

Введение

Tcl — это аббревиатура от Tool Command Language. Tcl означает как язык программирования, так и интерпретатор с этого языка. Интерпретатор Tcl можно без труда включать в состав прикладных программ. Tcl и используемый с ним набор инструментов создания графических пользовательских интерфейсов, называемый Tk, были разработаны профессором Джоном Остераутом (John Ousterhout) из калифорнийского университета. Tcl и Tk распространяются свободно, и их можно найти на многих узлах Internet и использовать для создания любых, в том числе коммерческих приложений. Интерпретатор Tcl был перенесен из Unix в системы DOS, PalmOS, VMS, Windows, OS/2, NT и Macintosh. Набор инструментов Tk может выполняться лишь в средах X Window, Windows и Macintosh.

О Tcl я впервые услышал в 1988 году, когда учился в калифорнийском университете. В то время мы занимались созданием сетевой операционной системы Sprite. Мы, студенты, работали над новым ядром, а Джон Остераут написал редактор и эмулятор терминала. В качестве языка он использовал Tcl, в результате чего пользователи получили возможность определять меню и выполнять другие действия по настройке своих программ. Мы работали во времена системы X10, и профессор Остераут планировал создать средства X Window, базирующиеся на Tcl. Это позволило бы взаимодействовать различным программам путем обмена Tcl-командами. На мой взгляд, возможность подобного взаимодействия инструментальных программ является одним из основных преимуществ Tcl.

Первоначально языку Tcl отводилась роль инструмента для настройки программ и организации высокоуровневого взаимодействия между ними. Основную же часть приложения, как нам казалось, должен был составлять скомпилированный код. Редактор `mx` и эмулятор терминала `tx`, написанные Остераутом, соответствовали этой модели. Однако впоследствии оказалось, что можно создавать приложения полностью на Tcl. Это стало возможным благодаря тому, что оболочка `wish` обеспечивала доступ к другим программам, к файловой системе и сетевым гнездам и в дополнение к этому позволяла создавать графический пользовательский интерфейс. Теперь нередко можно встретить приложения, содержащие тысячи строк Tcl-кода.

По мере накопления опыта работы с Tcl и Tk я все больше убеждался в их преимуществах, однако осознанию возможностей, предоставляемых данными средствами, предшествовали периоды разочарования в них. Это одна из причин, по которой была написана данная книга. Сотрудничая со специалистами Xerox PARC, я убеждался в преимуществах Tcl и Tk, но в то же время получал информацию о многих их недостатках. Мои коллеги по рабо-

те с удовольствием использовали Tcl и Tk в своих проектах и, обнаруживая изъяны, всегда ставили меня в известность. Я, в свою очередь, создал набор программных средств, позволяющих использовать возможности Tcl и Tk и избежать возникновения проблем. Данная книга задумана как практическое руководство, которое могло бы помочь читателю использовать Tcl и Tk в своей работе и в то же время избежать разочарований, которые в свое время испытал я.

С того момента, как я познакомился с Tcl, прошло около 14 лет, а первое издание данной книги было выпущено около восьми лет назад. Несколько лет я работал под руководством Джона Остераута, сначала в Sun Microsystems, а затем в Scriptics Corporation. В основном я писал программы на Tcl, в то время как остальные участники рабочей группы занимались реализацией средств Tcl на языке C. Я создавал такие приложения, как HTML-редакторы, программы для работы с электронной почтой, Web-серверы и базы данных. Опыт, полученный при этом, я попытался отразить в данной книге. Основная часть книги посвящена созданию Tcl-сценариев, однако внимание уделяется также написанию Tcl-расширений на языке C. В настоящее время я продолжаю работать с языком Tcl и получаю все большее удовольствие от этого. Надеюсь, что в данной книге мне удастся передать мое отношение к Tcl и Tk.

Некоторые аргументы в пользу выбора Tcl

Tcl, как и почти каждый язык сценариев, несколько напоминает такие языки оболочек Unix, как Bourne Shell (`sh`), C Shell (`csh`), Korn Shell (`ksh`) и Perl. Программы, написанные на языках оболочек, позволяют запускать другие программы. Они предоставляют разработчику выразительные средства (переменные, управляющие потоки и процедуры), достаточные для создания сложных сценариев, включающих в качестве составных частей существующие программы. Сценарии оболочки — незаменимое средство для автоматизации часто выполняемых рутинных операций.

Несмотря на некоторые общие черты, объединяющие Tcl с другими языками сценариев, он существенно отличается от них. Дело в том, что Tcl-интерпретатор может быть без труда включен в состав приложения. Tcl успешно справляется с ролью языка, используемого для настройки приложений. Используя Tcl, нет надобности придумывать новый формат файлов или командный язык для приложений. Включив в программу Tcl-интерпретатор, вы получаете возможность выполнять простые операции и объединять их в сценарий, полностью удовлетворяющий потребностям пользователей. Tcl также позволяет управлять приложениями из других программ, существенно упрощая тем самым их интеграцию.

Библиотека Tcl C предоставляет несложный для понимания интерфейс и проста в использовании. Данная библиотека реализует основные возможности интерпретатора и поддерживает набор базовых команд, позволяющих работать с переменными, формировать потоки управления и использовать процедуры. Богатый набор API предоставляет доступ к средствам операционных систем, позволяя запускать другие программы, работать с файловой системой и использовать сетевые гнезда. Tk добавляет команды, посредством которых можно создавать графические пользовательские интерфейсы. Tcl и Tk C API реализуют виртуальную машину, обеспечивающую работу в операционных средах Unix, Windows и Macintosh.

Виртуальная машина Tcl допускает расширение, поскольку приложения имеют возможность определять новые Tcl-команды. Эти команды можно связать с процедурами приложения, написанными на C или C++. В результате применения подобного подхода появляется возможность создать приложение как набор относительно простых элементов, написанных на компилируемом языке и оформленных как Tcl-команды. Для объединения простых элементов, или программных примитивов, в приложение используется Tcl-сценарий. Средствами уровня сценариев можно запускать другие программы, обращаться к файловой системе и непосредственно вызывать скомпилированные компоненты приложения посредством Tcl-команд, определенных разработчиком. На уровне C-программ вы можете вызвать Tcl-сценарии, читать значения Tcl-переменных и присваивать им новые значения и даже отслеживать работу Tcl-интерпретатора.

Многие Tcl-расширения свободно распространяются через Internet. Большинство из них включает библиотеку C, обеспечивающую новые функциональные возможности и Tcl-интерфейс к ней. В качестве примеров предлагаются системы управления базами данных, программы организации телефонных справочников, средства доступа к MIDI-контроллеру и другие программы. Tcl-команды, предназначенные для управления интерактивными программами, реализованы в продукте expect.

Наиболее популярным Tcl-расширением является Tk — набор инструментальных средств разработки графических пользовательских интерфейсов. В Tk определены Tcl-команды, которые позволяют создавать компоненты интерфейса и выполнять с ними различные действия. Применение сценариев для программирования пользовательских интерфейсов имеет следующие преимущества.

- Поскольку компиляция отсутствует, от редактирования исходного кода до запуска его на выполнение проходит совсем немного времени.
- Tcl-команды реализуют интерфейс гораздо более высокого уровня по сравнению со стандартными средствами создания пользовательских ин-

терфейсов, содержащихся в библиотеке C. Для реализации простого интерфейса достаточно нескольких команд. В то же время, если вам необходимо изменить ту или иную характеристику компонента, вы можете сделать это в любой момент.

- Пользовательский интерфейс может быть отделен от остальной части приложения. Разработчик имеет возможность сосредоточить все внимание на реализации основных функциональных возможностей приложения, а затем, не затрачивая больших усилий, создать необходимые интерфейсные средства. Базовый набор компонентов Tk, как правило, оказывается достаточным для формирования интерфейсов для большинства программ. По необходимости вы можете также разработать новые компоненты Tk на языке C, кроме того, в вашем распоряжении имеется большой набор компонентов, распространяемых по сети.

Для создания расширений могут использоваться самые различные языки, например Visual Basic, Scheme, Elisp, Perl, Python, Ruby или Javascript. Выбор конкретного языка зависит от вкусов разработчика. Tcl предоставляет простые языковые конструкции, похожие на средства языка C. Tcl-примитивы несложно создать путем написания процедур на C. Tcl прост в изучении. Благодаря его использованию некоторым разработчикам удавалось реализовать сложные проекты за очень короткое время (порядка нескольких недель). Такие результаты получались несмотря на то, что многие из них никогда ранее не писали программы на Tcl.

Приблизительно в то же время, когда вышло в свет первое издание данной книги, появился язык Java. Этот язык в короткие сроки приобрел большую популярность. При решении некоторых задач он с успехом заменил языки C и C++. Несмотря на то что Tcl был ориентирован на совместную работу с C-программами, его удалось быстро адаптировать для взаимодействия с виртуальной машиной Java. Там, где раньше шла речь о C или C++, теперь можно смело говорить "C, C++ или Java". Следует лишь учитывать некоторые детали, специфические для Java-программ. В данной книге интерфейс Tcl/Java не описывается, но на компакт-диске вы можете найти средства TclBlend. TclBlend загружает виртуальную машину Java в Tcl-приложение и позволяет вызывать Java-методы. Он также позволяет реализовывать Tcl-команды на Java вместо использования C или C++. Существует Tcl-интерпретатор, написанный на Java, который называется Jasl. Он имеет ряд ограничений по сравнению с Tcl-интерпретатором, написанным на C, но в тех случаях, когда вы не можете применить традиционный интерпретатор, Jasl оказывается незаменимым.

Язык Javascript, разработанный Netscape, предназначен для организации взаимодействия с элементами Web-страниц. Он часто используется при со-

здании интерфейсов на базе HTML. Однако Tcl позволяет реализовать более универсальные решения, применимые в самых различных приложениях. Дополнительный модуль Web-браузера Tcl/Tk позволяет выполнять Tcl-сценарии в среде браузера. В результате Tcl становится альтернативным решением не только для JavaScript, но и для Java. Дополнительный модуль, организующий выполнение Tcl-сценариев в среде браузера, описан в главе 20.

Версии Tcl и Tk

Tcl и Tk продолжают развиваться. Информацию о новых реализациях Tcl можно найти по адресу <http://www.beedub.com/book/>. Реализации Tcl и Tk выпускаются парами, предназначенными для совместной работы. В первом издании этой книги описывались средства Tcl 7.4 и Tk 4.0, но в ней также можно было встретить ссылки на Tk 3.6. В четвертом издании освещаются новые средства, добавленные в Tcl/Tk 8.4.

- Окончательная реализация Tcl 7.5 и Tk 4.1 была выпущена в мае 1996 года. В этих версиях был осуществлен перенос Tk в операционные среды Windows и Macintosh. В них же появился механизм защиты Safe-Tcl, предназначенный для выполнения апплетов, распространяемых по сети. Кроме того, в данных версиях была реализована поддержка сетевых гнезд и новой подсистемы ввода-вывода с поддержкой высокопроизводительного обмена, управляемого событиями.
- Реализации Tcl 7.6 и Tk 4.2 появились в октябре 1996 года. В них были модернизированы средства Safe-Tcl и обновлен диспетчер компоновки `grid`, впервые появившийся в Tk 4.1. Здесь же были реализованы виртуальные события (например, событие `<<Copy>>` может заменить `<Control-c>`), стандартное диалоговое окно и некоторые дополнительные команды работы с файлами.
- Tcl 7.7 и Tk 4.3 представляли собой внутренние реализации, предназначенные для разработки дополнительных модулей Tcl/Tk для Web-браузеров Netscape Navigator и Microsoft Internet Explorer. Их разработка реально осуществлялась параллельно с работой над версиями 7.6 и Tk 4.2. Дополнительный модуль был создан для различных платформ, включая Solaris/SPARC, Solaris/INTEL, SunOS, Linux, Digital Unix, IRIX, HP/UX, Windows 95, Windows NT и Macintosh. Этот модуль поддерживает Tcl-апплеты в составе Web-страниц; в нем используется механизм защиты на базе Safe-Tcl.
- К основным особенностям Tcl 8.0 относится динамический компилятор Tcl-кода, применение которого позволяет многократно повысить эффективность выполнения Tcl-сценариев. Tcl 8.0 поддерживает строки, ко-

торые могут содержать символы с нулевым кодом. Для Tcl-сценариев компилятор прозрачен, однако разработчики, которые занимаются созданием расширений, должны знать некоторые новые C API, в противном случае они не смогут воспользоваться новыми возможностями. Процесс создания Tcl 8.0 несколько затянулся. Это было связано с переходом Джона Остераута из Sun Microsystems в Scriptics Corporation. Реализация 8.0p2 была готова осенью 1997 года и вскоре приобрела популярность. Однако выход окончательной реализации, 8.0.5, задержался до весны 1999 года.

- Новая версия Tk была создана лишь для того, чтобы обеспечить наилучшее соответствие Tcl 8.0. Tk 8.0 включает механизм поддержки шрифтов, не зависящий от конкретной платформы, платформенно-ориентированные меню и панели инструментов, а также платформенно-ориентированные компоненты, внешний вид которых согласуется с другими приложениями, выполняющимися в системах Windows и Macintosh.
- Tcl/Tk 8.1 обеспечивают поддержку Unicode и содержат новые средства обработки регулярных выражений, которые предоставляют возможности, не уступающие Perl 5. Кроме того, эти версии поддерживают работу потоков, что позволяет без труда включать Tcl в состав многопоточковых приложений. Разработчики Tk проделали огромную работу, чтобы найти подходящие шрифты для представления символов Unicode, а каталоги сообщений существенно упрощают написание приложений с интерфейсом на различных языках. В работе над Tcl/Tk 8.1 также сказался переход Остераута из Sun в Scriptics. Первая альфа-реализация была закончена осенью 1997 года, а окончательная реализация, 8.1.1, была готова лишь в мае 1999 года.
- Реализация Tcl/Tk 8.2 задумывалась в основном для того, чтобы исправить замеченные ошибки и повысить надежность работы системы. Вместе с этим API библиотеки Tcl C был дополнен некоторыми средствами, увеличивающими набор доступных расширений. Работа над Tcl/Tk 8.2 закончилась быстро, и данные реализации вышли в свет летом 1999 года.
- Tcl/Tk 8.3 была существенно расширена по сравнению с предыдущими реализациями. В Tk были приняты шаблоны Dash и Image, предложенные Дженом Найтменсом (Jan Nijtmans). Реализация 8.3.0 была выпущена в феврале 2000 года, а последний вариант, 8.3.5, был реализован в октябре 2002 года.
- При разработке Tcl/Tk 8.4 основное внимание уделялось производительности, кроме того, были добавлены интерфейс виртуальной файловой системы и три компонента Tk: `spinbox`, `labeledframe` и `panedwindow`.

Работа над этой реализацией продолжалась достаточно долго. Первая бета-версия была выпущена в июне 2000 года, а реализация 8.4.2 вышла в мае 2003 года.

Расширения Tcl и Tk

Tcl разработан так, что дополнения создаются в виде расширений, при этом базовые средства Tcl не затрагиваются. В настоящее время доступны многочисленные расширения. Информацию о них можно получить в Web по следующему адресу:

<http://www.tcl.tk/resource/>

Однако для внесения некоторых изменений приходится модифицировать базовый комплект Tcl/Tk. Если вы хотите приложить свои усилия к разработке новых вариантов Tcl/Tk, ваша помощь будет с благодарностью принята. Необходимую информацию о деятельности группы разработчиков Tcl и Tk вы можете найти по следующему адресу:

<http://www.tcl.tk/cgi-bin/tct/tip/>

Исходные коды Tcl и Tk поддерживаются в рамках проекта SourceForge.

<http://www.sourceforge.net/projects/tcl>

<http://www.sourceforge.net/projects/tktoolkit>

Сообщения о замеченных ошибках и предложения о доработках регистрируются в базе данных. Наибольшие шансы быть принятыми имеют дополнения к исходному коду, сделанные в соответствии с руководствами *Tcl Engineering Manual*. В руководствах описаны правила оформления кода, требования к тестированию и документации.

Tcl в World Wide Web

Знакомство с материалами, посвященными Tcl, можно начать со следующих Web-страниц:

<http://www.tcl.tk/>

<http://tcl.activestate.com/>

<http://www.purl.org/NET/Tcl-FAQ/>

Узел Wiki часто обновляется по инициативе пользователей. На нем публикуются многочисленные материалы о Tcl и его расширениях (в качестве инициатора публикации новой информации можете выступить вы).

<http://wiki.tcl.tk/>

На Web-странице данной книги содержатся сведения об ошибках для всех изданий. За материалами, публикуемыми на этом узле, слежу лично я и планирую делать это и в дальнейшем.

<http://www.beedub.com/book/>

Кроме того, полезную информацию вы можете найти на Web-узле Prentice Hall.

<http://www.prenhall.com/>

Ftp-архивы

Ниже перечислены некоторые из FTP-узлов, содержащих информацию о Tcl.

<ftp://ftp.tcl.tk/pub/tcl>

<ftp://src.doc.ic.ac.uk/packages/tcl/>

<ftp://ftp.luth.se/pub/unix/tcl/>

<ftp://ftp.sunet.se/pub/lang/tcl>

<ftp://ftp.cs.columbia.edu/archives/tcl>

<ftp://ftp.funet.fi/pub/languages/tcl>

Для доступа к этим узлам можно использовать любые браузеры, например Mozilla, Netscape, Internet Explorer и Lynx.

Группы новостей

Группа `comp.lang.tcl` отличается высокой активностью. В ее рамках действует форум, посвященный обсуждению Tcl. Сообщения о Tcl-расширениях и приложениях поступают в группу `comp.lang.tcl.announce`. Для доступа к материалам групп новостей можно воспользоваться специальными Web-службами. Для этого достаточно ввести `comp.lang.tcl` в поле редактирования на следующей Web-странице:

<http://groups.google.com>

На кого рассчитана эта книга

Данная книга может быть полезна как программистам, впервые приступающим к работе с Tcl, так и квалифицированным специалистам. Всем категориям читателей я рекомендую внимательно прочитать главу 1. Программная модель Tcl достаточно проста, но она существенно отличается от многих других языков программирования. Данная модель основана на подстановке строк, поэтому, внимательно разобравшись с ней, вы избежите многих проблем в работе. В книге приведено большое количество примеров, посвященных эффективному использованию средств Tcl и Tk. Практически в каждой

главе содержатся таблицы, в которых приводится информация о Tcl-командах и компонентах.

Предполагается, что читатель данной книги имеет некоторый опыт программирования, однако даже новичок, читая эту книгу, сможет достаточно быстро приобрести необходимые навыки. Знание языков оболочек Unix поможет вам, но оно не является обязательным требованием. Там, где изложение материала тесно связано с принципами работы оконных систем, приводится соответствующая информация. В главе 2 описываются детали использования Tcl и Tk в системах Unix, Windows и Macintosh.

Как работать с данной книгой

Лучше всего в процессе чтения книги выполнять на компьютере приведенные в ней примеры. Часто программисты испытывают затруднения при изучении Tcl. Дело в том, что в справочной информации по Tcl и Tk отсутствуют примеры использования соответствующих языковых конструкций, а реальные программы оказываются плохо документированы. Данная книга призвана предоставить читателю недостающие сведения.

Для изучения конкретных команд Tcl и Tk я рекомендую использовать интерактивную справочную информацию. В ней приводится детальное описание соответствующих языковых конструкций. В данной книге содержатся многие данные из справочной системы, но в ней отсутствуют детали, которые могут изменяться в зависимости от реализации Tcl и Tk. HTML-версии страниц интерактивной справочной системы можно найти на компакт-диске, прилагаемом к данной книге.

Примеры программ

Данная книга поставляется с компакт-диском, на котором записаны коды всех рассмотренных в книге примеров. Диск может читаться в системах Unix, Windows и Macintosh. Кроме программ на нем находятся версии Tcl и Tk, которые были доступны на момент выхода данной книги, а также многие из программ, свободно распространяемых по Internet. Исходные коды примеров можно также найти в Web по следующему адресу:

<http://www.beedub.com/book/>

Соглашения о представлении материала

Примеры кода представлены в книге по-разному. Наиболее важные из них отделены от остального текста и оформлены в виде листингов. Коды

программ на Tcl и C представлены **моноширинным шрифтом**. Результаты выполнения Tcl-команд, кроме того, выделены **курсивом**. Следует заметить, что символы => не являются частью данных, возвращаемых командой.

```
expr 5 + 8
=> 13
```

Моноширинный шрифт также применяется для представления имен Tcl-команд и C-процедур в тексте.

В книге часто встречаются правила использования Tcl-команд. Имя команды и неизменяемые ключевые слова отображаются **моноширинным шрифтом**. Указания на то, что в вызове команд присутствуют значения переменных, опций или другие данные, выделены **курсивом**. Если параметр не является обязательным, слева и справа от него помещаются знаки вопроса. Пример, объясняющий использование команды `set`, приведен ниже.

```
set имя_переменной ?значение?
```

Пиктограммы



Информация, которой желательно уделить дополнительное внимание, помечается пиктограммами.

Как организована данная книга

Материал книги разделен на семь частей. В первой главе описаны основные средства Tcl. В первой главе рассматриваются механизмы, лежащие в основе языка. Этот материал очень важен. Вам обязательно надо ознакомиться с ним, если вы хотите эффективно использовать Tcl. Если вы уже программировали на Tcl, вам все равно следует хотя бы бегло просмотреть главу 1. В главе 2 описываются детали использования Tcl и Tk в системах Unix, Windows и Macintosh. В главе 3 представлен пример приложения (CGI-сценарий), который иллюстрирует основные особенности составления программ на языке Tcl. Остальные главы части I посвящены детальному рассмотрению основных Tcl-команд. В частности, здесь обсуждаются работа со строками, типы данных, использование потока управления и процедур, а также области видимости. Завершается часть I рассмотрением средств обмена с файлами и запуска других программ.

В части II описаны расширенные средства составления Tcl-программ. Вначале обсуждается команда `eval`, позволяющая динамически генерировать Tcl-программы. Дополнительные возможности по обработке строк предоставляют регулярные выражения. Если обработка данных составляет существенную часть вашего приложения и если приложение работает медленно, то вам, возможно, удастся повысить его производительность за счет использования

средств поддержки регулярных выражений. Пространства имен позволяют разделить глобальную область видимости процедур и переменных на отдельные области. Unicode и каталоги сообщений существенно упрощают создание многоязыковых приложений. Библиотеки и пакеты позволяют организовать код так, чтобы он мог использоваться в различных проектах. Средства интроспекции предоставляют важную информацию о внутреннем состоянии Tcl. Средства ввода-вывода, управляемого событиями, позволяют создавать приложения, работающие одновременно с несколькими клиентами. Для реализации HTTP-протокола, позволяющего получать Web-страницы, служат сетевые гнезда.

В последних нескольких главах части II описываются инструментальные средства разработки приложений и особенности их использования на различных платформах. Средства Safe-Tcl служат для организации защищенной среды, в которой можно выполнять Tcl-апплеты. Для работы в Web может использоваться расширяемый Web-сервер TclHttpd, написанный полностью на Tcl. На базе TclHttpd легко создавать прикладные программы; можно также интегрировать сервер с существующими приложениями, создавая тем самым для них Web-интерфейс. Starkit — новое эффективное средство, позволяющее создавать пакеты и выполнять доставку приложений Tcl/Tk. Для организации файловой системы в пакете используются средства поддержки виртуальной файловой системы (VFS).

В части III содержатся общие сведения о Tk. Возможности Tk проиллюстрированы рядом примеров. В этой же части описывается связывание — установление соответствия между Tcl-командой и событием, например щелчком мышью или нажатием клавиши. Три последних главы части III посвящены диспетчерам компоновки Tk, которые предоставляют мощные средства по организации пользовательского интерфейса.

В части IV данной книги описываются компоненты Tk. С помощью компонентов реализуются кнопки, меню, полосы прокрутки, текстовые метки, поля редактирования, текстовые области, поддерживающие различные шрифты, холст для рисования, окна списков и линейные регуляторы. Для каждого компонента имеются средства, позволяющие изменять его конфигурацию и организовывать взаимодействие с прикладными программами, однако поведение компонента, реализованное по умолчанию, чаще всего удовлетворяет требованиям разработчиков. Для упрощения настройки компонентов и обеспечения согласованности элементов интерфейса предусмотрены базы данных ресурсов.

В части V описываются те инструменты Tk, которым не было уделено достаточного внимания в предыдущих частях. В частности, здесь рассматриваются выделения, вопросы передачи фокуса ввода и стандартные диалоговые окна. Кроме того, детально рассматривается работа со шрифтами, с цветом

и изображениями — элементами, общими для всех компонентов Tk. Оканчивается данная часть несколькими примерами программ достаточно большого объема.

Часть VI представляет собой введение в программирование Tcl-расширений на языке C. Данная часть включена в книгу для того, чтобы читатель мог составить правильное представление о том, как можно дополнять Tcl командами, написанными на C, и интегрировать Tcl с существующими приложениями.

Каждая глава части VII посвящена одной из реализаций Tcl/Tk, упоминаемых в данной книге. В этих главах описываются новые возможности, а также характеристики, которые могут изменяться от версии к версии. Эти характеристики приходится учитывать, адаптируя программы для новых версий Tcl/Tk.

Особенности четвертого издания книги

В четвертом издании изложение материала ведется с использованием версии Tcl/Tk 8.4, в которой были реализованы многие новые возможности. В Tcl появилась виртуальная файловая система, позволяющая учитывать структуру каталогов при доставке приложений, а также работать с удаленными ресурсами на FTP- и Web-узлах посредством интерфейса с обычной файловой системой. В четвертое издание добавлена глава 22, посвященная Telkit и Starkit и использованию базы данных Metakit для хранения сценариев и других файлов. Благодаря VFS эти файлы размещаются в приватной файловой системе. Starkit — новое эффективное средство, позволяющее создавать пакеты и выполнять доставку приложений Tcl/Tk. В четвертое издание добавлена также глава 21, посвященная средствам поддержки многопоточковых программ. Эти средства оказываются очень полезными при включении Tcl в многопоточковые приложения. В данном издании также рассматриваются новые компоненты Tk. Инкрементный регулятор похож на поле редактирования, однако в нем реализованы дополнительные возможности. Фреймы с метками предоставляют новые возможности по оформлению окон. Компонент `panedwindow` представляет собой специализированный диспетчер компоновки, реализующий новые возможности по организации пользовательских интерфейсов.

Благодарности за помощь в подготовке первого издания книги

Я благодарю сотрудников Xerox PARC за помощь и понимание. Многие советы, приведенные в книге, стали результатом не только моей личной ра-

боты, но и опыта моих коллег. Вопросы, заданные Дейвом Николсом (Dave Nichols), воодушевили меня на изучение базовых механизмов работы интерпретатора Tcl. Ден Свайнхат (Dan Swinehart) и Лоренс Батчер (Lawrence Butcher) своими замечаниями поддерживали во мне постоянное желание продолжать работу. Рон Фредерик (Ron Frederick) и Берри Керчивел (Berry Kerchival) применили Tk для разработки графических интерфейсов своих систем. Полученные при этом результаты еще раз подтвердили высокую эффективность данного инструмента. Беки Барвел (Becky Burwell), Рич Голд (Rich Gold), Карл Хаусер (Carl Hauser), Джон Максвелл (John Maxwell), Кен Пайер (Ken Pier), Марвин Теймер (Marvin Theimer) и Моэн Вишвонет (Mohan Vishwanath) ознакомились с моими первыми набросками. Их комментарии и замечания оказали неоценимую помощь в дальнейшей работе над текстом. Керин Петерсен (Karin Petersen), Билл Шилит (Bill Schilit) и Терри Ватсон (Terri Watson) все время искали способы нестандартного применения Tcl, чем вносили приятное разнообразие в работу. Я хочу особо поблагодарить Марка Вейсера (Mark Weiser) и Дуга Терри (Doug Terry) за помощь и поддержку.

Выражаю благодарность автору Tcl и Tk Джону Остерауту. Разработанные им системы — самые замечательные из всех, которые мне когда-либо приходилось встречать. Джон снабжал меня последними реализациями Tk 4.0, благодаря чему я имел возможность изучить их новые возможности еще до выпуска первой бета-версии.

Благодарю программистов, с которыми мне приходилось общаться. От них я узнал много нового. Я считаю Джона ЛоВерсо (John LoVerso) и Стефена Алера (Stephen Uhler) самыми талантливыми разработчиками Tcl-программ из всех, кого я когда-либо знал.

Выражаю искреннюю благодарность тем, кто прочитал мои первые наброски и высказал ценные замечания. Это Пайери Девид (Pierre David), Клиф Флинт (Clif Flynt), Симон Кенион (Simon Kenyon), Эджен Ли (Eugene Lee), Дон Лайбс (Don Libes), Ли Мур (Lee Moore), Джо Мосс (Joe Moss), Хэдор Шемтов (Hador Shemtov), Френк Стаджано (Frank Stajano), Чарльз Тейер (Charles Thayer) и Джим Торнтон (Jim Thornton).

Многие присылали мне предложения и замечания по почте. Здесь я привожу имена лишь некоторых из них. Это Мигель Энджел (Miguel Angel), Стефен Бенсен (Stephen Bensen), Джеф Блейн (Jeff Blaine), Том Чарнок (Tom Charnock), Брайан Купер (Brian Cooper), Патрик Д'Круз (Patrick D'Cruze), Бенойт Десросайерс (Benoit Desrosiers), Тед Даннинг (Ted Dunning), Марк Эйчин (Mark Eichin), Пол Фриберг (Paul Friberg), Карл Гаутиер (Carl Gauthier), Дэвид Гердес (David Gerdes), Клаус Хекенберг (Klaus Hackenberg), Торкл Хэсл (Torkle Hasle), Марти Херст (Marti Hearst), Джин-Пайери Герберт (Jean-Pierre Herbert), Джейми Хонан (Jamie Honan), Норман Клейн (Norman Klein), Джо Констан (Joe Konstan), Сьюзен Ларсон (Susan Larson), Хекен Лилдже-

грен (Hekan Liljegren), Лайонел Маллет (Lionel Mallet), Деджен Милоджичич (Dejan Milojicic), Грэг Миншелл (Greg Minshall), Бернд Мор (Bernd Mohr), Вилл Морс (Will Morse), Хейко Нердменн (Heiko Nardmann), Герд Неуджебауэр (Gerd Neugebauer), Кери Рензема (Cary Renzema), Роб Рипел (Rob Riepel), Дэн Шенк (Dan Schenk), Джин-Гай Шнейдер (Jean-Guy Schneider), Элизабет Шолл (Elizabeth Scholl), Карл Швамб (Karl Schwamb), Рони Шапиро (Rony Shapiro), Питер Симани (Peter Simanyi), Винс Скехен (Vince Skahan), Билл Стамбо (Bill Stumbo), Глен Вандербург (Glen Vanderburg), Лэрри Вирден (Larry Virden), Рид Вейд (Reed Wade) и Джим Вайт (Jim Wight). К сожалению, я не смог ответить на все письма.

Выражаю признательность редакторам и всем сотрудникам Prentice Hall. Высококвалифицированные профессионалы Марк Тауб (Mark Taub), Линн Шнейдер (Lynn Schneider) и Керри Кидон (Kerry Reardon) очень помогли мне в работе над книгой.

Благодарности за помощь в подготовке второго издания книги

Я снова хочу поблагодарить Джона Остераута за поддержку во время работы в группе Tcl/Tk компании Sun Microsystems. Остальные участники группы делали все возможное для того, чтобы превратить Tcl и Tk в настоящие инструменты для создания кроссплатформенных программ. Скотт Стентон (Scott Stanton) занимался переносом Tk на PC, а Рей Джонсон (Ray Johnson) обеспечивал работу Tk в среде Macintosh. Джейкоб Леви (Jacob Levy) реализовал систему ввода-вывода, управляемую событиями, Safe-Tcl, и дополнительный модуль для браузера. Брайан Левис (Brian Lewis) создал компилятор Tcl. Кен Кори (Ken Corey) занимался интеграцией Tcl и Java, а также помогал в работе над SpecTcl. Сид Полк (Syd Polk) обобщил систему меню для работы с платформенно-ориентированными компонентами в средах Macintosh и Windows. Колин Стивенс (Colin Stevens) работал над поддержкой шрифтов и средствами интернационализации для Tk.

Стефен Алер (Stephen Uhler) заслужил мою искреннюю благодарность за интересные примеры, которые я использовал в своей книге. Он руководил группой разработчиков SpecTcl. Стефен создал библиотеку HTML, пользуясь которой я написал редактор. Мы вместе работали над первыми версиями TclHttpd. Он научил меня писать компактный эффективный Tcl-код и использовать регулярные выражения при решении многих задач. Я многому обязан Стефену и надеюсь, что хоть в чем-то смог быть полезен ему.

Я снова благодарю Марка Тауба, а также Эйлин Кларк (Eileen Clark) и Марту Вильямс (Martha Williams) из Prentice Hall. Джордж Вильямс (George Williams) помог мне собрать файлы для размещения на компакт-диске.

Благодарности за помощь в подготовке третьего издания книги

Джон Остераут продолжает играть важную роль в развитии и распространении Tcl, на этот раз уже как основатель Scriptics Corporation. Я хочу поблагодарить всех сотрудников Scriptics, в особенности Сару Дениэлс (Sarah Daniels), Скотта Стентона (Scott Stanton), Рея Джонсона (Ray Johnson), Брайана Серлеса (Bryan Surles), Мелиссу Чавла (Melissa Chawla), Ли Бернхарда (Lee Bernhard), Суреша Шастри (Suresh Sastry), Эмиля Скеффона (Emil Scaffon), Пэт Р. (Pat P.), Скотта Редмана (Scott Redman) и Берри Керчивела (Berry Kerchival). Все сотрудники Scriptics заслужили искреннюю благодарность за то, что создали в компании атмосферу, благоприятствующую творческой работе. Автор многих публикаций Джерри Пик (Jerry Peek) помог мне полезными советами. Кен Джонс (Ken Jones) рассказал об эффективных инструментах индексации.

Большое спасибо всем читателям, приславшим свои отзывы и предложения. Сведения о новых способах использования Tcl всегда интересовали меня.

Я благодарю также редакторов Prentice Hall Марка Тауба, Джоан Макнамара (Joan McNamara) и Джоан Юрел (Joan Eurell). Они помогли мне вовремя окончить работу над третьим изданием книги.

Благодарности за помощь в подготовке четвертого издания книги

Благодаря Джефу Хоббсу (Jeff Hobbs) и Кену Джонсу (Ken Jones) работа над данным проектом была успешно завершена. Джефф широко известен среди программистов, использующих Tcl. Новые реализации Tcl/Tk выходили во многом благодаря его руководящим усилиям и интенсивной работе. Кен — известный преподаватель Tcl, и благодаря моему сотрудничеству с ним данная книга пополнилась новым интересным материалом.

Я хочу поблагодарить участников Tcl Core Team; усилиями этой группы, Tcl становится все более мощным инструментом и приобретает новые возможности.

Я хотел бы поблагодарить Джин-Клод Випплер (Jean-Claude Wippler) и Стива Лендерса (Steve Landers) за Metakit, Tclkit и Starkits. Эти инструменты предоставляют уникальные возможности по созданию пакетов и доставке Tcl-приложений. Я надеюсь, что данные технологии будут развиваться и дальше. Информацию о Starkit я получал также от некоторых пользователей. Сообщения мне прислали Роберт Течентин (Robert Techentin), Стив Блинкхорн (Steve Blinkhorn), Френк Серджент (Frank Sergeant), Арден Маркус (Arjen Markus), Уве Колоска (Uwe Koloska), Лэрри Вирден

(Larry Virden), Том Крехбиел (Tom Krehbiel) и Дональд Портер (Donald Porter).

Выражаю свою благодарность руководству и сотрудникам Prentice Hall за постоянную поддержку. Марк Тауб продолжает быть “крестным отцом” этой книги. Важную роль в подготовке четвертого издания к выпуску сыграла Кетлин Керен (Kathleen Caren).

И, наконец, я благодарю свою жену Джоди за любовь, доброту, терпение и понимание в долгие часы работы над книгой. У меня три сына, Кристофер, Дэниел и Майкл. Мне приходится постоянно отвечать на их вопросы, поэтому я уверен, что умственная деградация мне не грозит.

Как связаться с автором книги

Я всегда готов выслушать мнение читателей о книге. Мой адрес: welch@acm.org. Если вы включите в поле subject письма слово “book” либо название данной книги, это поможет мне правильно сортировать приходящие письма. Новую информацию о данной книге и других сферах моей деятельности вы можете получить по адресу <http://www.beedub.com/>. На Web-узле я веду учет замеченных ошибок, поэтому ваши комментарии будут приняты с благодарностью.

Ждем ваших отзывов

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг. Наши координаты:

E-mail:	info@williamspublishing.com
WWW:	http://www.williamspublishing.com

Информация для писем из:

России:	115419, Москва, а/я 783
Украины:	03150, Киев, а/я 152