

Повесть блудного сына проекта БЕТА

М. И. Шварцман

ValueSearch Capital Management, LLC

“Истинный успех академического проекта состоит в оставленном им идейном наследии и в обученных при его разработке студентах. Если то и другое хорошо, то проект удался.”

Проф. Дж. Гуттаг¹, МТИ

“Первый миллион всего труднее заработать.”

Американская народная мудрость

“Любой идиот может выстоять перед кризисом.

Что изнуряет — это повседневные проблемы.”²

Антон Чехов, размышляющий в конце XIX столетия о бедственном положении людей вообще, а может быть, в частности, и руководителей некоторых сложных технических проектов XX века.

Введение

К 2004 году, когда я пишу эту статью, я посвятил равное число лет двум различным областям деятельности: семнадцать лет (1970–1987) я занимался информатикой и разработкой программного обеспечения и семнадцать (1987–2004) — инвестированием и финансовыми операциями. Здесь я хочу рассказать о том, что происходило давным-давно, в первые семнадцать лет.

© Michael I. Schwartzman, 2004, 2005. Статья написана специально для настоящего сборника. Шварцман М.И. Повесть блудного сына проекта Бета // Андрей Петрович Ершов - ученый и человек / ред.-сост. М. А. Бульонков [и др.]; отв. ред. А. Г. Марчук; Рос. акад. наук, Сиб. отд-ние, Ин-т сист. информатики им. А. П. Ершова. - Новосибирск: Издательство СО РАН, 2006. - 504 с. - (Наука Сибири в лицах). - С.185-207.

¹ Дж. Гуттаг — профессор Массачусетского технологического института (МТИ), в настоящее время возглавляет Группу сетей и мобильных систем в Лаборатории вычислительных наук и искусственного интеллекта МТИ.

² Обратный перевод с английского. Русский оригинал нам локализовать не удалось. — Прим. переводчика

Три года я принимал участие в проекте БЕТА, задуманном и руководимым Андреем Петровичем Ершовым — с момента его возникновения в 1970 году и до моей эмиграции в США в конце 1973 года. Я отвечал за разработку Внутреннего Языка системы; после моего отъезда эту работу продолжили другие участники проекта. Вначале я буду касаться лишь тех аспектов научного наследия А.П.Ершова, которые относятся к проекту БЕТА, а затем обращусь к другим, не связанным с наукой, моментам.

Если говорить кратко, целью системы БЕТА было создание многоязыкового компилятора, переводящего программы с каждого из допустимых в системе исходных языков на Внутренний Язык, а затем с него на любой допустимый в системе машинный язык. Компилятор должен был совершать интенсивную оптимизацию на уровне Внутреннего Языка. В начале разработки системы были приняты следующие исходные языки:

- Алгол 68
- ПЛ/I
- Симула-67
- Паскаль

Проект БЕТА, который прекратил существование в 1982 году, спустя двенадцать лет после своего появления, не создал, как это вроде бы общепризнанно, никаких работающих систем. В то же время проект сделал более ясным как структуру языков, с которыми он работал, так и различные аспекты проблемы построения компиляторов.

Часто утверждают, что судьба системы БЕТА постигла и все другие попытки создать практический многоязыковый компилятор.

Но хотя проект БЕТА так и не привел к созданию работающей системы, у него имеется некий интеллектуальный отпрыск, осуществивший всё, что в проекте БЕТА намечалось и, быть может, даже более того.

Этот проект называется «LPI Multi-Language Family of compilers (LPI-MLF)» и разработан он в компании «Language Processors, Inc.» («LPI»), которую я организовал в 1980 году в Массачусетсе. Вскоре в ней начал работать Джон Анкорн³, и до 1986 года он руководил всеми перспективными исследовательскими разработками, включая и LPI-MLF.

Хотя проект был широко известен и рассматривался в научно-исследовательских отделах всех американских и многих зарубежных фирм, производящих компьютеры, он никогда не привлекал внимания

³ Дж. Анкорн в настоящее время ведущий научный сотрудник в Исследовательских лабораториях Хьюлетт—Паккарда, специалист по мобильным вычислительным системам и сетям.

теоретиков информатики — ситуация, совершенно обратная проекту БЕТА. Но, разумеется, во время разработки и активной продажи системы компания не поощряла, скорее, даже запрещала своим сотрудникам публикацию любых материалов, которые могли бы дать конкурентам хоть какую-то информацию, полезную в их разработках.

LPI-MLF оказалась первой практически реализованной и успешной многоязыковой компилирующей системой. Порождаемые ею компиляторы были весьма конкурентоспособны, если сравнивать их с компиляторами, написанными для конкретных языков. Система была практичной, удобной в пользовании, и её действительно удавалось перенастраивать и использовать на вычислительных машинах с разной архитектурой. Можно сказать, что LPI-MLF — это первая многоязыковая компилирующая система с высокими производственными характеристиками.

Когда система LPI-MLF потеряла актуальность, главные участники ее разработки обрели новые интересы и занялись другими делами, порою весьма амбициозными. Джон Анкорн переехал в Японию, где создал компанию по производству вычислительной техники; я основал инвестиционную фирму в штате Массачусетс. В результате все, что было опубликовано о системе LPI-MLF — это рекламные материалы и внешние технические характеристики системы, необходимые в процессе ее продажи.⁴

Результаты, полученные в компании LPI

К 1984 году, когда все компоненты LPI-MLF были завершены и началось коммерческое использование системы, она включала следующие компоненты:

- Внешний интерфейс для Кобола, RPG-II, Фортрана, ПЛ/I, Бейсика, Паскаля и Си.
- Оптимизатор на уровне Промежуточного языка с несколькими уровнями оптимизации, выбираемыми пользователем.
- Большой и всё расширяющийся набор генераторов объектного кода для ЭВМ различной архитектуры, в том числе для Motorola 68000, Intel x86, IBM 8100, ATT 3B2, ATT 3B20.
- Отладчик на уровне исходного текста.
- Общая библиотека периода счёта.
- Набор средств для порождения генераторов объектного кода по таблицам, задающим новые компьютерные архитектуры. В

⁴ Одну из статей, посвященных LPI-MLF, и собственные рекламные материалы этой фирмы можно найти в Электронном архиве академика А.П.Ершова на сайте <http://ershov.iis.nsk.su/archive/caindex.asp?lang=2&gid=1464>.

самих генераторах использовались средства для редукции деревьев выражений.

- Достаточно изощренная система автоматического тестирования компиляторов.
- Инструментарий для создания внешних интерфейсов для новых языков на базе наиболее продвинутых исследований по синтаксическому анализу.

Система LPI-MLF использовалась в больших и малых компаниях, производящих вычислительную технику, и в «start-up» фирмах и подразделениях, только выходящих на рынок (хороший пример — AT&T Information System). Заключение контрактов на поставку системы фирмам требовало немало усилий, а конкуренция с собственными научно-исследовательскими командами из этих фирм была весьма интенсивной.

В 1987 году, когда система достигла пика своих производственных возможностей, прототипный генератор объектного кода для компьютера новой архитектуры создавался за неделю, а для выпуска работающего генератора с высоким качеством объектного кода требовалось около четырёх месяцев. Технология сборки компонентов достигла такого совершенства, что готовую версию компилятора можно было собрать и полностью протестировать за несколько часов, что обеспечивало очень быстрый цикл выпуска компиляторов. В те времена в промышленности на выпуск очередного компилятора уходило обычно от шести до двенадцати месяцев. А создание самостоятельного компилятора с конкретного языка для новой машины требовало нескольких лет, причём далеко не всегда проект оказывался успешным. Система же LPI-MLF позволяла получить новый компилятор за несколько месяцев, а на добавление компиляторов с других языков уходило несколько недель.

Лишь немногие фирмы покупали систему со всеми семью языками. Что нравилось пользователям LPI, так это возможность быстрого получения одного–двух компиляторов (чаще всего с Си или Фортрана, иногда с Кобола) с тем, чтобы использовать один из этих компиляторов для решения основных классов задач на машине новой архитектуры и добавлять новые компиляторы в случае необходимости.

Таким образом, технологический прорыв фирмы LPI был оценен пользователями не столько за то, что можно было сразу же приобрести многоязыковую компилирующую систему для их компьютеров, а, скорее, за возможность добавить новый компилятор быстрее и дешевле, чем делать самим или покупать у конкурентов LPI.

Вся система была довольно большой по стандартам середины 80-х годов. Она занимала 16 мегабайт, в то время как объем жёстких дисков составлял тогда от 30 до 50 мегабайт, и это было одной из причин,

почему многие пользователи не приобретали систему в целом. Большинство кодов исходного текста было написано на ПЛ/I, так что пользователи исходного текста нуждались также в ПЛ/I компиляторе нашей же фирмы.

Системе приходилось конкурировать с традиционными компиляторами, разработанными для одного языка и оптимизированными в соответствии со своими специфическими функциями. Однако никакие значительные продажи не имели бы места, уступай компиляторы LPI-MLF своим конкурентам. Но они и не уступали. Наши компиляторы с Фортрана, Си, Кобола при эталонном тестировании постоянно показывали лучшие результаты как по времени компиляции, так и по времени счёта. В действительности, дабы конкурировать с существующими компиляторами для конкретной архитектуры ЭВМ, львиная доля цикла создания LPI-MLF приходилась на эталонное тестирование и настройку генератора объектного кода на данную архитектуру.

Разборка нудных жалоб заказчиков, т.е. собственных программистов в фирмах-производителях компьютеров, часто склонных доказывать неполноценность полученного программного обеспечения, также отнимала много времени, но это, впрочем, имело бы место независимо от используемой технологии создания компиляторов. В 80-е годы фирмы, производящие компьютеры, совсем иначе относились к независимым разработчикам программ, чем в 90-е годы и ныне.

Возможно, эти перемены связаны с тем, что осталось мало производителей ЭВМ, из которых могли бы сейчас выбирать системные программисты. Тот факт, что именно производители, наиболее закоснелые в стремлении диктовать цены на разработку программного обеспечения, испытали наибольшие трудности или просто перестали существовать, как раз и мог стимулировать появление в 90-х и 2000-х годах нового подхода к этому бизнесу. Но это сейчас, а LPI продала свою систему LPI-MLF на пике битв «делать самим или покупать», которые шли внутри научно-исследовательских отделов фирм — производителей компьютеров.

Целые тома заняло бы описание всех трудностей, неудач, побед и огромной работы, затраченной на разработку, документирование и отладку системы LPI-MLF. Разочарование порой преобладало, и требовались огромные усилия для того, чтобы компания продолжала свой курс.

LPI-MLF была успешной системой, лицензированной на дюжине ведущих фирм — производителей компьютеров в США и Японии и на множестве фирм, только выходящих на рынок.

Компиляторы LPI-MLF не только успешно конкурировали с традиционными компиляторами, но и превосходили многие по таким

общепринятым критериям, как время компиляции и счёта при эталонном тестировании. Время поставки на рынок и стоимость компиляторов были в LPI сокращены многократно.

Система LPI-MLF достигла практически всех целей, поставленных перед системой БЕТА пятнадцатью годами ранее. Сделано это было для весьма широкого набора языков силами небольшой группы разработчиков — на самом пике процесса, занявшего от трёх до четырёх лет, научно-исследовательское подразделение компании LPI состояло из 35-40 человек.

Корни LPI-MLF и причины её успеха

LPI-MLF была создана после многолетних попыток со стороны разработчиков компиляторов (включая команду БЕТА) построить многоязыковую компилирующую систему. Конечно, сама идея многоязыковой компилирующей системы не родилась в проекте БЕТА. Ко времени его начала она уже много лет обсуждалась.

Интеллектуальное влияние проекта БЕТА на LPI-MLF несомненно; я был основателем, президентом и председателем правления фирмы LPI первые семь лет. Сама мысль о будущей системе LPI-MLF пришла ко мне, в частности, на основе моего опыта в проекте БЕТА. Проект БЕТА, последующий опыт создания общих компонентов для семейства компиляторов в корпорации DEC (Digital Equipment Corporation), руководство Отделом разработки компиляторов в фирме Prime Computer — все это, как и многое другое, определило успех LPI.

Следует сразу же отметить, что БЕТА — лишь одна из многих опор в основании принятого в LPI подхода. Джон Анкорн, чье научное руководство определило наши достижения, принёс в проект идеи и методы, не имеющие никакого отношения к БЕТА. И мой собственный опыт к моменту создания LPI вовсе не ограничивался работой в проекте БЕТА.

В научно-исследовательском отделе LPI работало с десятков специалистов мирового класса в области создания компиляторов, часть из них имела докторскую степень по информатике; все они имели собственное представление о том, как достичь успеха, и собственное понимание того, в чём состоит успех. Дж. Анкорн осуществлял повседневное руководство группой разработчиков. В мои обязанности входило общее руководство компанией, в том числе определение целей и темпов работы, формулирование требований к программному продукту, а также вопросы ценообразования, маркетинга и поиск источников финансирования.

БЕТА, как и другие подобные проекты, как многие проекты такой же чрезмерной сложности, погибла от тысяч комариных укусов, от

тысяч мелких неудач. Они провалились потому, что многие сравнительно малые проблемы, накапливаясь, достигают критической точки, и проект уже нельзя спасти. По сути, понимание того, что же помешало в прошлом успешно реализовать многоязыковый компилятор само по себе могло бы стать бесценным опытом. Любую малую проблему, относится ли она к некоей конструкции Промежуточного Языка, ограничениям на память или чему иному, специалист мог решить легко и сразу, но как справиться с нагромождением таких малых и взаимосвязанных проблем — вот в чем задача.

В фирме LPI необходимо было поддерживать баланс между прагматизмом с одной стороны и желанием ученых в LPI, обладающих исключительно хорошим знанием академической литературы, приложить все это знание — с другой. Нам блестяще удалось найти этот баланс и обуздать проблему решения тысячи малых проблем до того, как они приведут к катастрофическому провалу.

Можно сказать, что пробную попытку ввести общие компоненты в компилирующую систему я сделал, работая в DEC. Там, в условиях жесточайшего графика разработки нового компьютера (VAX), новой операционной системы (VMS) и серии новых компиляторов (первый — для Фортрана), была предпринята попытка ввести Общую библиотеку времени счёта, которая служила бы как для этого, так и для последующих компиляторов, и создать библиотеку подпрограмм, которую могли бы использовать другие подсистемы VAX. Находясь под давлением столь напряжённого графика, мы смогли добиться замечательного успеха: путь от спецификации на бумаге новой архитектуры до поставки первого компьютера VAX был пройден за три года.

Битвы были памятными, если не сказать больше. В рабочей записке Дэйва Катлера⁵ (см. ниже), который руководил сравнительно небольшой группой из двадцати пяти разработчиков будущей операционной системой VMS, адресованной Ричарду Гроуву⁶, работавшему под моим руководством и отвечавшему за разработку компилятора для Фортрана, сформулированы проблемы и вопросы, обсуждавшиеся при введении таких новых концепций, как Общая библиотека времени счёта и связанные с ней проблемы. Их введение затрагивало компоновщик (подсистему операционной системы),

⁵ Дэйв Катлер в настоящее время возглавляет большую группу системных программистов, разрабатывающую несколько версий Windows в корпорации Майкрософт.

⁶ Ричард Гроув в настоящее время — ведущий научный сотрудник и Fellow (самое высокое техническое звание в корпорации Интел), специалист в области разработки компиляторов.

разрабатываемый Тревом Портером⁷, который работал у Дэйва Катлера. В то время, когда Дэйв писал эту записку, разработка компоновщика уже была на грани провала из-за противоречащих требований и отстала от графика. Введение Общей библиотеки времени счёта не спасало ситуацию.

СЛУЖЕБНАЯ ЗАПИСКА⁸

Ричарду Гроуву
От Дэйва Катлера

Копии: Роджеру Гаурду⁹, Тревору Портеру, Михаилу Шварцману
16 ноября 1976

Тема: ПРОВЕРКА АРГУМЕНТОВ В МОМЕНТ СВЯЗИ

На: Ваша спецификация от 16-11-1976

Кажется, мне следует сначала поздравить Вас с хорошо написанной спецификацией, а затем перейти к рассмотрению недочётов. Однако я не могу так поступить ввиду моего крайнего разочарования. Вы заменили реализуемое, хотя, быть может, и несовершенное, предложение на предложение, которое **может** оказаться совершенным, но нереализуемо ввиду наложенных на проект ограничений.

Мы затратили уже слишком много времени на обсуждение этого вопроса, и я думаю, вы тоже. Можно спросить себя, нельзя ли было потратить ваше время лучшим образом. Почему вы не изобрели ещё какой-нибудь хитроумной оптимизации, которая улучшила бы эффективность Фортрана, или не нашли способа выдать Ваш компилятор раньше? Эти вещи стоят реальных денег. Вы довели предложение до такой точки, что оно стало абсурдным. Реализация этого предложения будет стоить нам по меньшей мере **одного человека-месяца**, и в результате мы получим убогодичный объектный язык. Ни у нас, ни у вас нет свободного человека-месяца. Ваше предложение превращает модель объектного языка из простого потока байтов в какой-то контекстно-чувствительный набор прощупываемых битов, который, кстати, должен интерпретироваться каждый раз, когда встречается такое поле независимо от того, обрабатывается оно или игнорируется, поскольку мы не можем узнать его длины, не взглянув на него. Думаю, Вы никогда бы не изобрели подобный язык программирования?

⁷ Тревор Портер — отвечал за разработку компоновщика и нескольких других подсистем на начальных этапах создания операционной системы VAX/VMS в корпорации DEC.

⁸ Английский оригинал см. в Электронном архиве А.П.Ершова на сайте <http://ershov.iis.nsk.su/archive/eaindex.asp?lang=2&did=27212>

⁹ Роджер Гаурд отвечал за разработку первой версии операционной системы VAX/VMS в корпорации DEC.

В том, что проступает, есть нечто знаковое, и надеюсь, что Вы это понимаете. Вы попытались удовлетворить всех **независимо** от стоимости, которая определяется через сложность, время реализации, время документации, объём и производительность. Вы довели спецификацию до такой точки, что она стала нереализуемой. Вы находитесь здесь не для того, чтобы создавать изящные вещи. Это второстепенно. Вы здесь для того, чтобы делать для корпорации деньги, а это значит делать всё, что в Ваших силах, для разработки надёжных проектов, реализуемых вовремя и с требуемыми эффективностью, стоимостью поддержки, ограничениями и т. д.

Можно сказать, что Роджер Гаурд был начальником Катлера, хотя в фирме DEC в это время было много сотрудников, которые не отчитывались ни перед кем, кроме собственной совести и нашего общего начальства — Гордона Белла¹⁰. Дэйв принадлежал к их числу, и в моей группе было несколько таких же вольных духом. Для руководителей, в частности, для Роджера и меня было подлинным испытанием определить, когда следует занять твёрдую позицию, когда подбодрить и повести за собой, а когда отойти в сторону. Роджер имел большой опыт в этом искусстве, я же только учился, но учился быстро.

У меня сохранилась небольшая подборка служебных записок, в том числе, адресованных и Дэйву Катлеру. Я их писал от руки и передавал лично, чтобы не посвящать посторонних, в том числе и машинисток (ведь электронной почты ещё не было). Записки эти, в коих язык и сильные выражения замечательны по своей ясности, если не сказать больше, и которые — дабы не вызывать ненужного замешательства у получателя — требовали крайней конфиденциальности, хранят следы сражений, разгоравшихся во время выпуска первой версии VAX/VMS.

Я храню также подобные записки, отправленные мне, моим подчинённым или руководителям. Читая их, я уже не чувствую себя задетым. Достаточно заметить, что в DEC во времена создания VAX слабые сотрудники не выдерживали гонки, а сильные, подвергнув друг друга уничтожающей критике, шли вместе пить пиво и, если и не становились друзьями, то сохраняли на всю жизнь взаимное уважение. Я отношу себя к этим выжившим.

Во время этих боёв одним из моих начальников в многомерной управленческой матрице фирмы DEC был Рон Хэм¹¹. После одного

¹⁰ Гордон Белл в прошлом — вице-президент по перспективным разработкам в DEC, был руководителем первого этапа создания ЭВМ VAX и ее программного обеспечения. В настоящее время — ведущий исследователь в корпорации Майкрософт.

¹¹ Рон Хэм руководил созданием компиляторов и файловых систем на заключительных этапах разработки PDP-11 и в начале разработки VAX.

особенно бурного заседания под моим председательством его настиг шквал телефонных звонков, требующих, чтобы он вмешался и отменил одно из моих решений. Рон Хэм сказал мне тогда, что порою ощущает себя дворником с метлой посреди главной улицы города после того, как по ней прошел цирк со слонами. Мои решения и мои позиции выжили в значительной мере благодаря мудрости Рона и его управленческому опыту, позволившим ему выстоять перед подобными натисками.

Идея же Общей библиотеки времени счёта для семейства VAX пережила все сражения и стала частью VAX/VMS.

Прочитую недавние слова Ричарда Гроува: «Общая библиотека времени счёта — большое достижение, и многие её черты мы сейчас воспринимаем как само собой разумеющиеся, среди них:

- общие соглашения о вызовах,
- многоязыковая интероперабельность во время счёта,
- система обработки исключительных ситуаций, поддерживающая [столь сложные языки, как] ПЛИ, Ада и т.д. и используемая в VMS для всех сообщений и информации об ошибках,
- современный многоязыковый отладчик исходного текста,
- интегрированная сетевая файловая система».

Все участники вышеописанного сражения вокруг Общей библиотеки времени счёта и других подобных битв сделали самую блестящую карьеру.

Дэйв Катлер, который уже был выдающейся фигурой в DEC, перешёл в компанию Microsoft, где стал отвечать за разработку нескольких ключевых версий Windows, в том числе NT. Он определённо заслуживает внесения в Книгу рекордов Гиннеса как единственный человек на Земле, возглавлявший разработку широко используемых операционных систем для по меньшей мере трёх поколений ЭВМ: PDP-11, VAX и серии компьютеров, базирующихся на Intel, с операционной системой Windows. Microsoft не стала ждать, пока Гиннесс примет решение, и уже присвоила Дэйву звание Senior Distinguished Engineer¹². Влияние Дэйва на будущее фирмы остается очень весомым.

Ричард Гроув, один из немногих в фирме Интел, удостоен звания Fellow.

Работа в DEC преподавала нам ценные уроки разработки большого программного продукта при экономном бюджете, показав, что можно сочетать передовые академические концепции с беспощадным прагматизмом. Опыт и навыки, полученные в DEC, были перенесены в

¹² Senior Distinguished Engineer — самое высокое техническое звание в корпорации Майкрософт

LPI Джоном Анкорном, мной и теми, кого мы впоследствии переманили из DEC.

Я был первым разработчиком Промежуточного Языка в проекте БЕТА, впоследствии названного там «Внутренним Языком». После того, как я покинул проект в 1973 году, мой подход, по-видимому, действительно слишком абстрактный и формальный для проекта, целью которого было создание работающего компилятора, был отвергнут. Я ничего не знал об этом новом подходе в то время, когда работал в LPI, да и до сих пор я с ним не знаком.

В последние годы существования Советского Союза, во времена отсутствия Интернета и электронной почты, которые кажутся теперь далёким прошлым, все мои контакты с проектом БЕТА полностью прекратились. В значительной степени именно я избегал их, дабы не запятнать моих бывших коллег связью с личностью, отнюдь не пользующейся благожелательностью со стороны советских властей. Я покинул страну в то время, когда эмиграция была редкостью, ей старались препятствовать всеми возможными способами и зачастую строго наказывали. Я обосновался в «цитадели капитализма», в Соединённых Штатах. Не следует осуждать членов БЕТА-команды за то, что они не предпринимали героических усилий для установления связи со мной — ведь ставкой могла быть их карьера, а то и сама возможность заниматься наукой.

Были немногие, почти тайные встречи (иногда в духе инспектора Клузо¹³) с тем или иным случайным гостем — участником проекта БЕТА, в том числе и с Андреем Петровичем Ершовым, но во время этих коротких и редких встреч мы не говорили подробно об этом проекте.

Был один, очень эпизодический письменный контакт между участниками БЕТА и мной. Помню, как в 1986 году, узнав о болезни Ершова и решив написать ему, я сильно колебался между желанием выразить свои тёплые чувства и стремлением не повредить ему контактом со мной. В письме, которое я, в конце концов, отправил, я намеренно использовал официально-строгое обращение «Уважаемый профессор Ершов» на бланке своей компании вместо привычного «Андрей Петрович» на личном рукописном письме, которое бы более подходило для этого случая. Я это сделал, чтобы обозначить некоторую дистанцию между мной и Ершовым. Я был уверен, что сам Ершов, как умный человек, все поймет правильно, а вот посторонних, которые могли бы письмо прочесть, я надеялся одурачить таким образом. Я до сих пор не знаю, получил ли Андрей Петрович это письмо.¹⁴

¹³ Инспектор Клузо — неуклюжий детектив из сериала «Розовая пантера» — Прим. переводчика

¹⁴ Английский оригинал см. в Электронном архиве А.П.Ершова по адресу <http://ershov.iis.nsk.su/archive/caindex.asp?lang=2&did=27213>

29 апреля 1986
Профессору Андрею П. Ершову
Вычислительный Центр
Сибирского отделения Академии Наук
Академгородок, Новосибирск, СССР

Уважаемый профессор Ершов:

Вчера я разговаривал с профессором Джеком Шварцем из Нью-Йоркского университета, и он упомянул о Вашей болезни.

Много мыслей сразу пронеслось в моей голове, и много дорогих воспоминаний пришло ко мне. Я хочу, чтобы Вы знали, насколько я огорчён известием о Вашей болезни, и что я очень желаю Вам быстрого и полного выздоровления. Если я могу что-нибудь для Вас сделать, пожалуйста, сообщите мне. Наталья присоединяется ко мне в моих пожеланиях.

Преданный Вам

Михаил Шварцман, президент LPI

P.S. Прилагаю маленькую брошюру, описывающую компанию, которую я основал несколько лет тому назад. Возможно, брошюра вас заинтересует.

Таким образом, контактов фактически не было. Проекты БЕТА и LPI-MLF, начатый в фирме LPI десять лет спустя после начала работы над системой БЕТА, развивались независимо друг от друга за исключением той связи, которую представлял собой я со своей осведомленностью, будь то к лучшему или худшему, о состоянии проекта БЕТА на 1973 год.

Я не принимал участия в разработке Промежуточного Языка в LPI. В мои функции входило то, что в рекламных проспектах называется «обеспечение стратегических направлений деятельности компании». Промежуточный Язык в LPI разрабатывался, в основном, Джоном Анкорном и его группой. Он имел в числе своих предшественников внутренний язык компиляторов для ПЛ/I, разработанный Робертом Фрейбургхаусом¹⁵ вначале для системы MULTICS и расширенный им позже, в его собственной компании Translation Systems, Inc. Фрейбургхаус стал впоследствии одним из основателей фирмы Stratus, успешного производителя компьютерных систем высокой надежности.

Что касается моей работы по обеспечению «стратегических направлений», то я нашёл старую записку, которую я написал, должно быть, после весьма бурной встречи с группой разработчиков, где «стратегические», так сказать, направления предлагались обеими

¹⁵ Роберт Фрейбургхаус — один из ведущих разработчиков системы MULTICS в МТИ

сторонами. Записка не нуждается в пояснениях и хорошо понятна каждому, кто руководил перспективными исследовательскими разработками.

СЛУЖЕБНАЯ ЗАПИСКА¹⁶

Компания LPI

Участникам заседания научно-исследовательской группы 21-4-1986

ОТ: Михаила Шварцмана

ДАТА: 1 мая 1986 г.

Десять лет тому назад, в DEC, я руководил научно-исследовательской группой из 15 человек и жонглировал своим вниманием, направляя его то на медленно умирающую PDP-11, то на машину, которая должна её заменить (ей стала VAX-11).

Иногда я чувствовал, что меня недооценивают (руководство уделяло больше внимания другим), иногда — что мной манипулируют и меня обманывают, порой я ощущал себя жертвой несправедливости, или чувствовал, что перетрудился, не способен контролировать окружающую обстановку, или разочарован процессом принятия решений в DEC. А иногда я чувствовал всё это сразу.

И вот что помогало мне идти вперёд:

Понимание того, что я вовлечён в нечто важное для DEC — нечто новое, перспективное и зависящее от моих суждений — будущая система VAX.

Чувство, что люди вокруг меня — у которых я работал, или которые работали у меня или со мной — мне нравятся и ценят мои усилия.

Некоторые книги (две или три), которые я читал и перечитывал снова и снова, когда всё шло действительно плохо.

Прилагаемая книга “*Bravely, Bravely, in Business*”¹⁷ — одна из них. Почитайте её.

Сердцем я с Вами, и я знаю, ЧТО Вы иногда чувствуете.

Михаил

У оптимизатора на уровне Промежуточного Языка и внешних интерфейсов были многочисленные практические и интеллектуальные предшественники, среди них выдающиеся оптимизаторы для Фортрана, разработанные в DEC, солидный внешний интерфейс, созданный Фрейбургхаусом, и интенсивные академические исследования по оптимизации в компиляторах.

Технология генераторов объектного кода развивалась от специализированных генераторов, переводящих с Промежуточного

¹⁶ Английский оригинал см. в Электронном архиве А.П.Ершова по адресу <http://ershov.iis.nsk.su/archive/eaimage.asp?lang=2&did=27214&fileid=173332>

¹⁷ Richard R Conarroe, *Bravely, Bravely, in Business* — AMACOM, 1978.

Языка на язык конкретной машинной архитектуры, к замечательной системе, которая использовала результаты академических исследований. Она допускала табличное представление целевого компьютера и позволяла порождать новый генератор за несколько недель. В рекламных изданиях LPI эта система называется «Генератором генераторов кода».

Конечно же, в LPI приходилось не только искать ускользающие решения проблемы создания работающей многоязыковой компилирующей системы. Нужно было также доставать деньги, привлекать таланты, иметь дело с большими фирмами — производителями компьютеров, собственные исследовательские отделы которых могли действовать враждебно или дружелюбно, но не всегда рационально, а то и прибегать к манёврам в стиле «плаща и шпаги» против нечестных конкурентов. В одном таком случае мы вместе с адвокатом LPI загнали в угол в суде по делам о банкротстве и заставили выйти из игры одного особенно мерзкого, закулисно действовавшего конкурента. Так была решена ещё одна из тысяч проблем на пути LPI-MLF.

LPI достигла успеха в создании системы LPI-MLF, потому что её основатели ясно понимали необходимость гибкого подхода при одновременном решении тысячи малых проблем без потери из виду общей картины и потому, что мы смогли поддерживать дисциплину и стимулировать разработчиков, что необходимо при руководстве научно-исследовательской командой, независимой по самой своей природе. Читателю, не знакомому с трудностями управления такой командой, можно лишь привести выражение «пасти стадо котов» как адекватное описание этого занятия.

Быть может, проект БЕТА, направленный на создание семейства работающих компиляторов, потерпел неудачу именно из-за свойственной советской Академии наук системы стимулов, не позволявшей привести к успеху проекта. Система эта обеспечивает материальными и академическими благами тех, у кого больше публикаций и научных степеней (это не относится к учёным, работающим на военные нужды). Эта извращённая система стимулов распространялась на всех участников проекта БЕТА, от самых молодых и вплоть до А.П.Ершова и его академического начальства.

Эта проблема присуща, по-видимому, многим другим академическим организациям, включая исследовательские подразделения американских университетов, но в Советском Союзе она усугублялась многими факторами, специфическими для командной экономики. Эти факторы в дальнейшем переплелись и привели к таким

зачастую непробиваемым препятствиям, как отсутствие мобильности научных работников, зависимость обеспечения жильем¹⁸ от академических званий и т. д.

Я рад заметить, что один из бывших участников проекта БЕТА Сергей Покровский¹⁹ высказал недавно наблюдение, сходное с моим, приведённым выше. Со столь характерной для него джентльменской, необвинительной отстранённостью он пишет: «Но [академическая] инерция ли общего замысла проекта [БЕТА], или [научное] воспитание, или ориентация на научный результат (естественная в виду диссертационных планов) — что-то резко изменило направленность работы М. И. Шварцмана, и с момента принятия его проекта за основу, он быстро стал обрастать чертами идеально-научной разработки ...». Я не вполне уверен, ЧТО именно Покровский понимал под словом «воспитание», но остальная часть его диагноза, даже изложенная с той осторожностью, которой Покровский славен, бьёт прямо в точку.

Справедливость требует заметить, что проект БЕТА был действительно сложным, а подобные начинания не удавались и там, где не существовало суровых ограничений командной экономики.

¹⁸ В этой связи см. письмо А.П.Ершова академику А.Г.Аганбегяну, которое сохранилось в архиве академика Ершова. Папка 104, лист 130 - Сост.

Глубокоуважаемый Абел Гезович!

Прошу у Вас следующего временного одолжения для Вычислительного центра. У нас есть весьма нуждающийся в улучшении жилищных условий сотрудник М.И.ШВАРЦМАН. Его супруга, ваша сотрудница Н.Г.БЕЛИХОВА, в свое время получила в Вашем институте 18-метровую комнату в трехкомнатной квартире в м-р «Щ». Насколько я знаю, обозримой перспективы расселения этой квартиры в Вашем институте сейчас нет.

Наша жилкомиссия говорит, что она в состоянии предоставить этой семье квартиру только в том случае, если она сможет временно использовать освобождающуюся комнату. Просьба к Вам состоит в том, чтобы взять у Вашего института в долг занимаемую семьей Н.Г.БЕЛИХОВОЙ комнату при условии возврата ее в течение трех лет или немедленно в случае ухода Н.Г.БЕЛИХОВОЙ из Вашего института. Соответствующие гарантии Гурий Иванович [Марчук] обеспечивает.

Я надеюсь на Ваше благожелательное отношение к этой просьбе. Прошу прощения за то, что вызов М.А.ЛАВРЕНТЬЕВА заставляет меня неожиданно уехать и не позволяет обратиться к Вам лично. Не откажите в любезности принять М.И.ШВАРЦМАНА и Н.Г.БЕЛИХОВУ по этому важному для них делу.

Уважающий Вас А.П.Ершов

¹⁹ Сергей Борисович Покровский — сотрудник Отдела программирования, участник проекта БЕТА, в настоящее время работает в Новосибирском отделении Интел — Россия.

И, конечно, как я уже говорил, оставшееся от проекта наследие идей и озарений — это совсем другая, гораздо более успешная история, и LPI-MLF является ее частью.

Закат LPI-MLF

К 1987 году система LPI-MLF столкнулась с ощутимыми проблемами, о которых речь будет идти ниже. Я предложил план, согласно которому следовало не замыкаться на LPI-MLF, а вложить средства в разработку нового перспективного продукта, которому предстояло заменить на рынке разработки компании Майкрософт: устаревавшей операционной системы MS-DOS и ее довольно примитивного тогда расширения — Windows. Но получить требуемое вложение средств оказалось трудно, потому что Совет директоров LPI разошелся во мнениях относительно перспектив системы LPI-MLF. Часть Совета не поддержала этих планов. В 1987 году я продал свои акции и оставил компанию.

Система LPI-MLF хорошо продавалась в течение немногих лет, с 1983 по 1990. В 1989 году Совет директоров фирмы наконец осознал проблему, решил не замыкаться на этой системе и сделал новые приобретения. LPI слилась с другой компанией и впоследствии стала называться «Liant Software Corporation». Ниже я привожу пылкий отчет моего преемника, Роя Финни²⁰, опубликованный в бюллетене для Liant клиентов Watchpoints, Winter 1991.²¹

Я рад сообщить, что прошедший финансовый год был периодом исключительного роста для Liant Software Corporation. К 30 сентября 1990 г. доходы фирмы выросли на 65% по сравнению с уровнем прошлого года. Такой существенный рост объясняется как значительным расширением спектра программных продуктов, предлагаемых Liant Software, так и увеличением спроса на открытые системы на рынке разработки прикладных программ.

Наши успехи отмечены в печати, в частности, журнал *Inc.* назвал нашу фирму среди 500 наиболее динамично растущих частных программистских компаний США в 1990 г. Основной причиной нашего успеха стала наша политика в области создания новых программных продуктов. В 1990 финансовом году Liant тратила на исследования и разработки на порядки

²⁰ Рой Финни — бывший президент фирмы Liant Software Corporation, которая образовалась после слияния LPI и еще одной фирмы.

²¹ Английский оригинал см. в Электронном архиве А.П.Ершова по адресу <http://ershov.iis.nsk.su/archive/caimage.asp?lang=2&did=27215&fileid=173334>

большой процент своей прибыли, чем это было в среднем по отрасли.

В 1991 году мы планируем продолжить быстрый рост группы Liant за счет приобретения компаний, приносящих новые продукты и методы, которые дополнят те, что существуют у нас, а также за счет выпуска новых программных продуктов и расширения наших продаж за рубежом.

Что касается компаний Language Processors, Inc., Ryan McFarland Corp. и Template Graphics Software, Inc., ныне входящих в группу Liant, то они по-прежнему будут соответствовать самым высоким стандартам отрасли и будут активно участвовать в формировании стандартов таких языков, как Кобол, Си и Си++, направляя все усилия на поддержку нашей философии открытых систем.

Мы планируем сделать 1991 год весьма успешным для наших клиентов, помогая вам в разработке и поддержке переносимых прикладных программ. Как обычно, мы приветствуем ваши замечания и предложения по улучшению нашей работы на ваше благо. Желаю вам успешного нового года,

Ваш Рой Финни

К тому времени старое руководство ушло (Джон Анкорн покинул фирму ещё в 1986 году), а угрозы, нависшие над LPI-MLF еще в 1987 году, наконец материализовались.

Liant сосредоточилась на разработке интерпретаторов для Кобола и сопутствующем инструментарии, в 2004 году в фирме работало шестьдесят человек в разных странах. Предлагаемые фирмой интерпретаторы с Кобола не базируются на технологии LPI-MLF, а получены в результате слияния. Система LPI-MLF, в которую добавлен компилятор для Си ++, всё ещё предлагается фирмой, но как продукт, «доставшийся по наследству», и, по-видимому, покупается плохо.

Более крупный и более старый конкурент компании Liant — Microfocus Corporation — также сосредоточен на Коболе. Обе компании обслуживают обширное наследство — прикладные системы на базе Кобола. Рынок компиляторов и интерпретаторов в наши дни весьма ограничен, так что годовой доход этих двух компаний составляет примерно 140 млн. долларов.

Система LPI-MLF, какой бы выдающейся в технологическом и рыночном отношении она ни была, просуществовала всего несколько лет. Она подвергалась нападкам и критике по самым разным направлениям и прекратила свое существование прежде, чем смогла перевоплотиться в программный продукт следующего поколения.

Иными словами, LPI-MLF сама достигла успеха и продвинула вперед технологию построения компиляторов настолько, насколько это было возможно. Используя аналогию, можно сказать, что LPI-MLF

была подобна большим паровозам 40-х годов в тот момент, когда паровые двигатели достигли пика своих возможностей, были построены самые тяжёлые локомотивы, но непосредственно перед тем, как эта технология уступила место гораздо более лёгким дизельным и электрическим двигателям. Или, используя другую аналогию, можно уподобить нашу систему огромным и тяжёлым мониторам с электронно-лучевой трубкой с диагональю в 21 дюйм в тот момент, когда технология CRT достигла своего предела, непосредственно перед тем, как им на смену пришли гораздо более лёгкие жидкокристаллические или плазменные экраны. Таких примеров можно привести множество. Система LPI-MLF оказалась гигантским паровозом в конце долгого периода развития технологии построения компиляторов.

Система LPI-MLF сравнительно быстро прекратила свое существование, все выглядело так, как будто атака на нее была спланирована, но, конечно, никто таких планов не строил.

- Язык Си, затем Си ++ и, наконец, Java довольно быстро вытеснили Фортран, Паскаль и ПЛ/I. Хотя Си ++ был быстро добавлен к семейству языков LPI, распространение бесплатных или почти бесплатных компиляторов для Си и Си ++ сделало такой шаг бесполезным.
- Системы баз данных и языки запросов вытеснили Кобол, RPG-II и ПЛ/I. Эти системы были органически чужды технологии, используемой в LPI-MLF, она мало что могла дать бы их пользователям. Число программистов, искусных в Коболе, RPG-II и ПЛ/I быстро уменьшалось. Потребность в компиляторах для этих трёх языков ограничена лишь пользователями, заинтересованными в оставшихся в наследство приложениях, отклоняющихся по природе своей от языковых стандартов.
- Исключительно быстрый рост быстродействия и объема памяти компьютеров сделал ненужной утончённость настройки LPI-MLF, покупать и поддерживать эту систему стало дорого (конечно, по сравнению со свободно распространяемым программным обеспечением). Преимущества во времени счёта и компиляции, а также по размерам требуемой памяти стали едва заметны на поколении компьютеров, скорость и память которых измеряются в мега-единицах. Компиляторы LPI-MLF с языков Си и Си ++ для новых приложений или с Кобола и Фортрана для старых, «оставшихся в наследство» задач ещё могли бы иметь спрос, но были быстро вытеснены легко создаваемыми, простыми интерпретаторами, многие из которых доступны бесплатно. Эти интерпретаторы, тягостно

неадекватные на медленных компьютерах, оказались вполне приемлемы на компьютерах новых, молниеносных, за исключением тех приложений (прогноз погоды, создание оружия), где даже сейчас никакие компьютеры не являются достаточно быстрыми.

- И, наконец, если не считать частных и специализированных областей, всё множество конкурирующих операционных систем и машинных архитектур сравнительно быстро приводится сейчас к одной архитектуре: серии всё более быстрых обратно совместимых микросхем фирмы Интел и серии (более или менее) обратно совместимых версий операционной системы Windows компании Майкрософт. Потребность в системе, которую можно было бы быстро перенести на радикально новую архитектуру, то есть необходимость и этого аспекта системы LPI-MLF, исчезла. Нет новых архитектур, на которые надо переходить. LPI-MLF стала последним «выморочным имуществом», предлагаемым фирмой Liant, наследницей LPI.

Пролить свет на судьбу LPI-MLF поможет наблюдение, что подобный процесс происходит и сейчас, в 2004 году. Система Windows осаждается сильной группой бесплатно или почти бесплатно распространяемых программных продуктов, таких как система Linux и ее инструментарий, подсистемы, офисные приложения на ее основе.

Операционная система Windows достигла, наконец, такой степени надёжности и функциональности, о которой в течение долгого времени можно было бы только мечтать. Возможно, она венчает собой определенный подход к созданию операционных систем. Дым схватки мешает участникам и наблюдателям сражений между Linux и Windows чётко понять, каков будет ее исход, какие маневры совершаются и на каких позициях стоят противники. Да и сами они едва ли знают, на каких позициях они находятся. Нельзя заранее утверждать, что Windows знаменует собой конец данной технологии. Далеко не всегда технологически превосходящий продукт (значительно усовершенствованная ОС Windows) в конце концов одерживает победу. Майкрософт знаком с этим жестоким утверждением лучше, чем многие другие, и часто извлекал из него выгоду. Смена поколений вычислительной техники и перемены на рынках делают туман над полем битвы еще более плотным, влияя на её исход, и определяют, в значительной мере, победу того или иного подхода. К тому же не обязательно победит один из двух данных подходов или какое-либо неожиданное их сочетание.

Свои выводы относительно того, что стало причиной «кончины» LPI-MLF, я делаю спустя пятнадцать лет после обсуждаемых событий.

Дым компьютерных битв восьмидесятых годов не позволял мне увидеть всю картину с той отчетливостью, с какой она представлена здесь, но уже в 1987 году я предчувствовал судьбу LPI-MLF, даже не зная всего того, что я знаю теперь.

У Билла Гейтса²², Дэйва Катлера и союзников Linux сейчас чертовски много проблем. Подождите: в каком-нибудь 2020 году кто-нибудь очень чётко объяснит нам всё, что происходит вокруг Windows и Linux.

Другие многоязыковые компилирующие системы

В этом разделе я не собираюсь представить исчерпывающий обзор всех попыток, как успешных, так и безуспешных, создания многоязыковой системы компиляторов. Это задача не сегодняшнего дня. Я рассмотрю только те попытки, которые так или иначе соотносятся с системой LPI-MLF.

Существует, по всей видимости, всеобщий закон, которому подчиняется любая деятельность людей: если доказано, что некая цель достижима, то и другим легче достичь той же цели. LPI-MLF был таким проектом-первопроходцем, и его достижения, пусть они и не копировались напрямую, в значительной мере определили судьбу других проектов, осуществление которых было невозможно всего лишь на год или два раньше. Так, LPI-MLF успешно включил в свою многоязыковую семью нерегулярный и неформализованный язык CO-BOL, стимулируя тем самым старания других разработчиков преуспеть в том же направлении.

Translation Systems, Inc. Эта компания, основанная Бобом Фрейбургхаусом, обладала наибольшим потенциалом для создания одного из первых многоязыковых компиляторов, но так и не сделала этого, поскольку такой проект не входил в ее бизнес-планы. В конце 1970 годов Translation Systems разработала несколько генераторов кода для своего Внутреннего Языка, но семейство ограничивалось всего двумя языками, ПЛИ и Фортраном; и хотя некоторые производители компьютеров, получившие лицензию от Translation Systems, создавали свои собственные генераторы кодов для Внутреннего Языка, семейство языков они не расширяли. Только компания Stratus расширила это семейство, но много лет не перенастраивала компилятор на другие архитектуры, ибо это не отвечало ее экономическим интересам.

Многоязыковое семейство компиляторов Stratus. После того, как Боб Фрейбургхаус оставил Translation Systems, он занялся разработкой программного обеспечения в новой компании, Stratus Computer, одним

²² Билл Гейтс — один из основателей и председатель Совета директоров корпорации Майкрософт.

из основателей которой он был. Начиная с 1980 года он осуществил в этой компании разработку и успешный последовательный выпуск нескольких компиляторов из многоязыкового семейства, которое к 1985 году расширилось и включало компиляторы для Фортрана, Кобола, ПЛ/I (подмножество G), Си и Паскаля. Это многоязыковое семейство компиляторов с общим Промежуточным Языком практически не перенастраивалось на новые архитектуры. В соответствии с бизнес-планами фирмы Stratus настройка компиляторов на несколько разных архитектур потребовалась лишь спустя много лет после того, как появилось семейство компиляторов Stratus, предназначенное для единственной архитектуры.

Многоязыковые компиляторы фирм LPI-MLF и Stratus имеют общий источник: гениальную интуицию Боба Фрейбургхауса, воплощённую в проекте Внутреннего Языка его компилятора для PL/I конца 1970-х годов. Этот внутренний язык послужил основой для разработки Внутренних Языков в обеих фирмах, LPI и Stratus, но к окончательной версии Внутреннего Языка для своих семейств компиляторов они пришли независимо. Интуиция Боба Фрейбургхауса и его опыт создания компиляторов существенно помогли этим двух командам успешно пройти по пути построения многоязыкового семейства компиляторов, по тому пути, который был до сих пор отмечен лишь неудачами.

Многоязыковое семейство компиляторов GEM. Ричард Гроув, ветеран баталий вокруг Общей библиотеки времени счёта времени разработки системы VAX, с 1984 по 2002 год был архитектором многоязыкового семейства компиляторов, планирование и разработка которого начались в фирме DEC в середине 1980х.

Это семейство компиляторов, GEM, постепенно включившее в себя языки Ада, Бейсик, Си, Си ++, Кобол, Фортран, Паскаль, ПЛ/I и Блисс, DEC разработал для своих компьютеров с архитектурой Alpha. Затем оно было перенесено на архитектуры MIPS и x86, но не получило широкого распространения на этих компьютерах.

Поучительный урок можно извлечь из судьбы такого технологически сложного программного продукта как семейство компиляторов GEM. Факторы, определяющие, будет ли система GEM переноситься на архитектуру Itanium, сложны, динамичны и не ограничиваются только техническими аспектами проблемы.

Рассмотрим следующие моменты:

- В 1998 году DEC был приобретён фирмой Compaq, которую затем купила фирма Hewlett-Packard, а Hewlett-Packard впоследствии передала команду разработчиков системы GEM фирме Intel. Четыре собственника сложной технологии в течение шести лет, вытекающая отсюда неуверенность по

поводу планов разработки и кадрового состава разработчиков — всё это может быть существенным фактором, потенциально дестабилизирующим накопленное программное обеспечение проекта и влияющим на перспективные планы.

- Теперь, когда архитектура Itanium возможно, станет целевой для семейства GEM, на процессы принятия решений и переноса системы влияют следующие факторы. За многие годы Intel развил технологию построения компиляторов, которая в настоящее время включает несколько промежуточных языков. Существует также множество технологий разработки внешних интерфейсов, оптимизаторов, отладчиков, генераторов кода и разнообразных инструментальных средств. Всё это имеет разные источники: кое-что приобреталось фирмой Intel, кое-что разработано в ней. Вопрос о том, какой промежуточный язык использовать, или же настраивать GEM на иной промежуточный язык, не тривиален с технической и управленческой точек зрения, так же как и вопросы использования тех или иных имеющихся компонентов, включая сюда и компоненты системы GEM. Проще говоря, существует немало команд разработчиков, состязающихся за право оснащать Itanium *своим* программным обеспечением, *своим* промежуточным языком, *своими* внешними интерфейсами, *своими* оптимизаторами, *своими* отладчиками, *своим* инструментарием, *своими* идеями, *своей* технологией. Создатели системы GEM — только одна из этих команд, по-видимому, самая новая и молодая среди разработчиков компиляторов в фирме Intel, и естественно, что ей не так просто конкурировать с другими. Вдобавок, смена четырёх хозяев за шесть лет вряд ли способствовало укреплению этой команды.
- Помните сравнение со «стадом котов»? Так вот, в случае с Itanium имеется несколько таких «стад», географически расположенных в разных местах. «Пасти» их, управлять проектом создания компиляторов для Itanium — это крайне сложная задача. В порядке вещей могут оказаться промедления с принятием решений, никого не удовлетворяющие компромиссы, смена курса, если верх одерживает другая команда.
- Сама архитектура Itanium должна доказать своё право на существование, и это ещё один фактор, определяющий судьбу переноса компиляторов GEM на Itanium.

Судьба технологии GEM для Itanium будет зависеть от многих факторов, в большинстве своём не технологических; большую роль

будут играть случайности, и только одним из факторов будет являться технологическая основательность структуры самой системы GEM. Лёгкость перенастройки семейства GEM была бы решающим фактором в более простой ситуации, но здесь она не является ни единственным, ни даже наиболее существенным фактором. Другие вышеперечисленные факторы, а также их непредсказуемые сочетания, — вот что определит успех семейства GEM для Itanium в качестве инструмента, которому пользователи отдадут предпочтение. В этой картине отсутствует ещё одна важная переменная: конкуренция со стороны компиляторов, в том числе почти бесплатных, предлагаемых разработчиками других фирм, не из Intel.

Какие проблемы стоят сейчас перед разработчиками многоязыковых системы компиляторов

Обстановка, в которой стремился к успеху дерзкий проект 70-х годов, БЭТА, была гораздо проще той, с которой имеет сейчас дело семейство компиляторов GEM. Конкуренция с другими разработчиками, конкуренция с другими технологиями и конкуренция с другими машинными архитектурами в значительно большей степени, чем раньше, определяют успех этого проекта. Продвижение, как и «выживание» программного продукта гораздо более динамично и непредсказуемо, чем в плановой экономике и в плановых научных исследованиях Академии наук бывшего Советского Союза.

В наше время существенным препятствием на пути к успеху многоязыкового семейства компиляторов является сокращение того сегмента рынка, который занимает большинство языков этого семейства. Эти языки, Фортран, ПЛ/I, Ада, Кобол, Паскаль и Бейсик, пользуются весьма ограниченным спросом. Это разработка оружия и исследование нефтяных залежей для ПЛ/I и Фортрана, оборонные проекты и космические исследования для Ады и Фортрана, оставшиеся в наследство прикладные коммерческие задачи — для Кобола.

Общим для всех этих областей является существование большого объема прикладных программ, написанных на версиях, которые слегка, а иногда и значительно отличаются от стандарта данного языка. Чтобы многоязыковая компилирующая система имела успех на рынке, она должна учитывать эти отличия. Модифицировать систему с учетом этих требований технологически несложно, но времени этот процесс требует очень много и его приходится повторять при каждой настройке многоязыкового семейства на новую архитектуру, поскольку приходится учитывать все отклонения от стандарта языка, возникшие с момента последней такой процедуры. Кроме того, на рынке всегда присутствует требование, чтобы эти языки сопрягались с различными

нестандартными подсистемами, например, теми, которые используются для введения или обнаружения параллелизма в программах. Эти новые и экспериментальные подсистемы не остаются стабильными и могут измениться за пару лет. Их постоянные изменения требуют соответствующих, столь же постоянных, изменений в интерфейсных расширениях данных языков.

Если многоязыковая система компиляторов попадает в трясину постоянных переделок, её весьма убедительное преимущество, короткий путь к рынку, нейтрализуется. Таким образом, многоязыковая система компиляторов успешнее всего реализуется для семейств стандартных и широко используемых языков, пользователи которых строго следуют языковым стандартам. В настоящее время такое семейство лишь одно — оно включает Си и Си ++.

Как я уже отмечал ранее, полную историю всех проектов создания многоязыковых компиляторов, как успешных, так и неудачных, еще только предстоит написать. Однако на примере пяти рассмотренных в этой статье проектов — БЕТА, LPI-MLF, Translation Systems, Stratus и GEM — видно, насколько возрастает со временем сложность обстановки, в которой разрабатываются проекты, и как вовсе не технические факторы влияют на успех, широту распространения и длину пути к рынку этих продуктов.

Способность перенастройки семейства Stratus не проверялась в течение многих лет, поскольку Stratus просто не имел разных целевых архитектур; у Translation Systems было лишь весьма узкое семейство компиляторов (Фортран и ПЛ/I), а перенастраиваемость семейства GEM является в весьма сложной управленческой проблемой.

LPI (ибо от этого зависел её коммерческий успех) была первой фирмой, предложившей широкое многоязыковое семейство компиляторов, реально перенастраиваемых на ряд весьма различных машинных архитектур.

Эпилог

Наследие Ершова в информатике далеко не ограничивается рассматриваемыми здесь проектами БЕТА и LPI-MLF, а также многими научными достижениями, о которых, несомненно, напишут его ученики и коллеги. Мне представляется гораздо более важным и непреходящим во времени наследием Андрея Петровича Ершова то сообщество ученых-исследователей и практиков-программистов, которое он создал и которое продолжает жить и сейчас.

Ершов обладал сверхъестественной способностью распознавать, привлекать и удерживать хороших, морально честных и благородных

людей. Они могли отличаться мерой своего таланта, ссорились и пререкались друг с другом, они были разными людьми, но обладали одной общей чертой — порядочностью. Исключения, тем более заметные, что они редки, можно объяснить главным образом молодостью Ершовских учеников и недомыслием, со временем проходящим.

Не следует забывать, что создание и сохранение большого коллектива морально достойных людей из года в год, из поколения в поколение — это само по себе огромное достижение, оценить которое может лишь тот, кому довелось работать в условиях командной экономики СССР и его сателлитов и может сравнить «уровень порядочности» в отделе Ершова со многими другими группами ученых.

И в этом качестве я утверждаю, что такое сравнение всегда было в его пользу. Добавим, что это наблюдение принадлежит тому члену Ершовской команды, чьи отношения с некоторыми коллегами не всегда были безоблачными.

Этот аспект наследия Ершова, который никогда не упоминался в научных трудах и скучных обзорах его академических работ, является одним из наиболее важных достижений Андрея Петровича, который, к сожалению, прожил недостаточно долго для того, чтобы по достоинству оценить это своё достижение и порадоваться ему.

Благодарности

Эту статью просмотрели Джон Анкорн, Джо Карчиди²³, Дэйв Катлер, Боб Фрейбургхаус, Ричард Гроув, Вадим Котов и Дейвид Какк²⁴. Они прямо и откровенно высказывали как поддержку и одобрение, так и критические замечания. Я принял далеко не все их предложения и несу полную ответственность за все изъяны этой работы.

Откровенный обмен мнениями с этими людьми вновь напомнил мне, как благодарен я судьбе за возможность работать с каждым из них в какой-то момент из прошедших тридцати лет, и как благодарен я судьбе за то, что мои пути пересекались с путями этих гигантов

²³ Джо Карчиди руководил Группой сопровождения и создания последовательных версий операционной системы VAX/VMS для компьютеров фирмы DEC

²⁴ Дейвид Какк — работал директором Центра исследований и разработок в области суперкомпьютеров в Университете штата Иллинойс в Урбана-Чемпейн, в настоящее время Intel Fellow, возглавляет Группу программного обеспечения и решений Отдела параллельных и распределенных систем в корпорации Интел.

программирования, для которых старомодные ценности их профессии остались ныне в той же силе, что и тогда, когда мы работали вместе:

- не уходить из проекта до тех пор, пока он не закончен, какую бы заработную плату ни предлагали конкуренты, и как бы ни сводила с ума структура управления этого проекта;
- вести себя лояльно по отношению к работодателю и хранить коммерческие тайны;
- сохранять преданность своей профессии, откладывая в сторону свои эмоции во имя получения наилучшего программного продукта.

Я благодарен всем вышеперечисленным рецензентам статьи за их замечания и за то, что представился случаем снова поработать с ними.

Рон Хэм и Джефри Шрисхейм²⁵ любезно поделились со мной своими наблюдениями и воспоминаниями, связанными с различными проектами разработки компиляторов, за что я им крайне признателен.

Высказывания относительно положения дел в фирмах, которые упомянуты в этой статье, и где работают мои некоторые рецензенты, принадлежат только мне и не обязательно разделяются ими.

Энтони Шварцман²⁶, мой сын и мой редактор в течение последних десяти лет, чей литературный талант не уступает его терпению при редактировании моих работ и его умению сохранять мой стиль при устранении моих ошибок и недосмотров, широко образован, что позволяет ему редактировать все мои тексты — посвященные как теории и практике инвестирования, так и информатике и технологии программирования. Его помощь трудно переоценить.

Мои бывшие коллеги из Российской академии наук Александр Рар, Сергей Покровский, Георгий Степанов и Александр Замулин перевели эту статью на русский язык и отредактировали перевод. Я высоко ценю быстроту и профессионализм их работы и дружеский диалог коллег, разделенных океаном. Иван Черкес²⁷ просмотрел русский перевод и сделал множество полезных замечаний.

Я отдаю должное спокойным и целенаправленным усилиям Натальи Черемных по координированию представления, перевода, редактирования и публикации этой статьи на русском языке.

²⁵ Джефри Шрисхейм ранее отвечал за перенос программного обеспечения Windows NT на архитектуру Alpha в корпорации DEC. В настоящее время возглавляет несколько перспективных проектов в корпорации EMC, Хопкинтон, Массачусетс.

²⁶ Энтони Шварцман — драматург и театральный режиссер, основатель театральной студии *Oblivion Productions* в Бостоне.

²⁷ Иван Черкес — старший консультант Швейцарской технологической фирмы TG Consultancy Services.

