Проект «Тектоника»: предвидение опыта TEX 21-го века

Питер К.Г. Уильямс

Абстрактный

Tectonic — это программный проект, построенный на основе альтернативного движка TEX, ответвленного от X TEX. Он был создан для изучения ответов на два вопроса. Первый вопрос касается документов: в эпоху технологий 21-го века, когда интерактивные дисплеи, вычисления и подключение к интернету, как правило, дешевы и повсеместны, какие новые формы технических документов стали возможны? Второй вопрос касается инструментов: как мы можем использовать те же самые технологии, чтобы лучше предоставить людям возможность создавать отличные технические документы?

Предпосылка проекта Tectonic заключается в том, что, хотя TEX может быть почтенным, он по-прежнему является идеальной системой для создания технических документов «21-го века», но проект с независимой идентичностью и инфраструктурой может добиться прогресса способами, которые невозможны в основном TEX. Tectonic компилируется с использованием стандартных инструментов Rust, устанавливается как один исполняемый файл и загружает файлы поддержки из готового дистрибутива TEX Live по требованию.

В прошлом году долгожданная работа над родным языком HTMLВывод наконец-то начал приземляться, включая, возможно, новую схему рендеринга математики Unicode, вдохновленную подмножеством шрифта. Текущие усилия направлены на то,чтобы конкретизировать этоHTMLподдержка с использованием *X TEX: Программа* как тестовый случай, с целью существенного улучшения документации самого Tectonic. Хотя Tectonic позиционирует себя как «вне» традиционного TEX в определенном смысле, проект не мог бы существовать без усилий всего сообщества TEX, которому автор и проект в большом долгу.

1 Введение

Эта статья будет мотивировать проект Тектоника (§2), обсудите некоторые из его отличительных характеристик (§ 3), вникнуть в то, как это реализуетсяHTMLвыход (§4) и кратко обсудим перспективы его будущего (§5).

2 Мотивация

Я (PKGW) буду мотивировать проект Tectonic несколько стилизованной историей моего путешествия по экосистеме TEX. Я имею опыт научных исследований (астрономия), и, насколько я помню, впервые я использовал TEX для набора задач в колледже. Я до сих пор помню удовлетворение от создания прекрасно набранного уравнения и понимание того, что в мире нет другого инструмента

Питер К.Г. Уильямс

doi.org/10.47397/tb/43-2/tb134williams-tectonic

Image Sharing and Recycling

change_be96.1

It is not hard too see how reasonably sized papers, especially scientific articles, can require the use of many hundreds of external images. For this reason, image sharing and recycling is of critical importance. In this context, ``sharing'' refers to the use of one image in more

Рисунок 1:Скриншот типичного LaTEKC2HTML выход. Отwww.sci.utah.edu/~macleod/latex/latex2html/Enode8.html,выбрано произвольно.

которые могли бы набирать математические тексты почти так же хорошо — по крайней мере, ни один из них не мог бы свободно использоваться студентом колледжа.

Во время моей учебы в колледже (2002–2006, если вам так хочется спросить), было ясно, что Интернет и Всемирная паутина были на пути к преобразованию общества. Но по большей части дизайн в Интернете был печально известен своим плохим качеством. LaTEX можно преобразовать вHTMLи визуализировал, но результаты напоминали Рисунок 1: в основном читаемые, но абсолютно хуже того, что можно было бы сделать вPDF-файл.И покаHTMLдокументы имели возможности гипертекста, они, как правило, были*статический* документы: слова и цифры, расположенные на странице в факсимиле «реальной вещи»: чернила на бумаге.

Для меня было два главных «переломных момента», показавших, что веб-документы не всегда будут хуже. Во-первых, выпуск Google Maps (февраль 2005 г.; maps.google.com) показали, что веб-сайты могут быть *приложения*, а не просто статические документы. По моему мнению, это открыло захватывающие возможности для новых форм научно-технической коммуникации: не просто гипертекстовые факсимиле бумаги, но*интерактивный* документы.[1]В более широком смысле я определю*Документы 21 века*как те, которые используют технологии, которые разворачивались с тех пор: документы, разработанные для мира, где интерактивные цифровые дисплеи, вычисления и подключение к интернету часто дешевы и повсеместны. (Я не люблю эту терминологию — попахивает наивным футуризмом — но у меня нет ничего лучше.) В принципе, документы 21-го века могут быть нацелены на любую из множества технологических платформ, но, по моему мнению, Интернет — единственное, что имеет значение. Веб-контент может быть просмотрен практически где угодно, от смартфонов до рекламных щитов, и частная промышленность тратит*миллиарды* долларов каждый год для увеличения своей мощи и охвата. Ничто другое не сравнится с этим.

[1]Посмотрите слайды к моему выступлению, которые находятся в HTML-код,для примера встроенного интерактивного сюжета (tug.org/tug2022/assets/html/Peter_K_G_Williams-TUG2022-slides/). Потому что *Буксир*доставляется вPDFформат, я не могу воспроизвести здесь сюжет.

Но в 2005 году состояние веб-типографики было все еще довольно плохим, и поэтомуPDFпо-прежнему был лучшим выбором для научных работ и других видов*технические документы*. Хотя я не буду пытаться точно определить эту категорию, общие характеристики технических документов включают в себя значительную длину; использование математики, цифр или таблиц; сложную структуру; плотные внутренние или внешние ссылки; и в последнее время интеграцию с исходным кодом и вычислениями. Хотя каждый вид документа заслуживает отличной типографики, я утверждаю, что технические документы, вероятно, страдают*более*от плохой типографики, чем нетехнической. Вторым переломным моментом для меня был выпускpdf.jsбиблиотека для отображенияPDFs в веб-браузере (июль 2011 г.; mozilla.github.io/pdf.js/).Какой лучший способ продемонстрировать, что вы можете создавать высокоточную типографику в Интернете, чем продемонстрировать, что вы можете отображать произвольные PDFфайлы?

После просмотрapdf.js,Я был убежден, что все элементы были готовы к тому, чтобы начать пытаться привнести типографское качество TEX в мир веб-родных документов 21-го века. Даже попытка сделать это, безусловно, потребовала бы создания нового программного обеспечения, но наряду с моей научной подготовкой я занимался разработкой программного обеспечения с открытым исходным кодом еще до поступления в колледж, и я более чем счастлив решать такие проблемы самостоятельно.

Моим первым начинанием, предпринятым где-то в 2014 году, была попытка реализовать в чистой комнате ядро движка TEX на JavaScript, используя*Учебник* в качестве справки. Все прошло так же хорошо, как вы могли ожидать. Я добился приличного прогресса, но быстро понял, что сам движок TEX — это всего лишь вершина пресловутого айсберга кода, необходимого для компиляции настоящего современного LaTEX документы, и что*Учебник*является лишь частичным — фактически, иногда вводящим в заблуждение — руководством по работе современных движков TEX, без обсужденияε-TEX, Unicode, шрифты OpenType и многое другое. Я пришел к выводу, что для компиляции «настоящего» LaДокументы TEX для Интернета, необходимо будет построить на основе «реального» LaTEKC.

Поэтому я начал изучать взлом TEX. А точнее, я изучал модификацию движка X TEX, поскольку он включает поддержку шрифтов Unicode и Open-Type, что показалось мне необходимым для создания по-настоящему веб-родных документов. Как человек с большим опытом в проектах с открытым исходным кодом, этот опыт был откровенно разочаровывающим и обескураживающим. Даже традиционный первый шаг к пониманию кодовой базы — проверка исходного кода из системы контроля версий — казался испытанием, в первую очередь из-за отсутствия ясности относительно того, какой репозиторий использовать, огромного размера и глубоко вложенной структуры репозитория TEX Live, и использование Subversion. Современные удобства разработки программного обеспечения, прежде всего наличие механизма «pull request» (подобного GitHub) и непрерывная интеграция (КИ),пропали без вести.

Хотя в мире разработки ПО нет недостатка в модных тенденциях, были и некоторые реальные достижения, и как разработчик я считаю их чрезвычайно важными. Для меня начинать программный проект без них — это как пытаться написать исследовательскую работу в Microsoft Word, а не в LaTEX. Я могу это сделать, если придется, но мне это не понравится, а худшие инструменты закрывают целые способы работы, от которых я не хочу отказываться. Я не чувствовал, что смогу работать с кодом TEX так, как хотел, если буду вынужден использовать существующую инфраструктуру.

Но если вы возьмете программный проект и перестроите его инфраструктуру разработки, то вряд ли будет возможно объединить ваши изменения обратно в исходный код. Такие изменения приводят к долгоживущему*вилка*, не временный *ветвь*. Форкинг проекта — это весомое решение, к которому нельзя относиться легкомысленно. Но когда я думал об экспериментах, которые хотел попробовать, я пришел к выводу, что форкинг — это подходящий путь. Помимо того, что это позволит мне изучить новые инструменты разработки, это позволит мне изучить новую «персону» для проекта — отличную индивидуальность бренда. Это может звучать как деловой жаргон, потому что это так, но это отражает правильную концепцию. Я хотел иметь возможность попробовать*все виды* вещей, которые вы не могли сделать с традиционным TEX: привести в порядок вывод, изменить поведение по умолчанию, отказаться от совместимости с различными древними пакетами. Было бы неправильно описывать такую систему как обычную систему TEX. Новый брендинг дает возможность сбросить ожидания пользователей сразу, без необходимости объяснять детали отдельных технических изменений.

3 Тектонический проект

В описании предыдущего раздела был выявлен ряд взаимосвязанных пробелов в экосистеме TEX:

- поддержка создания современныхHTMLвывод с помощью полнофункционального движка TEX;
- модернизированный опыт разработчика;
- модернизированный пользовательский опыт; и
- проект с ярко выраженной индивидуальностью бренда, служащий платформой для экспериментов.

Проект Tectonic, запущенный в 2016 году, призван заполнить эти пробелы. Ключевые элементы его дизайна следующие.

3.1 Форм-фактор

Tectonic поставляется в виде одного исполняемого файла под названием тектонический,который объединяет возможности X TEX, бибтекс, xdvipdfmx,и вспомогательные машины для вождения

эти*двигатели*. Исполняемый файл разработан так, чтобы быть максимально автономным, с минимальными зависимостями от системных библиотек, переменных среды, файлов конфигурации пользователя или внешних инструментов. В частности, зависимости от Ghostscript были удалены в целях безопасности, что исключает возможности PostScript.

3.2 Реализация движка

Двигатели, особенно X TEX, реализованы с использованием C/C++код получен из стандартаВЕБКонвейер 2C, реализованный TEX Live. C/C++Файлы, извлеченные из конвейера, были значительно переформатированы и рефакторированы, чтобы сделать их более удобными для чтения человеком и, например, повторно ввести символические константы, которые не сохраняютсяВЕБ Рабочий процесс 2C. Несколько рефакторингов были проведены автоматически скокцинелла [2]. Однако из-за этих настроек обновления движка из TEX Live не могут быть автоматически включены в кодовую базу Tectonic. Это цена разветвления. Чтобы помочь процессу синхронизации Tectonic с TEX Live, фреймворк называетсятектоническая стадийность (репозиторий кода hagithub.com/tectonic-typesetting/tectonic-staging/) содержит конвейер, который может автоматически генерировать читаемый набор C/C++« «Источники ссылок» из репозитория TEX Live. Когда обновления из нового выпуска TEX Live должны быть включены в Tectonic, конвейер запускается, и изменения в источниках ссылок вручную импортируются в кодовую базу Tectonic. Эта система в значительной степени использует функции отслеживания измененийmerзавецсистема контроля версий.

3.3 Использование ржавчины

За исключением движков, Tectonic реализован на языке Rust. Rust — это язык системного уровня, ориентированный на производительность, надежность и продуктивность. Модель упаковки и компиляции Rust отлично подходит для такого проекта, как Tectonic: в то время как Rust предлагает сложную экосистему пакетов, которая позволяет легко импортировать поддержку для чего угодно изHTTPS протокол загрузки образа, он компилируется по умолчанию в самодостаточные исполняемые файлы, которые не имеют внешних зависимостей. Rust также имеет отличную поддержку для кроссплатформенной работы и моста с C и C++Код. Инструмент для упаковки Rust,груз,позволяет организовать кодовые базы в «ящики» с четко определенными интерфейсами и имеет систему «функций» для управления параметрами сборки. Основная кодовая база Tectonic в настоящее время состоит из 22 ящиков.

В более широком смысле, язык Rust имеет схожий дух с TEX. Оба считаются лучшими в своем классе инструментами, которые могут быть требовательными, но также и полезными. Оба имеют репутацию сложных и трудных для изучения. Несмотря на эту репутацию, Rust имеет достиг огромного успеха за относительно короткий период времени, будучи названным «самым любимым языком» stackexchange.comсемь лет подряд на момент написания этой статьи. Сторонники Rust обычно приписывают этот успех нескольким факторам. Во-первых, Rust технически превосходен: он действительно выполняет свои обещания неукоснительно, а сторонние пакеты Rust часто хорошо спроектированы, производительны и надежны. Во-вторых, Rust имеет превосходный инструментарий с высококачественной встроенной поддержкой управления пакетами (груз), документация, тестирование и многое другое. В-третьих, сообщество пользователей Rust явно ценит гостеприимство и инклюзивность. Многие аспекты дизайна Rust направлены на поддержку новых пользователей, наиболее известные из которых — сообщения об ошибках компилятора Rust, которые обычно предлагают впечатляюще четкую диагностику проблем и полезные советы по их устранению. Охват этих факторов — тема*дизайн, ориентированный на опыт*: элементы экосистемы Rust разрабатываются в первую очередь вокруг видения того, как люди будут их использовать, с техническими целями, вытекающими из этого видения. Tectonic явно стремится подражать этим характеристикам экосистемы и сообщества Rust.

3.4 Связки

Tectonic может быть доставлен как единый исполняемый файл, поскольку он может загружать файлы из резервного дистрибутива TEX на лету, во время компиляции документа. Эта функциональность реализуется путем виртуализации Ввод/выводПодсистема, лежащая в основе движков, так что она может искать файлы не только в локальной файловой системе, но и в удаленных «связках». Файлы из связок кэшируются локально, а реализация разработана таким образом, что сеть нужна только в случае, если необходимо загрузить новый файл. Связки создаются с использованием воспроизводимого автоматизированного конвейера на основе процесса установки TEX Live (github.com/tectonic-typesetting/tectonic-texlive-bundles/).Файл пакета, передаваемый по сети, по сути является большим файлом Unixсмола файл с соответствующим индексом, который тектонический программа загружается по частям с помощью HTTPSЗапросы диапазона байтов.

Схема пакета также является основой подхода Tectonic к воспроизводимым сборкам документов. Файлы пакета должны быть неизменяемыми, и можно связать заданный документ Tectonic (см. ниже) с определенным пакетом, идентифицированным по его URL илиSHA256 Криптографический дайджест на основе его содержимого. Таким образом, можно указать точное распределение TEX, на основе которого должен быть построен документ. С этим связана потеря гибкости: чтобы обновить или расширить пакет, содержащийся в пакете, необходимо сгенерировать собственный пакет или установить файлы пакета локально, в настоящее время на основе каждого документа.

3.5 Модель документа

Tectonic предлагает «модель документа» для определения компиляций. Его дизайн во многом обязан дизайну Rustгруз Инструмент. Тектонические документы — это структуры каталогов, обозначенные существованием файла с именем Тектоника.томлв корне. Этот файл, в TOMLформат структурированных данных (toml.io),объявляет основные метаданные о документе и о том, как он должен быть построен. По умолчанию исходный код документа верхнего уровня содержится в подкаталоге с именемисточникв файлах с именамипреамбула.tex, индекс.tex,ипостамбула.текс, которые обрабатываются в указанном порядке.

Документ можно создать, выполнив команду тектоническая -X-складкав любом месте исходного дерева. Это создаст одну или несколько версий документа встроить каталог нижеТектоника.томл Файл. -XФлаг отмечает использование интерфейса командной строки Tectonic «версии 2», который использует парадигму «подкоманды» или «мультиинструмента», напримермерзавецилисвн. Для совместимости режим работы по умолчанию по-прежнему «версия 1»:тектонический myfile.texскомпилирует указанный входной файл без вызова модели документа. Эта форма одноразовой компиляции доступна в интерфейсе версии 2 ctectonic -X компилирует myfile.tex.Интерфейс версии 1 в конечном итоге будет объявлен устаревшим, а -X флаг станет необязательным.

Основная цель модели документа Tectonic — сделать сборки документов автоматизированными, воспроизводимыми и анализируемыми, представляя специфичные для документа варианты в виде конфигурации, а не (например) строки параметров командной строки. Например, Тектоника.томлfile может записывать, какой файл формата TEX требуется для сборки или нужна ли ей функциональность shell-escape. (Флаг времени выполнения может переопределить эту настройку, если документ для сборки не из надежного источника.) Как упоминалось выше, спецификация может определять несколько выходных данных для одного документа, напримерPDFв обоих НАСПисьмо иA4размеры.

Хотя модель документа еще не была тщательно разработана в Tectonic, ожидается, что она предоставит платформу для дополнительных утилит в будущем. Например, будущаятектонический -X форматкоманда может автоматически переформатировать исходные тексты документа в едином стиле илитектонический -X док может генерировать метадокументацию о доступных последовательностях управления, настроенных для конкретного набора пакетов документа.

4HTMLвыход

Хотя и высокого качестваHTMLвыход был целью проекта Tectonic с самого начала, мало что было сделано на этом фронте — до этого года. Интересный прогресс начал происходить.

Общий подход, принятый при внедрении HTMLвыход для Tectonic был сосредоточен на достижении высококачественных результатов с документами, которые специально нацелены на этот выходной формат. В то время как конечная цель состоит в том, чтобы иметь возможность производить хорошиеHTML вывод из произвольных входных документов, что является более крупной проблемой, которую пытаются обойти в настоящее время. Текущие усилия также отдают приоритет визуальному виду над правильной семантической маркировкой и фокусируются на английском языке.

В широком смысле, когдаHTMLтребуется вывод, активируется специальный флаг в движке X TEX, который изменяет различные аспекты его поведения. Разрывы строк в абзацах отключены, чтобы избежать работы с переносами, и \особенныйs вставлены для указания предлагаемых двигателем мест для вставкиHTMLтеги, такие как <p>.Полученный выходной файл по сути находится в X TEXXDVформат, но он переименован в новый «СПХ"формат во избежание путаницы. (СПХ) означает «семантически-пагинированный»XDV",(Но это неправильное название, поскольку семантическая пагинация оказалась технически неосуществимой.)

Новый этап обработки, написанный на Rust,spx2html, использует шаблонизатор Tera (tera.netlify. приложение) чтобы преобразовать синглSPXфайл в один или несколько HTMLвыводит и создает или копирует связанные файлы, такие какCSSтаблицы стилей, код пользовательского интерфейса JavaScript и файлы шрифтов.spx2htmlэтап разработан с учетом того, что входной документ использует шрифты OpenType везде, включая математику, через юникод-математикаpackage. Это значительно сокращает проблемное пространство, позволяя коду работать только со шрифтами, которые могут быть отображены напрямую браузером.

4.1 Точная типографика на холстах

ИсходныйHTMLработа была сосредоточена на демонстрации точного размера и позиционирования символов, необходимых для визуализации конструкций, таких как «TEX». В Tectonic основная часть текста документа выводится непосредственно в HTML-код,но области, требующие тщательной типографской разметки, обрабатываются специально, как*холсты*. Макет в режиме холста может быть активирован автоматически движком (например, в математическом режиме) или вручную автором (с помощью \особенныйc).

Потому чтоCSSкоманды могут быть использованы для перемещения отдельныхHTMLэлементы произвольно, фактическое позиционирование несложно, хотя необходимо соблюдать осторожность, чтобы добиться правильного выравнивания относительно базовой линии текста для встроенных выражений. Более сложным является тот факт, чтоSPXфайл определяет, как *глифы* в шрифте следует размещать, в то время какHTML вывод должен быть текстом Unicode — и это разные концепции. Во многих случаях есть прямое отображение

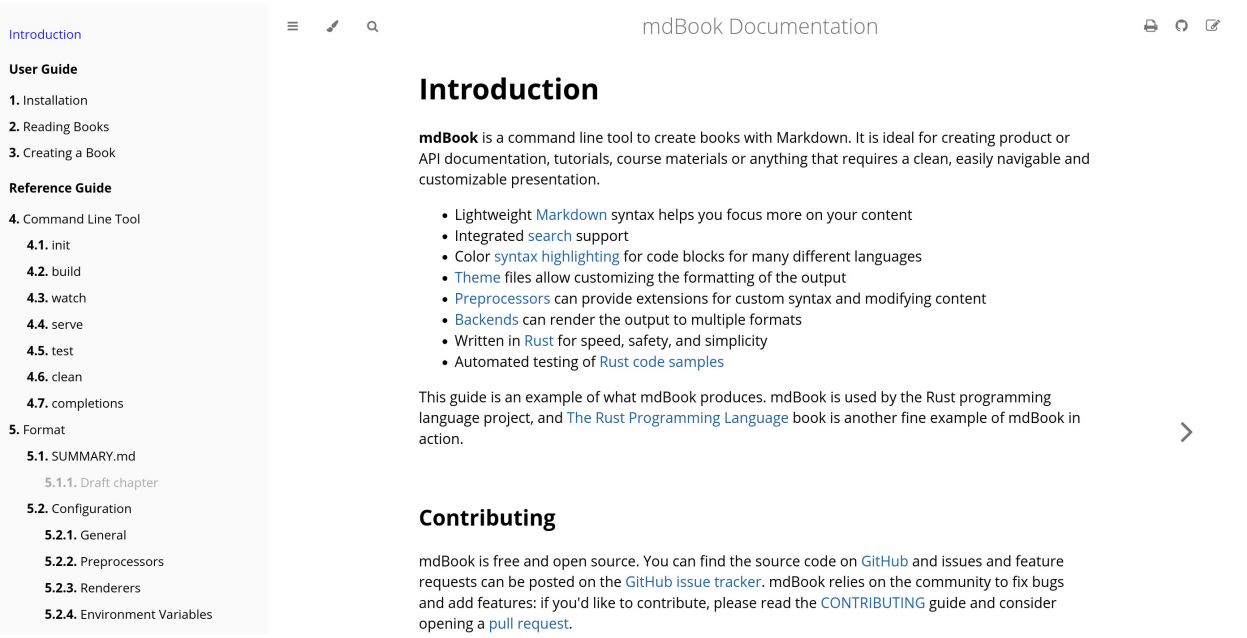

Рисунок 2:СтандартmdBookмакет, обсуждаемый в подразделе 4.2. Изrust-lang.github.io/mdBook/.

между ними, закодированным в Unicode шрифтаKMAП таблица; но не всегда. Например, среда отображения математики может потребовать большую версию знака интеграла, который не может быть «достигнут» путем испусканияU+222B ИНТЕГРАЛсимвол, который соответствует уменьшенной версии глифа.

Когда Tectonic сталкивается с этой проблемой, он решает ее, создавая дополнительную версию соответствующего файла шрифта с настроеннымKMAПстол. Это *вариант глифа*подход представляет собой слегка обобщенную форму поднабора шрифтов, хотя реализация Tectonic гораздо более наивна, чем «настоящий» поднабор шрифтов. В приведенном выше примере новый шрифтKMAП таблица может заменить отображениеU+222B ИНТЕГРАЛ от интегрального глифа по умолчанию до большой версии, необходимой для рассматриваемой математики отображения.HTML для этого холста будет затем включать <диапазон>тег, стилизованный для загрузки этого шрифта, имеющий соответствующий размер и расположение, содержащий одинU+222B ИНТЕГРАЛхарактер.

Чтобы определить, нужно ли создавать новый вариант шрифта, Tectonic должен проанализировать и инвертироватьKMAПтаблицы шрифтов, используемых в обрабатываемом документе. Этот процесс потенциально уязвим, поскольку в полной общности он по сути требует инвертирования алгоритмов «формирования» символов. Обратите внимание, однако, что это необходимо только для символов, которые встречаются в холстах. Для основного текста документа, почти во всех случаях Tectonic может выдавать вывод Unicode непосредственно из АктуальныйТекст информация, выдаваемая движком X TEX.

4.2 Хром:HTML, CSS,JavaScript

Подход Tectonic направлен на минимизацию объема веб-дизайна, происходящего на уровне TEX. Вместо этого, HTMLКонтент, полученный из ввода TEX, вставляется в предварительно разработанные шаблоны. Важно подчеркнуть, что в современном веб-дизайне такие шаблоны неизбежно состоят из взаимозависимых частейHTML, CSS,и код JavaScript. Эти части объединяются, чтобы сформировать*хром*итогового веб-документа. Chrome охватывает все: от высокоуровневой разметки страницы до интерактивных функций, таких как поиск, скрываемые боковые панели и нелинейная навигация. Высококачественный хром по умолчанию является важнейшим компонентом конвейера производства веб-документов.

Текущие усилия сосредоточены на чистом дизайне, имитирующем дизайн инструментамdBook (rust-lang.github.io/mdBook/),веб-система документации на основе Markdown. Скриншот по умолчаниюmdBook Макет показан на рисунке 2. На большом экране вид по умолчанию разделен на область основного контента и боковую панель. Основной текст располагается по центру в области основного контента с максимальной шириной, чтобы длина строк не становилась чрезмерной. Ненавязчивая строка заголовка прилипает к верхней части страницы, но автоматически скрывается, когда читатель прокручивает основной контент. На мобильных дисплеях боковая панель остается скрытой по умолчанию и может быть открыта с помощью «гамбургер-меню» строки заголовка.

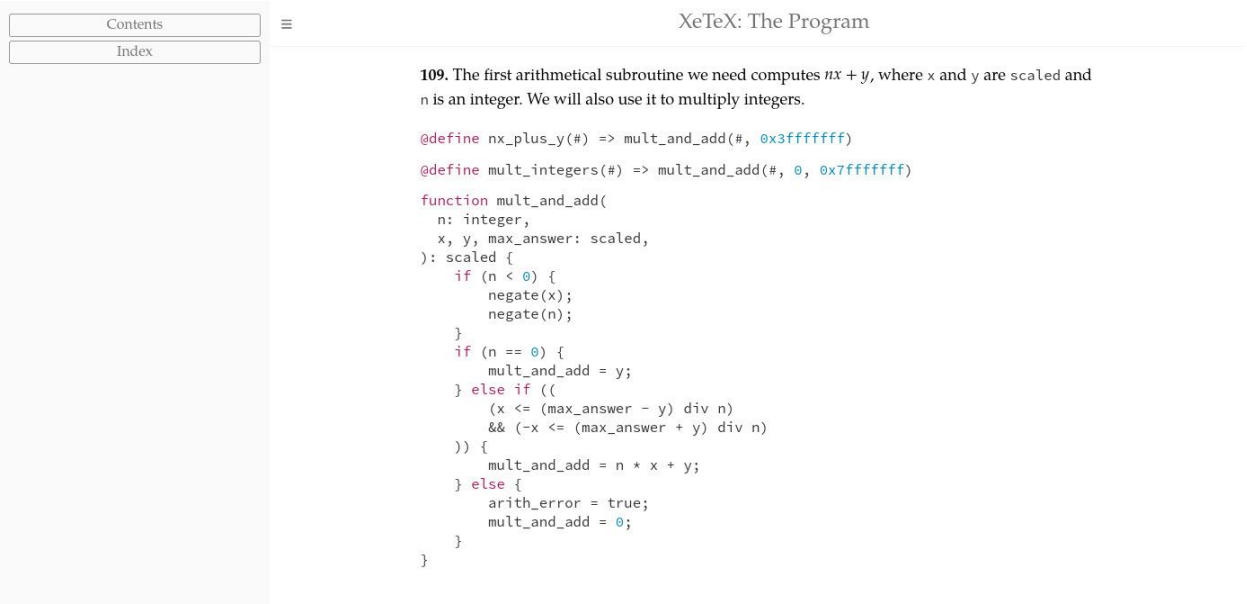

Рисунок 3:Снимок в стадии разработкиtt-ткатьпрезентация*X TEX: Программа*, с очевидными долгамиmdBook (Рисунок 2).

После долгих исследований я пришел к выводу, что эта схема вполне может быть*the* Оптимальный дизайн для представления технических документов общего назначения в Интернете. На широких экранах размещение основного текста становится неудобным, если он не центрирован в окне браузера. Если максимальная ширина текста не ограничена, строки становятся слишком длинными, как это видно в Википедии. На мобильных устройствах места достаточно мало, чтобы на экране не было практически ничего, кроме основного текста во время чтения — ограничение, которое хорошо компенсируется комбинацией липкой, автоматически скрывающейся строки заголовка и переключаемой боковой панели.

Наконец, многие проекты Chrome пытаются втиснуть информацию в несколько боковых панелей, заголовков и нижних колонтитулов. Они загромождают страницу и их трудно использовать на мобильных устройствах. Лучшей альтернативой является предоставление таких дополнительных элементов в виде «модальных окон», наложений, которые можно быстро вызывать и закрывать с помощью значков в строке заголовка (на всех платформах) или нажатий клавиш (на настольных компьютерах). Это пример того, как Chrome состоит не только из макета страницы: документы 21-го века могут иметь полноценные пользовательские интерфейсы.

4.3tt-ткать

Текущая работа над TectonicHTMLвывод фокусируется на очень конкретном тестовом случае:*X TEX: Программа*, вариант*TEKC: Программа*[1] произведено из пропатченного X TEX'sВЕБкод. Он близок к идеалу, поскольку он длинный, хорошо структурированный, имеет много перекрестных ссылок, часто используется автором и доступен в виде исходного текста TEX.

Исходный текст TEX, сгенерированный традиционным способом ткатьПрограмма очень хорошо настроена на вывод на печать. Хотя я мог бы потенциально поработать над созданием HTMLвыход изткать-сгенерированный TEX-код, у меня была дополнительная цель. С сожалением должен сказать, что я всегда находил листинги кода, сгенерированныеткать чрезвычайно трудно читать, хотя я знаю, что в их дизайн вложено много внимания. Я хотел посмотреть, смогу ли я создатьткать-как инструмент, который может переформатировать X TEX ВЕБкод в моноширинный формат с цветным синтаксисом, который мне более знаком.

Результатом этой работы стал инструмент Rust под названием tt-ткать (github.com/pkgw/tt-weave).По сути, он выполняет ту же функцию, что иткать,но анализирует части PascalВЕБкод с высоким уровнем семантической осведомленности и выдает их как блоки специализированного кода TEX в отступе, моноширинном формате со встроенными командами, управляющими раскрашиванием синтаксиса и взаимосвязью. Синтаксис выданного кода переписывается, чтобы внешне напоминать C и Rust. Например, логическое «и» представлено с помощью &&, а не∧. Однако никаких семантических преобразований не предпринимается. Индексная информация выдается вJSONФайлы данных, которые могут быть использованы веб-браузером Chrome.tt-ткать программа не предназначена для использования в качестве универсального приложенияВЕБпроцессор, и содержит многочисленные хаки, характерные для пропатченного xetex.webВходной файл. Снимок его выходных данных показан на рисунке 3. Показанный здесь дизайн обновлен относительно

версия включена в комплектHTMLслайды, связанные с этой презентацией.

На момент написания статьи соответствующий хром находится в стадии разработки. Весь текст книги может быть отображен в виде одногоHTMLфайл (~10МБ) который на самом деле удобно использовать в браузере, и онлайн-слайды, связанные с этой статьей, включают снимок этой формы вывода. Такая большая страница является непрактичным механизмом доставки для общего использования, однако, и ведется работа по подразделению вывода для динамической загрузки. Это также поможет с интеграцией вывода Tectonic в стандартные отраслевые фреймворки веб-разработки (например,npm, вебпак), что значительно повысило бы производительность разработки, сделав удобным внедрение таких технологий, какCACC (sass-lang.com)или Type-Script (typescriptlang.org).

5 Перспективы

Проект «Тектоника» до сих пор был успешным, собрав ~2800 «звезд» на GitHub и регистрация 47 отдельных участников проекта на момент написания этой статьи. Хотя еще многое предстоит сделать, чтобы сделатьHTMLвыходная структура, пригодная для общего использования, метод вариантных глифов успешно решает наиболее технически сложную проблему в текущей системе.

Документация по проекту «Тектоника» — как это ни парадоксально, учитывая его предмет и устремления — отсутствует.tt-ткатьусилия продемонстрировали хороший успех с существующим*X TEX: Программа*в книге предполагается начать создание новой документации для исправления этой ситуации.

До сих пор главным человеком, руководящим работой над Tectonic, был автор этой статьи. Однако с самого начала проекта была надежда сделать его гостеприимным местом для новых участников, и по мере развития проекта это становится важнее, чем когда-либо. Есть*многочисленные*области — нелатинские шрифты, доступность, не-LaTEX workflows, TEX internals, где больше опыта со всего мира TEX было бы очень полезно. Людям, заинтересованным во взаимодействии с сообществом Tectonic, следует посетить форум обсуждений Tectonic, прикрепленный к его репозиторию GitHub по адресуgithub.com/tectonic-typesetting/tectonic/discussions.

Конечно, Tectonic существует только потому, что он основывается на работе сотен, если не тысяч, людей, которые сотрудничали, чтобы построить экосистему TEX за последние несколько десятилетий. Хотя Tectonic позиционирует себя как «вне» традиционного TEX в определенном смысле, искреннее намерение состоит в том, чтобы отдать должное и отметить работу всех этих людей как можно полнее. С огромной благодарностью я благодарю вас за то, что вы поделились своим замечательным творением с миром.

Ссылки

[1] Д. Э. Кнут.*TEKC: Программа*, том Б *Компьютеры и набор текста*. Эддисон-Уэсли, Реддинг, Массачусетс, США, 1986.

[2] Дж. Лоуэлл, Г. Мюллер. Coccinelle: 10 лет автоматизированной эволюции ядра Linux. В *Ежегодная техническая конференция USENIX 2018 (USENIX ATC 18)*, стр. 601–614, Бостон, Массачусетс, июль 2018 г. Ассоциация USENIX.www.usenix.org/conference/atc18/presentation/lawall

• Питер К.Г. Уильямс 60 Гарден Стрит. MS-20 Кембридж, Массачусетс 02138
США
pwilliams (at) cfa точка harvard точка edu
https://newton.cx/~peter/
ОРКИД0000-0003-3734-3587