Программирование на Эрланге.

> Наш мир существует во времени параллельно и одновременно.
>
> Если мы хотим писать программы, которые ведут себя также как объекты реального мира, то эти программы должны иметь параллельную структуру.
>
> Используйте для этого язык, который был специально разработан для написания параллельных приложений и их разработка станет для вас гораздо проще.
>
> Модель программирования Эрланга — это то как мы, на самом деле, думаем и взаимодействуем.

Джо Армстронг

Начало.

> О, нет! Еще один язык программирования! Зачем мне его учить? Разве уже существующих не достаточно?

Я понимаю такую вашу реакцию. Сегодня в мире уже существует множество языков программирования, зачем же изучать еще один?

Вот вам пять причин, почему вам может быть полезно изучение Эрланга:

- Вы хотите писать программы которые будут работать быстрее, если их запустить на много-ядерных компьютерах.
- Вы хотите писать отказоустойчивые приложения, которые могут быть модифицированы прямо в процессе их работы.
- Вы что-то слышали о «функциональном программировании» и вас заинтересовало, насколько оно полезно на практике.
- Вы хотите использовать язык который был многократно проверен в «боевых» условиях реального мира в масштабных промышленных продуктах и который имеет громадные библиотеки и активное сообщество пользователей.
- Вы не хотите стачивать ваши пальцы от набивания множества строчек кода.

Разве все это возможно? В разделе 20.3 *Работа с SMP Эрлангом* мы рассмотрим некоторые программы, которые линейно ускоряют свою работу на 32-ядерном компьютере. В главе 18 *Разработка систем с помощью OTP,* мы рассмотрим как создавать высоко-надежные системы, которые работают круглосуточно, без

остановки, в течении многих лет. В разделе 16.1 *Путь к обобщенному (Generic) серверу,* мы поговорим о написании серверов, приложения на которых могут быть модифицированы «на лету», без остановки работы сервера.

Во множестве мест этой книги, мы будем восхвалять добродетели функционального программирования, которое, в частности, запрещает наличие у программ побочных эффектов. Побочные эффекты и параллельное программирование просто несовместимы. Либо вы программируете с побочными эффектами последовательные программы, либо параллельные, но без побочных эффектов. Выбор за вами, но третьего, просто, не дано.

Эрланг — это язык в котором параллельность программ встроена в сам язык, а не связана с операционной системой. Эрланг облегчает создание параллельных программ, путем моделирования окружающего мира в виде множества параллельных процессов, которые взаимодействуют, только обмениваясь сообщениями. В мире Эрланга все параллельные процессы существуют без взаимных блокировок, методов синхронизации и возможности воздействия на общую память, так как ее просто не существует.

Программы на Эрланге могут состоять из тысяч и миллионов очень «маленьких» процессов, которые могут исполняться на одном процессоре компьютера, на много-ядерном процессоре, или же на сети компьютеров.

Краткое описание глав книги.

- Глава 2 «Введение» - это, типа, быстро «запрыгнуть и осмотреться» что это все такое.
- Глава 3 «Последовательное программирование» - это первая из двух глав о последовательном программировании на Эрланге. В ней, например, вводятся такие понятия, как «соответсвие по образцу» и «неразрушающее присвоение».
- Глава 4 «Исключения» посвящена обработке исключений. Не бывает программ без ошибок. И эта глава посвящена обнаружению и обработке ошибок в последовательных программах на Эрланге.
- Глава 5 «Продвинутое последовательное программирование» - это вторая глава о последовательном программировании на Эрланге. Она рассматривает несколько тем по-сложнее и заканчивает рассмотрение последовательного программирования.
- Глава 6 «Компиляция и запуск программ» рассказывает о различных способах

компиляции и запуска ваших программ.

- В главе 7 «Параллельность» мы делаем пересадку. Это не-техническая глава. Он посвящена, скорее, идеологии того, как мы программируем вообще и как мы видим наш окружающий мир.

- Глава 8 «Параллельное программирование» посвящена параллельности в Эрланге. Как нам создать параллельные процессы? Как они будут взаимодействовать между собой? Насколько затратно по времени их создание?

- Глава 9 «Ошибки в параллельных программах» рассматривает эту тему. Что произойдет при падении процесса? Как мы можем обнаружить его падение и что мы можем с этим сделать?

- Глава 10 «Распределенное программирование» является введением в эту тему. В ней мы рассмотрим несколько небольших распределенных программ и покажем как их запускать на кластере узлов Эрланга или на независимых компьютерах в сети, используя socet механизмы взаимодействия.

- Глава 11 «Маленький IRC» посвящена одному этому приложению. Мы сложим вместе темы параллельности и socet-распределенности в нашем первом, нетривиальном, приложении: небольших IRC-подобных клиенте и сервере.

- Глава 12 «Интерфейсы взаимодействия» посвящена вопросам взаимодействия программ на Эрланге и программ на других языках.

- Глава 13 «Программирование с Файлами» дает множество примеров использования файлов в программах.

- Глава 14 «Программирование с Socet-ами» дает примеры таких программ. Мы рассмотрим как построить параллельные и последовательные серверы на Эрланге. В завершении этой главы мы рассмотрим второе масштабное приложение: SHOUTcast медиа-сервер. Который может раздавать MP3 данные используя SHOUTcast протокол.

- Глава 15 «ETS и DETS: механизмы хранения больших объемов данных» описывает низко-уровневые модули Эрланга ets и dets. Модуль ets предназначен для очень быстрых, «деструктивных» операций с хеш-таблицами и их данными в памяти, а dets разработан для хранения данных на диске.

- Глава 16 «Введение в OTP» именно этому и посвящена. OTP — это набор библиотек и операционных процедур Эрланга предназначенных для построения серьезных промышленных приложений. В этой главе вводится понятие поведения (behavior) – центральной концепции OTP. Используя типы поведения мы можем

сосредоточиться на только на функционале компонент наших приложений, в то время как поведенческое окружение OTP позаботится обо всем остальном. Это окружение, например, может само реализовать отказоустойчивость или масштабируемость нашего приложения, в то время как (написанные нами) модули обратного вызова приложения реализуют всю его специфическую функциональность. Эта глава начинается с общего обсуждения как работают поведения вообще, а потом переходит к описанию поведения gen_server (обобщенный сервер), как части стандартной библиотеки Эрланг OTP.

- Глава 17 «Mnesia («Мнезия») - база данных Эрланга» рассказывает о базе данных (СУБД) Mnesia. Мнезия — это интегрированная в Эрланг СУБД с очень быстрым временем отклика в реальном времени. Она может быть сконфигурирована на репликацию данных по нескольким, физически различным, узлам для обеспечения отказоустойчивости.

- Глава 18 "Создание систем при помощи OTP" - это следующая глава об OTP. Она посвящена практическим вопросам создания OTP-приложений. Реальные приложения нуждаются для работы во множестве мелких, необходимых деталей. Они должны запускаться и останавливаться определенным образом. Если они или их компоненты падают, они должны быть определенным образом перезапущены. Также нужен журнал ошибок, позволяющий определить, что же произошло при падении. Эта глава как раз и описывает все эти мелочи, позволяющие создавать полноценные ОТП приложения.

- Глава 19 "Мультиядерная прелюдия" - это краткое объяснение, почему Эрланг хорошо приспособлен для программирования многоядерных компьютеров. Мы обсудим, в общем, проблему разделяемой памяти и параллельных программ с передачей сообщений и почему мы свято верим, что параллельные языки без взаимного влияния идеально подходят для программирования многоядерных компьютеров.

- Глава 20 "Программирование многоядерных ЦПУ" посвящена программированию многоядерных компьютеров. Мы поговорим о том, как сделать так, чтобы ваша программа эффективно использовала многоядерность компьютера. Мы представим вам несколько абстрактных понятий используемых для ускорения работы последовательных программ на многоядерных компьютерах. Кроме того мы проведем ряд измерений и напишем нашу третью серьезную программу - движок полнотекстового поиска. Для его реализации мы, для начала, напишем функцию mapreduce - высокоуровневую функцию, используемую для распараллеливания вычислений на некоторое множество вычислительных элементов.

- Приложение A - описывает систему типов применяемых при документировании

функций в Эрланге.

- Приложение B - описывает установку Эрланга в операционной системе Виндоус (и как сконфигурировать Emacs, во всех операционных системах).
- Приложение C - содержит каталог ресурсов по Эрланг.
- Приложение D - описывает библиотеку lib_chan предназначенную для создания socet-распределенных приложений.
- Приложение E - рассматривает техники анализа профилирования, отладки и трассировки вашего кода.
- Приложение F - содержит однострочные описания самых используемых стандартных модулей Эрланга.

Еще раз с начала.

Однажды программисту попалась в руки книга, описывающая забавный язык. У него был незнакомый синтаксис, равенство вовсе не означало равенства, переменным не разрешалось изменяться. Хуже того, он даже не был объектно-ориентированным. А программы были, как бы это сказать, немного другими...

Но не только программы были другими, но и весь подход к программированию был другим. Автор все время говорил про параллельность и распределенность программ, про их отказоустойчивость и про метод программирования, называемый параллельно-ориентированное программирование - что бы это там не значило.

Но некоторые примеры были весьма забавными. В тот вечер, программист рассматривал пример программы для чатов. Он был очень маленьким и легким для понимания, даже не смотря на немного странный синтаксис. Невозможно, чтобы все было так просто.

Основная программа была простой, а с помощью еще нескольких строчек кода появились и возможности обмена файлами и зашифрованные разговоры. Программист начал нажимать клавиши на клавиатуре...

О чем, вообще, эта книга?

Она о параллельности. Она о расределённости. Она об отказоустойчивости. Она о функциональном программировании. Она о написании распределенных параллельных

систем без взаимных блокировок и общих областей, а только на чистом обмене сообщениями. Она о ускорении ваших программ на многоядерных процессорах. Она о написании распределенных приложений, которые позволяют людям взаимодействовать друг с другом. Она о методах дизайна и поведения систем для написания отказоустойчивых и распределенных приложений. Она о моделировании параллельности мира и отображении этих моделей в компьютерные программы. Этот процесс я называю параллельно-ориентированным программированием.

Мне было очень приятно писать эту книгу. Я надеюсь, что вам будет также приятно ее читать.

А теперь, начните читать книгу, писать примеры и получать от этого удовольствие.

Благодарности.

Многие люди помогли мне в подготовке этой книги и я хотел бы всех их сердечно поблагодарить здесь.

Во-первых, Дейва Томаса, моего редактора: Дейв учил меня как писать и засыпал неимоверным количеством бесконечных вопросов. Почему это так? А почему это эдак? Когда я начал писать книгу, Дейв сказал мне что я пишу ее в стиле "проповедей с высокой скалы". Он сказал мне: "я хочу, чтобы ты просто говорил с людьми, а не проповедовал им". Так книга стала гораздо лучше, спасибо, Дейв.

Далее, у меня был небольшой комитет моей поддержки, состоящий из экспертов по языку. Они помогли мне решить, что оставить за бортом рассмотрения. А также, помогли мне прояснить пару деталей, тяжелых для рассмотрения. Спасибо (без определенного порядка) Бьерну Густавсону, Роберту Вердингу, Костису Сагонас, Кеннет Лундин, Ричарду Карлосону и Ульфу Вайгеру.

Спасибо также Клаису Викстрёму который дал ряд полезных советов по Мнезии, Рикарду Грину за SMP Эрланг, и Хансу Нильсону за морфологический алгоритм, использованный в индексировании текстов.

Шон Хинди и Ульф Вайгер помогли мне понять, как использовать многие внутренние аспекты OTP, а Сергей Алейников обьяснил мне активные сокеты, так что даже я смог их понять.

Хелен Тейлор (моя жена) критически прочитала несколько глав этой книги и обеспечила меня многими сотнями чашек чая в подходящие для этого моменты. Более того, она как-то справлялась с моим, скорее, одержимым поведением последние семь месяцев. Спасибо, также, Томасу и Клэр, и еще спасибо Баху, Генделю, Зорро, Дейзи и

Доррис, которые помогали мне не сойти с ума, мурлыкали, когда их чесали и помогали мне добраться домой по правильному адресу.

И, наконец, спасибо всем читателям бета-версии этой книги, которые прислали мне свои замечания. Я вас проклинал и я безмерно благодарен вам. Когда была опубликована первая бета этой книги, я был просто не готов, что она будет прочитана за два дня и просто распотрошена по кусочкам, каждая страница, вашими комментариями. Но это привело к созданию, гораздо более улучшенной версии книги, чем я вообще мог себе представить. Когда (как это было несколько раз) десятки людей писало "я не понимаю, что написано на этой странице", я был вынужден обдумать все снова и переписать указанный материал. Люди, спасибо вам всем, за вашу помощь.

Джо Армстронг

Май 2007-го года.