

# Программирование плюс ... что?

Доктор физ.-мат.наук, профессор А.Н.Терехов  
"Read.Me", N7-8, СПб, 1995

Редакция журнала ReadMe оказала мне честь, дав возможность представить на страницах журнала возглавляемый мною коллектив - лабораторию системного программирования НИИММ С-Петербургского университета и государственное предприятие "Терком". Основной состав включает в себя около 60 человек, еще около 100 человек практически постоянно работают во временных творческих коллективах. Мы занимаемся программным обеспечением правительственных систем связи, разработали свою технологию программирования для встроенных систем реального времени, новую ЭВМ "Самсон", ориентированную на современные языки программирования, модем, обеспечивающий передачу 16 КБит/сек по отечественным телефонным каналам, которые по печально известному приказу N50 Минсвязи хуже международных стандартов, но на законных основаниях. Имеем несколько заказов от итальянских и американских фирм по программному обеспечению и электронике. Надеюсь, что еще будет возможность описать эти работы подробнее.

В первой же статье хотелось бы рассказать об одной, довольно амбициозной идее, которая уже воплощается в нескольких проектах.

Чем отличается профессиональное программирование от любительского? Сегодня любой молодой специалист, следящий за доступной периодикой, в ответ на такой вопрос начнет скучным голосом говорить про важность спецификаций и качественной документации, использование мощных инструментальных средств и наличие регулярного сопровождения и т.д. Но, к счастью для профессионалов, разница проявляется не только в этих, безусловно необходимых вещах. И 20-30 лет назад было известно, что два программиста, сидящих за соседними столами и получающих примерно одинаковую зарплату, могут писать программы, отличающиеся по скорости счета, скажем, в 10 раз, но на практике "живые" эксперименты встречались редко. Чаще действительно классные программисты группировались в нескольких академических или университетских центрах и общались только между собой, снисходительно наблюдая за робкими попытками непрофессионалов воспользоваться ЭВМ.

Не была исключением и лаборатория системного программирования НИИММ Ленинградского университета, где я начал работать еще студентом. Общение с выдающимися учеными Г.С.Цейтиным и С.С.Лавровым, с А.П.Ершовым, а через них - с известными зарубежными учеными (например, в 1976 году мало кто мог похвастаться личным общением с Э.Дейкстрой, Т.Хоаром, Д.Шварцом - отцом первого языка сверхвысокого уровня SETL) создавало (возможно, и незаслуженно) ореол исключительности. Соответственно выбирались и направления исследований: трансляторы с самых сложных и лучших (с нашей точки зрения) языков Алгол 68 и Ада, изучение и реализация первых языков

искусственного интеллекта, создание новых языков. Общение с программистами "от сохи" могло быть только случайным.

Один такой случай произошел на колхозном поле, где мы, при всей нашей элитности, вынуждены были бывать. К сожалению, я не слишком силен в разговорах о чем-либо, кроме моей науки, поэтому и в поле разливался соловьем про искусство программирования. В ответ я услышал множество грустных историй про то, что задачи по механике считаются долго, что результаты двухчасового счета можно получить только на следующий день, т.к. своей БЭСМ-6 у университета нет, а если ошибся в одной перфокарте, то... Вот если бы можно было свести счет к получасу, то задача попадала бы в другую очередь и можно было бы сразу получать результат. Я, естественно, поспорил на коньяк, что, не видя заранее задачи и программы, берусь ускорить программу в 4 раза. Мой расчет был прост - в любом случае распить коньячок не так уж плохо. Но я ни разу не проиграл.

Слух об оптимизации на спор дошел до директора НИИММ Г.П.Самосюка, который не одобрил такого начинания. Но, чтобы сохранить для института экономию машинного времени, он устроил цикл моих лекций "Оптимизация программы ее автором", куда обязал ходить всех руководителей лабораторий и ведущих специалистов. Что самое удивительное для меня, в то время совсем молодого человека, они ходили и задавали массу вопросов, в свою очередь, я интересовался их задачами, методами, трудностями, в общем, набирал опыт.

Я стал коллекционировать задачи, для решения которых советскими учеными были найдены алгоритмы, но из-за нехватки вычислительных ресурсов этими решениями трудно было воспользоваться. Вначале это было своего рода хобби, сегодня многие из них приносят коммерческий успех.

Разумеется, первооснова закладывается математиком - автором алгоритма. Никакими программистскими ухищрениями не добиться превращения экспоненциально сложной задачи в задачу с полиномиальной сложностью. Как-то раз меня попросили помочь в задаче поиска наилучшего приближения путем триангуляции минимальной площади. Грустно было сознавать, что, даже улучшив программу в 1000 раз, уменьшаешь количество действий с 1 со 150 нулями до 1 со 147 нулями.

Но бывает, что совершенный алгоритм "гробится" любительским программированием. 20 лет у меня из головы не выходит фраза М.Донского, сказанная им в Репино на Международном конгрессе по искусственному интеллекту: "В программе "Пионер" М.М.Ботвинника заложены глубокие идеи, но кто же представляет в программе шахматную доску массивом 8x8?". Кстати, там же я познакомился с очень симпатичным и совсем не старым автором ставшей уже классической идеи подравненных деревьев Г.М.Адельсоном-Вельским. Теперь мне становится не по себе, когда в нашей литературе пишут "ADV-деревья" вместо "АДВ-деревья".

Мы пытаемся развивать следующую стратегию. В системах, которые нам заказывают, мы выделяем задачи, имеющие самостоятельное научное и коммерческое значение. Если задача близка по профилю к системному программированию, ищем эффективное решение сами, если же далека - стараемся найти соответствующих специалистов, благо в университете это практически всегда возможно. Пополняется и коллекция интересных задач, решенных не нами, но пока не имеющих достойной реализации. В любом случае схема такова. Профессионалами пишется программа на алгоритмическом языке высокого уровня (АЯВУ), затем выполняется серия экспериментов с замерами времени счета отдельных фрагментов. Как известно, в любой программе есть 10% текста, где спрятаны 90% времени счета. Теперь можно забыть про всю программу и сосредоточиться на найденных "узких местах" (американцы их называют time-consuming, т.е. время-пожирающими).

Иногда удается оптимизировать фрагмент, не выходя за рамки АЯВУ. Например,

```
for j to n do ... a[i,j] ... od;
```

вырезку по  $i$  хорошо бы вынести из цикла по  $j$ . На Алголе 68 эту программу можно преобразовать следующим образом:

```
ref [] real ai = a[i,];  
for j to n do ... ai[j] ... od;
```

Никогда мне не забыть удивления, когда одна из слушательниц моих лекций через год вполне успешной работы призналась в своей уверенности, что весь фокус в имени идентификатора (первая буква от массива, вторая - от индекса).

Стандартным приемом оптимизации является замена фрагмента на АЯВУ подпрограммой на ассемблере, но существенного эффекта можно добиться только, если удастся использовать какие-то специализированные команды ЭВМ, которые не порождаются транслятором. В ЭВМ, допускающих динамическое микропрограммирование, можно расширять систему команд специализированными функциями, получая выигрыш в 10-30 раз.

На практике этот мощный способ почти не применяется, например, все ЕС ЭВМ Ряд 2 имеют загружаемую микропамять, но что-то я не слышал об успехах в этой области, хотя попытки в Москве, Минске и Вильнюсе предпринимались. На мой взгляд, в основном это обусловлено низким уровнем инструментальных средств - микроассемблеры типа языков регистровых передач. Когда 8 лет назад мы начали проектировать "Самсон", вопрос микропрограммирования встал очень остро. Перепробовав несколько советских и зарубежных инструментальных средств, руководитель этого направления в нашем коллективе Н.Ф.Фоминых решил разработать собственный способ записи микропрограмм на Алголе 68. В принципе, это тоже микроассемблер, поскольку каждая микрокоманда выписывается отдельно, но использование всех возможностей современного АЯВУ (вызовы, циклы,

определение новых типов и операций над ними, а самое главное - жесткий видовой контроль) качественно изменило ситуацию, многие технические детали оказались спрятанными от микропрограммиста, а уровень статического и динамического контроля обеспечил нахождение большого количества ошибок, не обнаруживаемых типичными микроассемблерами. В результате, на Алголе 68 были реализованы 5-6 различных архитектур, причем в конечном итоге микропрограммирование упростилось до уровня студенческой работы.

Микропрограммная замена 5 строк в программе шифрации с открытым ключом дает выигрыш в 18 раз, микропрограммная реализация разыгрывания точки, равномерно распределенной на N-мерной сфере, дает выигрыш в 10 раз в задаче решения уравнений в частных производных методом Монте-Карло.

В тех случаях, когда и микропрограммирование не спасает, можно разработать плату, аппаратно реализующую наиболее критичные по времени функции, при наличии опыта и соответствующего оснащения это совсем не трудно.

Пожалуй, наиболее полно все перечисленные возможности были использованы при разработке модема, передающего 16 КБит/сек по телефонным каналам. При такой высокой скорости передачи начинает сказываться межсимвольная интерференция в канале (каждый импульс некоторое время остается в канале в виде затухающего колебания и накладывается на последующие импульсы), кроме того, характеристики канала постепенно изменяются. Таким образом, задача реализации модема в основном относится к теоретической кибернетике. Для нашего модема на основе некоторого алгоритма, предложенного связистами, группа из 5 кандидатов наук с кафедры теоретической кибернетики, разработала корректные и эффективные алгоритмы. Гладкую, легко вычислимую П-образную функцию для фильтрации нашел специалист в области конструктивной теории функций. Эти алгоритмы были запрограммированы на Алголе 68 и отлажены на математических моделях. Затем ключевые места были замикропрограммированы, а быстрое преобразование Фурье было реализовано в виде отдельной платы. В настоящее время 8 макетов модема, реализованных совместно с НПО "Импульс", прошли испытания и показали хорошие результаты на реальных каналах.

В заключение хочу еще раз подчеркнуть основную мысль статьи. В наше смутное время, когда выходцы из России заполнили все программистские рынки труда в Европе и особенно в США, очень трудно рассчитывать на успех с чисто программистской работой. Нужно иметь какую-то особенность, изюминку, в качестве которой вполне может быть оригинальная математическая идея представителей С-Петербургской математической школы, глубокие традиции которой трудно поколебать даже такими потрясениями.

Мы занимаемся теоретико-числовыми задачами (например, шифрацией, "электронной подписью" и т.п.), символьными выкладками, распределенными отказоустойчивыми базами данных, вероятностным моделированием, распознаванием образов, пытаемся развернуть совместные работы с

лабораторией аэродинамики. Приглашаем к сотрудничеству всех, у кого есть оригинальные идеи, нуждающиеся в профессиональной программной и аппаратной поддержке.