# МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/17/9

УДК 681.3.06

## К ВОЗРОЖДЕНИЮ РУССКОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Г. П. Агибалов

*Национальный исследовательский Томский государственный университет, г. Томск, Россия*

**E-mail:** agibalov@isc.tsu.ru

Сообщается о цели, задачах и первых результатах работ по возрождению русского языка программирования, известного в Отечестве как ЛЯПАС — Логический Язык для Представления Алгоритмов Синтеза. Возрождение ориентировано на создание доверенного системного и прикладного программного обеспечения автоматизированного синтеза безопасных компьютерных систем логического управления критически важными объектами (космическими системами, ядерными установками, вооружёнными силами и т. п.). Среди рассматриваемых задач важное место занимает и аппаратная реализация ЛЯПАСа как ЛЯПАС-машины, или компьютера с ЛЯПАСными операциями в качестве машинных команд, выполняемых непосредственно на схемном уровне.

**Ключевые слова:** *ЛЯПАС, доверенное программное обеспечение, безопасные компьютерные системы логического управления.*

### Введение

В условиях засилья компьютерных технологий наиболее серьёзные угрозы безопасности страны (как внутри неё, так и извне) проистекают от использования в компьютерных системах управления критически важными объектами (космическими системами, ядерными установками, вооружёнными силами и т. п.) недоверенного программного обеспечения, заимствованного у своего же потенциального противника. Оно, это программное обеспечение, как правило, несёт в себе недокументированные закладки, через которые возможна утечка одной информации и навязывание другой, в том числе разрушительной, и обнаружить которые, даже в свободном коде, часто бывает невозможно, в особенности, когда он «запутан» мощными средствами обфускации. То, как эффективно такие закладки могут сработать, очень красноречиво показали многие известные военные операции США в Югославии, Ираке, Ливии.

Для предотвращения возможности подобных угроз предлагается к использованию в компьютерных системах управления собственное программно-аппаратное обеспечение на базе русского языка программирования. Да, такой язык есть, он существует с начала 1960-х годов, с той поры, когда в западном мире ещё только вступали в обиход первые языки программирования — Фортран и Алгол. Русским его назвали американцы — Russian programming language, а его оригинальное название звучит как ЛЯПАС (с ударением на второй слог) — Логический Язык для Представления Алгоритмов Синтеза. Он создан в научной школе прикладной дискретной математи-

ки (ПДМ) Томского государственного университета (ТГУ) под руководством Аркадия Дмитриевича Закревского.

Можно ли построить безопасные компьютерные системы управления на базе другого, «не русского» языка программирования (Си, Ада и пр.)? Наверно, можно, если создать при этом ещё и доверенный компилятор с этого языка. Но зачем, когда есть свой язык программирования, не менее, но много более приспособленный для задач логического управления и криптографической защиты информации, реализация которого много дешевле разработки доверенного компилятора для Си или Ада. Ведь речь идёт не о бытовых проблемах населения, которые могут решаться, да и то только до некоторой степени, с помощью «общечеловеческих» компьютерных систем. Речь идёт о безопасности Отечества, которую нельзя обеспечить «оружием» противника.

Реализация предлагаемого проекта возрождения ЛЯПАСа в полном объёме

— вернёт стране престиж компьютерной державы, каковой она была в 50–60-е годы XX столетия (пусть даже, если это будет не столько в сознании внешнего мира, сколько в самосознании собственного народа);

— обезопасит страну от угроз со стороны «кибертеррористов» в лице как отдельных людей, так и отдельных государств — потенциальных противников.

Конкуренции этому проекту не видно, поскольку все озабочены собственной прибылью, но не безопасностью страны и её престижем.

## 1. Безопасные компьютерные системы логического управления

Автоматическое логическое управление современными техническими объектами (летательными аппаратами, ядерными реакторами, безлюдным производством, другими технологическими процессами) требует применения доверенных управляющих компьютерных систем (КС), защищённых от вмешательства со стороны кибертеррористов или злоумышленных хакеров средствами криптографической защиты и разграничения доступа в соответствии с некоторой политикой безопасности. Эти свойства доверенности и защищённости КС определяют её безопасность, а КС с этими свойствами называется безопасной КС. Алгоритмы управления и защиты в управляющих КС могут иметь программную и аппаратную реализацию. Соответственно этому управляющая КС в общем случае состоит из программных и аппаратных модулей, решающих в совокупности все задачи защиты и управления в реальном времени, отсчитываемом моментами поступления управляющих команд, и операционной системы (ОС), организующей в реальном времени вызовы программных модулей и обращения к аппаратным модулям в соответствии с протоколом управления.

Применительно к вычислительному модулю (программному или аппаратному) реальность времени означает, что время его работы (от момента подачи данных на вход до момента выдачи результата на выход) укладывается в промежуток между соседними командами в протоколе управления, а применительно к ОС — что её полная реакция на любой запрос, инициализируемый управляющей командой извне, включая работу всех задействованных ею модулей, должна завершиться до прихода следующего запроса. Это свойство модулей и ОС называется далее реактивностью, а модули и ОС с этим свойством — реактивными. Реактивная ОС в безопасной управляющей КС, в свою очередь, должна обладать свойством безопасности, не допускающим возможности запуска модулей, способных (по ошибке или злому умыслу) обойти защиту компьютерной системы, обеспечиваемую модулями защиты, или существенно понизить её уровень.

Существующие ныне операционные системы общего пользования в семействах Windows, Linux и др., допуская настройку на адекватное использование средств защиты КС на аппаратном уровне, тем не менее по большому счёту не могут быть использованы в безопасной реактивной КС, главным образом, по причине недоверенности, проистекающей из закрытости или недостаточной контролируемости программного кода, а также из-за недостаточной их производительности для управления задачами в реальном времени, на которое они, будучи универсальными, специально не «заточены».

Кроме того, в безопасных КС свойство безопасности должно быть присуще также и программным реализациям прикладных алгоритмов и, в первую очередь, тех, что решают задачи защиты. Оно состоит в отсутствии в программах уязвимостей, через которые возможно проникновение в КС злонамеренного программного кода.

## 2. Математические и программные средства проектирования управляющих КС

В настоящее время на рынке научно-технической продукции нет ни доверенных КС, обладающих указанными свойствами реактивности и защищённости, ни программного обеспечения для их построения. Это вместе с чрезвычайной важностью безопасных КС для управления современными техническими объектами (подчас потенциально опасными, как Чернобыльская АЭС или Саяно-Шушенская ГЭС, например) и огромной трудоёмкостью их создания доказывает актуальность проекта, нацеленного на разработку математических и программных средств, автоматизирующих процесс проектирования таких КС.

Математические и программные средства, подлежащие разработке, должны включать в себя: язык программирования, наиболее подходящий для написания компонент требуемой КС; эффективные средства отладки и трансляции в машинный код программ, написанных на этом языке; расширяемую библиотеку прикладных программ на нём, охватывающую весь спектр задач как логического управления (на комбинационном, конечно-автоматном, микропрограммном и других уровнях представления), так и защиты информации (шифрование, аутентификация, цифровая подпись, контроль целостности и др.) и состоящую из программ двух типов: одни реализуют алгоритмы управления и защиты, а другие — алгоритмы синтеза схем, реализующих алгоритмы управления и защиты; операционную систему для управления задачами в КС.

Язык программирования должен гарантировать безопасность написанных на нём программ первого типа, а его операционные средства должны обеспечивать максимально возможную эффективность машинных программ и в то же время быть удобными для пользователей. Он должен допускать компактное и пригодное к публикации выражение программ и их эффективную трансляцию в машинный язык.

Задачи логического управления и защиты информации являются дискретно-математическими. Они ставятся на целых числах и конечных множествах с отношениями и операциями, могут быть перечислительными, поисковыми, оптимизационными, иметь аналитическое и алгоритмическое решения, допускать аппарат теории чисел, общей алгебры, комбинаторного анализа, дискретных функций, математической логики, кодов, графов, автоматов и т. д. и т. п. Длина значений целочисленных переменных в них может достигать нескольких тысяч бит.

Для того чтобы алгоритмы решения этих задач были эффективными, язык программирования для них должен содержать по существу машинные операции над булевыми векторами и их компонентами и иметь средства для создания эффективных

макроопераций над длинными такими векторами, над большими целыми числами и над большими булевыми и целочисленными матрицами. Заметим, что этих средств языка достаточно и для написания операционной системы.

Библиотека прикладных программ должна удовлетворять следующим требованиям безопасности и реактивности:

1) каждая программа в ней, реализующая алгоритм управления или защиты, не должна иметь уязвимостей, нарушающих её безопасность, и должна быть снабжена индексом реактивности — оценкой времени её выполнения при различных значениях параметров, полученной в результате предварительных исследований в компьютерном эксперименте на представительной выборке примеров;

2) каждая программа, реализующая алгоритм синтеза схем управления, должна быть также снабжена индексом реактивности, который в данном случае является оценкой времени работы доставляемых программой схем, полученной аналогичным образом. Значение индекса реактивности библиотечной программы влияет на широту класса реактивных управляющих КС, в создании которых может быть использована данная программа: чем он меньше, тем шире класс. Поэтому из двух равнофункциональных программ предпочтение при внесении в библиотеку отдаётся программе с меньшим индексом реактивности.

Операционная система также должна удовлетворять требованиям безопасности и реактивности, а именно: она не должна допускать выполнения в КС вредоносных действий и должна поддерживать исполнение прикладных задач (управления и защиты) с соблюдением заданного ограничения на ресурс времени.

## 3. О ЛЯПАСе

Среди всех языков программирования, известных в настоящее время, только один в достаточной степени удовлетворяет всем сформулированным требованиям. Это — ЛЯПАС.

Первые две версии ЛЯПАСа были разработаны соответственно к 1964-му [1 – 3] и к 1970-му [4] годам в ТГУ, а третья и последняя версия, называемая ЛЯПАС-М, — к концу 1974 г. в Институте технической кибернетики (ИТК) АН БССР [5, 6]. В отличие от предыдущих, в ней учтены байтовая структура данных и возможность измерения времени в современных компьютерах, а также система символов на современных отечественных устройствах отображения информации.

В своё время язык ЛЯПАС был реализован на всех отечественных ЭВМ, начиная с одноадресной машины «Урал-1» и кончая БЭСМ-6, а также на ЭВМ семейств ЕС, СМ и VAX (всё это в ТГУ совместно с ИТК) и на персональных компьютерах (ПК) первых поколений (в ИТК); на нём написан ряд крупных систем автоматического синтеза дискретных управляющих систем для многочисленных предприятий МЭП и МРП СССР [7 – 9]; его изучали, реализовывали и применяли также за рубежом — в США [10 – 12], Польше [13], Югославии [14], Чехословакии, ГДР; странами-участницами СЭВ он был принят в качестве международного языка программирования. Позднее, на волне демократии, с ликвидацией военно-промышленного комплекса и производства отечественных ЭВМ интерес к ЛЯПАСу заметно поугас. Сейчас же, когда осознание необходимости для национальной безопасности собственного программного обеспечения компьютерных систем управления возвращается, стал возвращаться интерес и к ЛЯПАСу.

Более подробно с историей создания и развития ЛЯПАСа, с его особенностями и характеристикой систем программирования, созданных на его основе для различных

типов компьютеров, можно познакомиться по обзору [15]. Здесь мы отметим только те его стороны, которые важны в решении дискретно-математических задач логического управления и защиты информации в реальном времени.

Язык ЛЯПАС был задуман и создан для разработки эффективных алгоритмов решения задач прикладной дискретной математики. Его применение не ограничивается написанием и отладкой программ, но предполагает их исследование в эксперименте на компьютере с одновременной доработкой как алгоритма, так и программы, его реализующей, с целью достижения наибольшей эффективности последней, характеризуемой экспериментальной оценкой времени исполнения программы: чем она меньше, тем выше эффективность. Соответственно выбрана и система операций и операндов в нём — с расчётом, с одной стороны, на максимизацию скорости работы получаемых программ, а с другой — на удобство пользователей. Это нашло отражение в двухуровневой архитектуре языка, первый уровень которой максимально приближен к машинному языку, а второй, представляющий собой иерархию библиотечных функций, делает язык сколь угодно выразительным. Наличие в языке «промашинного» первого уровня позволяет создавать на нём эффективные операционные системы, в том числе, что для нас немаловажно, обладающие определёнными выше свойствами безопасности и реактивности. В ЛЯПАСе количество, имена, типы операндов и возможные операции над ними фиксированы, а их размеры задаются и контролируются. Таким образом, ЛЯПАС является одним из тех редких языков программирования, на которых можно писать как безопасные прикладные программы, так и безопасные реактивные ОС. Кроме того, благодаря его двухуровневой архитектуре, для постановки ЛЯПАСа на конкретный класс машин достаточно написать транслятор на машинный язык лишь с первого уровня языка, а для подключения библиотечных функций использовать компилятор (препроцессор), написанный на ЛЯПАСе однажды и независимо от класса машин. Близость первого уровня ЛЯПАСа к машинному языку делает разработку такого транслятора не слишком трудоёмкой, а сам транслятор высокоскоростным.

Есть много других свойств ЛЯПАСа, подтверждённых на практике, которые существенны для систем программирования алгоритмов логического управления, защиты информации и синтеза схем для них. Это и компактность программ на ЛЯПАСе, их обозримость и пригодность к публикации, и реальная возможность написания и контроля их самим разработчиком алгоритма, и их более высокая эффективность (по сравнению с программами на других языках), и т. п.

## 4. СПО на базе ЛЯПАСа для проектирования безопасных управляющих КС

Всё перечисленное выше делает ЛЯПАС едва ли не идеальным средством для разработки системного и прикладного программно-аппаратного обеспечения (СПО), ориентированного на создание безопасных управляющих КС. Существовавшее когда-то СПО на основе ЛЯПАСа-М для ПК функционировало под управлением MS-DOS. Ориентированное на задачи логического проектирования, оно не имело в своей библиотеке программ криптографической защиты информации и разграничения доступа и не отвечает современным требованиям безопасности и реального времени. Предполагается, используя опыт и идеи этой разработки, создать СПО на основе ЛЯПАСа с операционной системой и библиотекой прикладных программ, удовлетворяющими как условиям безопасности и реального времени, так и требованиям к составу библиотечных программ. В нём безопасную реактивную ОС и транслятор с ЛЯПАСа в машинный язык планируется разрабатывать «методом раскрутки», без использования других (проме-

жуточных) языков и преобразований, понижающих доверенность и эффективность системного ПО.

При наличии такого СПО создание конкретной управляющей КС сводится к выбору из него готовой операционной системы, возможно, с адаптацией её ядра под платформу этой КС, а из его библиотеки — нужных прикладных программ с индексом реактивности, не превышающим заданного ресурса времени. В отсутствие таковых может потребоваться пополнение библиотеки недостающими программами. Последнего, впрочем, не понадобится при достаточной развитости библиотеки, достигаемой со временем. Выбранные программы, реализующие алгоритмы управления и защиты, используются в КС непосредственно в качестве её программных модулей, а программы, реализующие алгоритмы синтеза схем, — для синтеза соответствующих аппаратных модулей КС.

Достоинство результата выражается в отличительной совокупности следующих существенных признаков ожидаемого СПО:

— на все 100 % отечественный продукт с открытым и легко читаемым кодом;
— язык программирования, гарантирующий написание программ без уязвимостей;
— безопасный компилятор, исключающий возможность создания программ с уязвимостями;
— безопасная операционная система, исключающая возможность запуска злонамеренного кода;
— библиотека прикладных программ с реальным временем исполнения;
— операционная система, гарантирующая исполнение программ в реальном времени;
— гарантированное отсутствие недокументированных закладок в КС, создаваемых с его помощью.

С его реализацией обеспечиваются:

— улучшение потребительских свойств производимой продукции — управляющих КС;
— повышение производительности труда разработчиков продукции;
— уменьшение стоимости выпускаемой продукции;
— защищённость производимых КС от злоумышленника и ошибок персонала;
— инновационность производства, выход на рынок высокоинтеллектуальной и наукоёмкой продукции.

## 5. ЛЯПАС-машина

Когда-то А. Д. Закревский мечтал о создании L-машины [16] — компьютера для решения логических задач синтеза дискретных автоматов. В настоящее время, когда высокоскоростные логические микросхемы стали общедоступными, эта мечта может стать реальностью путём аппаратной реализации ЛЯПАСа с помощью существующих методов и систем автоматизированного проектирования [17 – 19]. Дело в том, что базовые операции в ЛЯПАСе являются одноадресными, что позволяет создать реально («в железе») ЛЯПАС-машину — недорогой одноадресный компьютер с системой команд, являющихся операциями ЛЯПАСа, в котором (компьютере) программы на языке программирования являются одновременно и исполняемым кодом — исполняются компьютером непосредственно без промежуточной трансляции в машинный язык и без интерпретации операций в них машинными подпрограммами. Это будет компьютер, в котором язык программирования высокого уровня (в данном случае ЛЯПАС) реализован аппаратно, благодаря чему скорость исполнения программ в нём будет на несколько порядков выше скорости их исполнения в существующих компьютерах.

Столь же простая аппаратная реализация других языков программирования вряд ли возможна ввиду их принципиально иного устройства.

Вместе с хранилищем ЛЯПАСных программ ЛЯПАС-машина может стать простейшей и вместе с тем эффективнейшей из мыслимых универсальной безопасной компьютерной системой управления любыми сложными объектами. Для её применения в управлении конкретным объектом достаточно будет загрузить в это хранилище программу на ЛЯПАСе, реализующую алгоритм управления данным объектом.

## 6. Первоочередные задачи возрождения ЛЯПАСа

Возрождение ЛЯПАСа с целью создания на его базе высокоэффективных безопасных компьютерных систем логического управления предполагает решение ряда теоретических и практических задач, направленных в первую очередь на разработку и исследование

- средств трансляции и отладки программ на ЛЯПАСе для их исполнения на современных компьютерах;
- алгоритмов и программ на ЛЯПАСе для логического управления;
- алгоритмов и программ на ЛЯПАСе для логического синтеза управляющих автоматов;
- алгоритмов и программ на ЛЯПАСе для криптографической защиты управляющей информации;
- реактивной ОС на базе ЛЯПАСа для управления в реальном времени;
- архитектуры ЛЯПАС-машины

и на реализацию последней на базе ПЛИС и (или) заказных интегральных схем.

С советских времён в научной школе ПДМ ТГУ сохранилась богатая библиотека программ на ЛЯПАСе для решения задач логического синтеза, которая с успехом может быть использована в синтезе аппаратных модулей будущих безопасных КС логического управления и ЛЯПАС-машины. Реанимация и адаптация этих программ под новые требования и условия также предполагаются.

## 7. Текущее состояние дела

Работы по возрождению ЛЯПАСа ведёт кафедра защиты информации и криптографии (ЗИиК) ТГУ. На начало 2012 г. «оптимизирована» под современные возможности отображения символика ЛЯПАСа и разработаны и запущены в опытную эксплуатацию программные средства, оживившие ЛЯПАС, а именно: интерпретатор ЛЯПАСа, компиляторы с ЛЯПАСа в языки Си и Ассемблер и отладчик ЛЯПАСных программ. С их помощью уже возможна отладка, исполнение, экспериментальное исследование алгоритмов на ЛЯПАСе и их доработка по результатам исследования. В ТГУ язык ЛЯПАС включён в образовательную программу по специальности 090301 — Компьютерная безопасность. Студенты кафедры ЗИиК, обучающиеся по этой специальности, не только осваивают программирование на ЛЯПАСе, но и участвуют в решении перечисленных выше задач по его полноценному возрождению, создавая в рамках курсовых и дипломных работ отдельные средства СПО на базе ЛЯПАСа для проектирования безопасных управляющих КС и для синтеза ЛЯПАС-машины.

## ЛИТЕРАТУРА

1. *Закревский А. Д.* Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов. Томск: Изд-во Том. ун-та, 1966. 266 с.

2. Логический язык для представления алгоритмов синтеза релейных устройств / под ред. М. А. Гаврилова. М.: Наука, 1966. 342 с.
3. Труды Сибирского физико-технического института. Вып. 48. Автоматизация синтеза дискретных автоматов. Томск: Изд-во Том. ун-та, 1966.
4. *Закревский А. Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.
5. *Закревский А. Д.* Язык программирования ЛЯПАС-М // Вычислительная техника в машиностроении. Минск: Ин-т техн. кибернетики АН БССР, 1974. С. 99–111.
6. *Закревский А. Д., Торопов Н. Р.* Система программирования ЛЯПАС-М. Минск: Наука и техника, 1978. 240 с.
7. Синтез асинхронных автоматов на ЭВМ / под ред. А. Д. Закревского. Минск: Наука и техника, 1975. 184 с.
8. Автоматизация проектирования цифровых устройств / под ред. С. С. Бадулина. М.: Радио и связь, 1981. 238 с.
9. *Панкратова И. А., Быкова С. В., Николаева С. В., Оранов А. М.* Система автоматического синтеза комбинационных схем СИНТЕЗ-Ф // Управляющие системы и машины. 1991. № 1. С. 3–9.
10. LYaPAS: A programming language for logic and coding algorithms / ed. M. A. Gavrilov and A. D. Zakrevskii. New York; London: Academic Press, 1969. 475 p.
11. *Charles J. and Albright Jr.* An interpreter for the language LYaPAS. University of North Carolina at Chapel Hill: Department of Computer Science, 1974. 127 p.
12. *Nadler N.* User Group for Russian Programming Language // IEEE, Newsletter for Computer-Aided Design. 1971. Iss. 3.
13. *Michalski A. and Wiewiorowski T.* Odra Ljapas. Warszawa: Computation-Centre Polish Academy of Sciences, 1970. 33 p.
14. *Tratnik I.* Seminar «Analiza in primerjava jezikov za podro'je digitalne tehnike». Lljubljani: Univerzav, 1979. 63 c.
15. *Торопов Н. Р.* Язык программирования ЛЯПАС // Прикладная дискретная математика. 2009. № 2(4). С. 9–25.
16. *Закревский А. Д.* Машина для решения логических задач типа синтеза релейных схем // Синтез релейных устройств. Труды Междунар. симп. по теории релейных устройств и конечных автоматов. М.: Наука, 1965. С. 346–356.
17. *Соловьёв В. В.* Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. Минск: ПК ООО Бестпринт, 1996. 252 с.
18. *Зотов В. Ю.* Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WtbPACK ISE. М.: Горячая линия-Телеком, 2003.
19. *Wollinger T., Guajardo J., and Paar C.* Cryptography on FPGAs. State of the art implementations and attacks // ACM Trans. on Embedded Computing Systems. 2004. V. 3. No. 3. P. 534–574.