысленным писать еще одну методичку по языку C. Если вы знаете язык C на уровне книги Кернигана и Ритчи "Язык программирования Си", то у вас не будет проблем с изучением материала этой методички.

Если вы знаете язык C++, но не знаете C, ничего страшного. Эти два языка хотя и разные по своей сути, но имеют одну и ту же синтаксическую основу. C++-программисту не надо обладать феноменальной сообразительностью, чтобы разобраться в программе, написанной на C.

Большая часть материала этой методички относится и к другим современным Unix-подобным операционным системам (например, FreeBSD). Если вы еще не знаете Linux, то больших проблем не будет.

## 1.2. Условные обозначения

В методичке используется достаточно мало условных обозначений. Листинги программ, а также вырезки из листингов и содержимое конфигурационных или иных файлов выделяются синим цветом:

```
#include <stdio.h>

int main (void)
{
    printf ("Hello World\n");
}
```

Примеры команд, вводимых в оболочку выделяются оранжевым цветом и начинаются с символа \$ (приглашение командной строки). Вывод команд выделяется жирным шрифтом:

```
$ gcc
gcc: no input files
$
```

Примеры команд, вводимых от имени суперпользователя (root) выделяются так же, как и обычные команды, только в качестве приглашения командной строки используется символ #

```
# ls
hello.c
#
```

Некоторые вещи в основном тексте, на которые я хотел бы обратить ваше внимание, будут выделяться **жирным шрифтом**. Некоторые термины, которые следует запомнить, будут выделяться *курсивом*. Кроме того, жирным шрифтом могут быть отделены от основного текста имена файлов, имена функций и т. п.

### 1.3. Что нужно иметь

То, что нам нужно иметь, называют инструментарием. Инструментарий программиста зависит от того, какие программы и на каких языках он будет писать. Наш инструментарий будет состоять из: 1) Операционной системы Linux (GNU/Linux) с текстовым редактором; 2) Компилятора с линковщиком; 3) Дополнительных утилит. Ну и конечно же из *терпения* и *усердия*.

Для начала нужно иметь Linux (здесь правильнее будет сказать GNU/Linux). Подойдет практически любой современный дистрибутив. Посмотрите, какая версия ядра Linux у вас используется. Наберите в оболочке следующую команду:

```
$ uname -r  
2.6.28-11-generic  
$
```

У меня ядро версии 2.6.28. У вас, возможно всего, будет что-то другое.

Следующий шаг - компилятор. Мы будем использовать компилятор gcc. Практически все дистрибутивы Linux поставляются вместе с компилятором языка C. На начальном этапе обучения версия компилятора не имеет значения. Если у вас есть компилятор gcc, то и линковщик (о том, что это такое, будет рассказано позже) тоже есть (вероятность 99%).

Многие программисты используют различные оболочки для программирования (KDevelop, Glade и прочие). Мы откажемся от этого подхода по двум причинам: 1) это претит идеологии Unix; 2) за всеми этими средами разработки стоят те же самые утилиты, которые мы будем использовать непосредственно. Нам на начальном этапе потребуется лишь утилита make.

И, наконец, немаловажная деталь. Текстовый редактор. Здесь на вкус и цвет товарища нет. В мире Unix-систем есть два лагеря: любители Vi и любители Emacs. В настоящее время написано много клонов редактора Vi (Vim, Elvis, GVim) и много расширений редактора Emacs. И для Vi и для Emacs существуют как консольные так и графические версии. Нет смысла обсуждать преимущества и недостатки этих двух "гигантов". Вам придется самостоятельно выбрать наиболее удобный для вас текстовый редактор. Возможно вам не понравится ни один из них и вы будете использовать что-нибудь наподобие gedit, mcedit или nano. Между прочим, редактор nano открывает свой третий лагерь поклонников. Очень важно попробовать разные редакторы "в полную силу", а не просто "взглянуть". Только опытный пользователь сможет за видимой неуклюжестью ощутить мощь и удобство. Это одна из важнейших граней идеологии Unix.

подавляющую часть работы мы будем проводить в командной строке. "Командная строка" - общий термин, подразумевающий собой сочетание терминальной программы и командной оболочки. Терминальная программа (или программа эмуляции терминала), это то, что вы видите на экране, когда работаете с командной строкой. Терминальная программа позволяет пользователю вводить команды и видеть результаты их выполнения. Под терминальной программой запускается командная оболочка (командный интерпретатор или просто shell). Командная оболочка через интерфейс терминальной программы общается с пользователем и выполняет всю основную работу. Bash, Csh, Ksh - это все командные оболочки или, как их еще называют, командные интерпретаторы. В Linux я рекомендую использовать оболочку Bash (Bourne Again SHell). Однако не произойдет ничего страшного, если вы будете использовать другую командную оболочку, например C-Shell (csh), у которой достаточно своих поклонников. Как вы позже узнаете, любая командная оболочка - это не просто исполнитель команд, но и полноценный интерпретатор собственного языка программирования. Чтобы узнать, какая оболочка запущена под терминальной программой, просто наберите команду ps. В одной из строк (скорее всего - в первой) будет написана ваша

командная оболочка:

```
$ ps
PID  TTY  TIME  CMD
5003 pts/0 00:00:00 bash
6025 pts/0 00:00:00 ps
$
```

В моем случае это bash.

## 1.4. Лицензирование

Данное методическое пособие распространяется на условиях GNU Free Documentation License. В соответствии с условиями этой лицензии вы можете свободно скачивать и распространять его без каких-либо разрешений, но не нарушая и не изменяя условий Лицензии. Если вы хотите поместить копию этого методического пособия на своем Интернет-ресурсе или просто распечатать - делайте это свободно!