

ВВЕДЕНИЕ

Caveat emptor²: стоимость сопровождения программного обеспечения пропорциональна квадрату творческих способностей программиста.

(Первый закон творческого программирования,
Роберт Д. Блис (Robert D. Bliss), 1992)

Перед вами сборник программных приемов, которые я собирал много лет. Большинство из них работают только на компьютерах, на которых целые числа представлены в дополнительном до 2 коде. Хотя в данной книге речь идет о 32-разрядных машинах с соответствующей длиной регистра, большую часть представленных здесь алгоритмов легко перенести на машины с регистрами других размеров.

В этой книге не рассматриваются сложные вопросы наподобие методов сортировки или оптимизации компилируемого кода. Основное внимание здесь уделяется приемам работы с отдельными машинными словами или командами, например подсчету количества единичных битов в заданном слове. В подобных приемах часто используется смесь арифметических и логических команд.

Предполагается, что прерывания, связанные с переполнением целых чисел, замаскированы и произойти не могут. Программы на C, Fortran и даже Java работают именно в таком окружении, но программистам на Pascal и ADA следует быть осторожными!

Представление материала в книге — неформальное. Доказательства приводятся только в том случае, если алгоритм неочевиден, а иногда не приводятся вообще. Все методы используют компьютерную арифметику, функции типа “пол”, комбинации арифметических и логических операций и тому подобные средства, а доказательства теорем в этой предметной области часто сложны и громоздки.

Чтобы свести к минимуму количество типографских ошибок и опечаток, многие алгоритмы реализованы на языке программирования высокого уровня, в качестве которого используется C. Это обусловлено его распространенностью и тем, что он позволяет непосредственно комбинировать операции с целыми числами и битовыми строками; кроме того, компиляторы языка C генерируют объектный код высокого качества.

Ряд алгоритмов написан на машинном языке. В книге применяется трехадресный формат команд, главным образом для повышения удобочитаемости. Использован язык ассемблера для некой абстрактной машины, которая является представителем современных RISC-компьютеров.

² Caveat emptor (лат.) — да будет осмотрителен покупатель.

Отсутствие ветвлений в программе всячески приветствуется. Это связано с тем, что на многих машинах наличие ветвлений замедляет выборку команд и блокирует их параллельное выполнение. Кроме того, наличие ветвлений может препятствовать выполнению оптимизаций компилятором — таких как расписание выполнения инструкций, распределение регистров и других. Оптимизирующий компилятор более эффективно работает с несколькими большими блоками кода, чем с множеством небольших.

Крайне желательно использовать малые величины при непосредственном задании операнда, сравнение с нулем (а не с некоторыми другими числами) и параллелизм на уровне команд. Хотя программу часто можно значительно сократить за счет использования поиска в таблице (расположенной в памяти), этот метод не слишком распространен. Дело в том, что по сравнению с выполнением арифметических команд загрузка данных из памяти занимает намного больше времени, а методы поиска в таблице зачастую не представляют большого интереса (хотя и весьма практичны).

Напоследок мне хотелось бы напомнить исходное значение слова “хакер”³. Хакер — это страстный любитель компьютеров; он создает что-то новое, переделывает или совершенствует то, что уже есть. Хакер очень хорошо разбирается в том, что делает, хотя часто не является профессиональным программистом или разработчиком. Обычно он пишет программы не ради выгоды, а ради собственного удовольствия. Такая программа может оказаться полезной, а может остаться всего лишь игрой интеллекта. Например, хакер может написать программу, которая во время выполнения выводит точную копию себя самой⁴. Таких людей называют хакерами, и эта книга написана именно для них, а не для тех, кто хочет получить совет о том, как взломать что-либо в компьютере.

Благодарности

Прежде всего я хочу поблагодарить Брюса Шрайвера (Bruce Shriver) и Денниса Эллисона (Dennis Allison), оказавших мне помощь в опубликовании книги. Я признателен коллегам из IBM, многие из которых упомянуты в списке литературы. Особой благодарности достоин Мартин Хопкинс (Martin E. Hopkins) из IBM, которого по праву можно назвать “Мистер Компьютер”. Этот человек неудержим в своем стремлении подсчитать в программе каждый такт, и многие его идеи нашли отражение в этой книге. Благодарю также обозревателей из Addison-Wesley, которые значительно улучшили мою книгу. Со многими из них я не знаком, но не могу не упомянуть выдающийся 50-страничный обзор одного из них, Гая Л. Стила-мл. (Guy L. Steele, Jr.), в котором, в частности, затронуты вопросы перемешивания битов, обоб-

³ В последнее время хакерами часто называют тех, кто получает несанкционированный доступ к банковским системам, взламывает веб-узлы и ведет прочую разрушительную деятельность либо ради получения денег, либо для демонстрации всем своей “крутости”. К настоящим хакерам таковые не имеют никакого отношения. — *Примеч. пер.*

⁴ Вот пример такой программы на языке программирования C.

```
main() {char*p="main() {char*p=%s%c; (void)printf(p,34,p,34,10);} %c"; (void)printf(p,34,p,34,10);}
```

щенного упорядочения и многие другие. Ряд предложенных им алгоритмов лучше тех, которые я использовал ранее. Помогла мне и его пунктуальность. Например, я ошибочно написал, что шестнадцатеричное число 0хAAAAAAAA можно разложить на множители 2·3·17·257·65537; Гай указал, что 3 необходимо заменить на 5. Он не упустил ни одной из подобных мелочей и намного улучшил стиль изложения материала. Кроме того, весь материал, связанный с методом “параллельного префикса”, появился в книге исключительно благодаря Гаю.

Дополнительный материал по данной книге можно найти по адресу www.HackersDelight.org

Г.С. Уоррен-мл. (H.S. Warren, Jr.)
Йорктаун, Нью-Йорк
Июнь 2012