Разработка программ на C

А. Г. Фенстер, `fenster@fenster.name`

24 апреля 2010 г.

Разработка программ напоминает строительство зданий тем, что необходимо придерживаться некоторых правил при написании программы, в противном случае она почти в прямом смысле слова развалится — дальнейшая поддержка её станет невозможной. Сегодня мы поговорим про правила написания программ в структурном стиле вообще и на языке C в частности.

Дополнительное чтение: Ален Голуб «Правила программирования на Си и Си++» — возможно, спорная в некоторых моментах, но заслуживающая прочтения книга.

1. Принципы структурного программирования: разбиение программы на подпрограммы (функции), три способа комбинирования действий (последовательное выполнение, условная инструкция, цикл), отсутствие `goto` (вообще говоря, спорный момент, но в студенческом коде `goto` быть не должно).

2. Ещё раз о раздельной компиляции в C. Модульность. Как устроена утилита `make`.

3. Замечания о программировании на C:

   (a) глобальные переменные: вызывают неявные зависимости функций друг от друга; вызывают проблемы при многопоточном программировании (необходима синхронизация). Следовательно, все контейнеры (списки, стеки) реализуем так, чтобы не было необходимости хранить что-либо в глобальной переменной:

   ```
   void push(struct item **head, T data);
   ```

(b) дисциплина работы с памятью: даже в небольших учебных программах необходимо освобождать выделенную память. Помимо того, что это является правилом хорошего тона, освобождение всей памяти может выявить ошибку в программе (если у вас Segmentation fault при освобождении списка — вероятно, эта ошибка может «выстрелить» и при работе с ним).

(c) именование: как минимум, отделяйте макросы от функций (макросы — заглавными буквами). Не определяйте макросы для совершенно простых действий.

(d) минимизация уровня вложенности (пример: вставка элемента в конец списка). Использование `break`.

(e) возвращаемые значения важны, их нужно проверять у стандартных функций (`malloc`, `fopen`) и свои функции писать желательно с возможностью контроля ошибок.

(f) помните про знак типа (особенно касается `char`).

(g) дублирование подсчётов, особенно `strlen` и `strcmp`.

4. Инструменты разработчика:

(a) Минимальный набор: редактор, компилятор.

(b) Интегрированные среды.

(c) Отладка программы. Отладочная печать.

(d) Отладчики. Понятие отладочной информации. Функции отладчика: пошаговый прогон, печать значений переменных, печать стека вызовов.

(e) Динамические верификаторы. `valgrind`, `crtdbg.h`.

(f) Статические верификаторы (lint-style).