

최신 컴퓨팅 환경의 T_EX: 엔진, 자동화, 인공지능

Tectonic + GitHub Actions + LLM

이재호

2023년 6월 3일

목차

엔진: Tectonic

자동화: GitHub Actions

인공지능: ChatGPT/GitHub Copilot

엔진: Tectonic

T_EX 생태계



Tectonic은...

- $X_{\text{T}}\text{E}_{\text{X}}$ 을 기반으로 한 $\text{T}_{\text{E}}\text{X}$ 엔진
 - ▶ 2016년에 $X_{\text{T}}\text{E}_{\text{X}}$ 에서 포크
 - ▶ 모던 폰트와 유니코드를 지원
- $\text{T}_{\text{E}}\text{X}$ Live 배포판 사용
 - ▶ 패키지들을 필요에 따라 다운로드

Tectonic은...

- X₃T_EX을 기반으로 한 T_EX 엔진
 - ▶ 2016년에 X₃T_EX에서 포크
 - ▶ 모던 폰트와 유니코드를 지원
- T_EX Live 배포판 사용
 - ▶ 패키지들을 필요에 따라 다운로드
- 러스트(Rust)로 작성
 - ▶ 메모리 안전성을 보장하는 프로그래밍 언어

Tectonic은...

- X₃TeX을 기반으로 한 TeX 엔진
 - ▶ 2016년에 X₃TeX에서 포크
 - ▶ 모던 폰트와 유니코드를 지원
- TeX Live 배포판 사용
 - ▶ 패키지들을 필요에 따라 다운로드
- 러스트(Rust)로 작성
 - ▶ 메모리 안전성을 보장하는 프로그래밍 언어
- 멋진 이름: τέκτων (tektōn)
 - ▶ 목수
- 거인의 어깨 위에
 - ▶ TeX, L^ATeX, X₃TeX, xdvipdfmx

러스트?

- 2010년 Mozilla에서 개발 시작
 - ▶ 타입 수준에서 메모리 안전성을 보장
 - ▶ C/C++과 (거의) 동일한 성능

러스트?

- 2010년 Mozilla에서 개발 시작
 - ▶ 타입 수준에서 메모리 안전성을 보장
 - ▶ C/C++과 (거의) 동일한 성능
- X₃T_EX: WEB2C로 생성된 C 코드를 재구성
 - ▶ Tectonic은 이를 Rust로 다시 작성해나가는 중
- 러스트가 지원하는 웹어셈블리(WASM)도 지원할 수 있는 가능성

러스트?

- 2010년 Mozilla에서 개발 시작
 - ▶ 타입 수준에서 메모리 안전성을 보장
 - ▶ C/C++과 (거의) 동일한 성능
- X₃TEX: WEB2C로 생성된 C 코드를 재구성
 - ▶ Tectonic은 이를 Rust로 다시 작성해나가는 중
- 러스트가 지원하는 웹어셈블리(WASM)도 지원할 수 있는 가능성
- 러스트 crate(라이브러리)로서, 다른 프로그램에 임베딩 가능
- 러스트의 훌륭한 CLI와 프로젝트 구성을 계승
 - ▶ cargo 스타일의 tectonic CLI 명령어
 - ▶ 기본적으로 텍 소스와 조판된 문서를 분리하고, 텍 소스도 프리앰블과 본문 등으로 분리

러스트의 친절한 CLI

```
>> cargo build
Compiling lamb v0.1.0 (/Users/jay/Developer/RustProjects/Lamb)
error[E0277]: a value of type 'HashSet<str>' cannot be built from an iterator over elements of type '&str'
   --> src/lib.rs:25:34
    |
25 |     Expr::Lamb(x, e_body) => free_var(e_body)
    |                                ^
26 |         .difference(&HashSet::from([x.as_str()])))
    |         ~~~~~ value of type 'HashSet<str>' cannot be built from 'std::iter::Iterator<Item=&str>'
27 |         .collect(),
    |         ----- required by a bound introduced by this call
    |
   = help: the trait 'FromIterator<&str>' is not implemented for 'HashSet<str>'
   = help: the trait 'FromIterator<T>' is implemented for 'HashSet<T, S>'
note: required by a bound in 'collect'
   --> /Users/jay/.rustup/toolchains/stable-aarch64-apple-darwin/lib/rustlib/src/rust/library/core/src/iter/traits/iterator.rs:1832:19
1832 |         fn collect<B>: FromIterator<Self::Item>=>(self) -> B
    |         ~~~~~ required by this bound in 'collect'

For more information about this error, try 'rustc --explain E0277'.
error: could not compile 'lamb' due to previous error
```

```
DBA90BB382A7A234.pygtx default.pyg`
note: running shell command: 'rm default.pyg`
Rerunning TeX because "default.toc" changed ...
warning: lineno.sty:296: Invalid UTF-8 byte or sequence at line 296 replaced by U+FFFD.
warning: open of input /dev/null.tex failed
caused by: access to the path '/dev/null.tex' is forbidden
warning: open of input /dev/null.tex failed
caused by: access to the path '/dev/null.tex' is forbidden
note: running shell command: 'uname -s > "default.w18"'
note: running shell command: 'rm -- "default.w18"'
note: running shell command: 'mkdir -p _minted-default'
note: running shell command: 'which pygmentize && touch default.aex`
/Users/jay/.pyenv/shims/pygmentize
note: running shell command: 'rm default.aex`
warning: default:2: Overfull \vbox (15.63992pt too high) detected at line 2

warning: default:125: Overfull \vbox (41.9229pt too high) detected at line 125

warning: default:134: Overfull \vbox (41.9229pt too high) detected at line 134

warning: default:152: Overfull \vbox (2.52692pt too high) detected at line 152

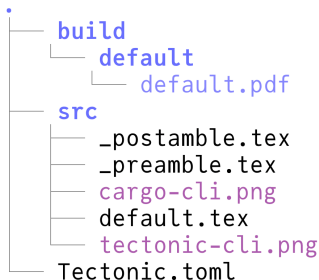
note: running shell command: 'rm default.pyg`
warning: warnings were issued by the TeX engine; use --print and/or --keep-logs for details.
Running xdvipdfmx ...
warning: open of input /Users/jay/Library/Fonts/KoPubWorld Dotum_Pro Bold.otf/rsrc failed
caused by: access to the path '/Users/jay/Library/Fonts/KoPubWorld Dotum_Pro Bold.otf/rsrc' is forbidden
warning: open of input /Users/jay/Library/Fonts/KoPubWorld Dotum_Pro Bold.otf:0-0-GID failed
caused by: access to the path '/Users/jay/Library/Fonts/KoPubWorld Dotum_Pro Bold.otf:0-0-GID' is forbidden
warning: open of input /Users/jay/Library/Fonts/FiraSans-Regular.otf/rsrc failed
```

사용법

tectonic -X로 V2 인터페이스 사용

- `tectonic -X new hello-tectonic`으로 새 프로젝트 생성
- `cd hello-tectonic` 후
- `tectonic -X build`로 문서 조판
 - ▶ 변경 사항이 없으면 조판하지 않음

예시:



설정 파일

Tectonic.toml (TOML 포맷)

```
[doc]
name = <string>
bundle = <url or filesystem path>
```

```
[[output]]
name = <string>
type = <"pdf">
tex_format = [string]
shell_escape = [bool]
preamble = [string]
index = [string]
postamble = [string]
```

Visual Studio Code 연동 i

LaTeX Workshop 확장과 함께 사용

- LaTeX Workshop 확장을 설치
- `ctrl` + `⇧` + `p` (macOS: `⌘` + `⇧` + `p`)로 명령 팔레트 열어서
Preferences: Open User Settings (JSON) 입력

Visual Studio Code 연동 ii

- settings.json 파일에 다음과 같이 추가:

```
"latex-workshop.latex.recipe.default": "tectonic",
"latex-workshop.latex.autoBuild.run": "onSave",
"latex-workshop.latex.outDir": "%WORKSPACE_FOLDER%/build/default",
"latex-workshop.view.pdf.viewer": "tab",
"latex-workshop.latex.recipes": [
  {
    "name": "tectonic",
    "tools": ["tectonic"]
  }
],
"latex-workshop.latex.tools": [
  {
    "name": "tectonic",
    "command": "tectonic",
    "args": ["-X", "build"],
    "env": {}
  }
]
```

Visual Studio Code 연동 iii

- Tectonic.toml 파일에 다음과 같이 변경:

```
[[output]]  
name = 'default'  
index = 'default'
```

- default.tex 첫 줄에 다음과 같이 추가:
% !TEX root = ./default.tex

참고: <https://github.com/tectonic-typesetting/tectonic/discussions/896#discussioncomment-6073309>

임베딩

```
use tectonic;
let latex = r#"
\documentclass{article}
\begin{document}
Hello, world!
\end{document}
"#;
let pdf_data: Vec<u8> = tectonic::latex_to_pdf(latex)
    .expect("processing failed");
println!("Output PDF size is {} bytes", pdf_data.len());
```

이외에도 `tectonic::XdvipdfmxEngine` 등의 엔진을 사용할 수 있음¹

¹<https://docs.rs/tectonic/latest/tectonic/>

참재력

XeTeX: A Pseudoprogram

Preamble. This digital book describes the XeTeX typesetting program, which converts input TeX source code to the XDV binary format.

About XeTeX

XeTeX is a modified version of Donald E. Knuth's TeX typesetting program that adds support for Unicode text and OpenType fonts, among other changes. It is the foundation of the [Tectonic project](#), a modernized TeX system, which is built around a modified clone of the XeTeX engine.

About this book

The original TeX system is written in WEB, a literate programming language — one that combines code and documentation. WEB code can be turned into either an executable program (a process called “tangling”) or into a book combining the formatted source code and extensive documentation (“weaving”). The TeX program — viewed holistically as both code and documentation — is the work of Donald E. Knuth, and is renowned for its staggering attention to detail, comprehensiveness, and polish. The associated book is published in hardcopy as [Computers & Typesetting, Volume B — TeX: The Program](#).

This book is “woven” from the WEB code underlying XeTeX, which is expressed as a series of patches to Knuth’s TeX file. It is the work of many authors building atop Knuth’s foundation. Unlike TeX it is not the product of a singular vision, and even the best of us struggle to match the standard of exactitude set by Knuth. Abrupt changes in style will be apparent if you read this book linearly. While much of the book’s text will say that it is describing TeX, it is always and only describing XeTeX.

However, the full XeTeX program consists of not only WEB code, but also extensions written in C and C++. Those extensions are not included in this book. Some of the WEB code can only be understood in the context of its interaction with the absent extension code. Furthermore, XeTeX itself must be combined with other programs and data files to do anything useful with a modern TeX document. Those ecosystem-level aspects are not addressed here either.

Furthermore, this book is derived from the WEB code through a program called [tt-weave](#), which transforms the WEB file into TeX code that is then compiled into a [Vue.js](#) web application using [Tectonic](#) and [Parcel](#). The tt-weave program emits the code component of the

한계

- 인터넷 연결이 없다면 패키지를 다운로드할 수 없음
- Tectonic 프로젝트가 마지막으로 업데이트한 $\text{T}_{\text{E}}\text{X}$ Live 버전에 의존
 - ▶ 최신 버전의 패키지를 직접 CTAN에서 받아 사용해야 함
 - ▶ src 디렉터리 (`tectonic -X build -Z search-path=<path>`)
 - ▶ 오버리프(Overleaf)를 사용하는 경우와 유사

참고: ksminitex

정보글 | ksminitex: 아마도 아주 작은 텍 시스템

2021.12.22 16:49

noname

조회 수:1127

다시 텍을 쓸 가능성이 거의 없는 분들을 대상으로 텍 맛보기 강작을 하게 되었다고 할 때, overleaf에 의존하는 방법도 있겠지만 그건 그것대로 번거로운데요, texlive 설치를 요구하는 것은 무리, 요즘 주목받는(?) tinytex이란 것도 있지만 설치 후 설정 등을 아주 무시할 수는 없지요. miktex도 만만치 않을 뿐더러 특히 jiwonlipsum을 쓸 수 없다는 것 때문에 고민하다가, 이런 걸 만들었습니다.

[tectonic](#)을 이용하는 건데요, 약간의 설정을 덧붙이고 유틸리티도 몇 개 추가하고, 사실저장소 패키지를 넣고, 심지어 함초롬 LVT까지 포함하였는데 90메가 정도밖에 안 됩니다.

<https://bit.ly/3qbR5h4>

사용법과 주의사항은 README 폴더 안에 있는 readme.pdf를 읽어보시면 되구요. 웬만한 LaTeX 작업은 거의 되는 걸로... 더 좋은 건, 그냥 uninstall할 수 있다는 것입니다. 잠깐 쓰고 버리자는 목적이라고나 할까... 혹시 다른 컴퓨터에서 잠깐 TeX을 써야 할 때 도움이 될지도 모릅니다.

반응이 좋으면 유지 보수할 생각도 있습니다.

자동화: GitHub Actions

지속적 통합 및 배포 (CI/CD)

- 지속적 통합(Continuous Integration, CI): 코드 저장소에 변경 사항이 있을 때 자동으로 빌드 및 테스트
 - ▶ 코드의 오류를 빠르게 발견하고, 코드 저장소의 코드 상태가 항상 정상인지 확인
- 지속적 배포(Continuous Deployment, CD): 지속적 통합을 통해 빌드 및 테스트를 통과한 코드를 자동으로 배포
 - ▶ 보다 빠르게 세상에 새로운 기능을 제공

GitHub Actions

- GitHub에서 제공하는 CI/CD 서비스
- GitHub 저장소에 특정 이벤트가 발생하면 자동으로 작업 수행
- 공개된 여러 액션들을 조합하거나 직접 액션을 만들어 원하는 작업을 수행

Tectonic ♥ GitHub Actions

TeX Live 도커 이미지를 사용하는 통상의 GitHub Action은 느릴 수 밖에 없지만, Tectonic은 필요한 패키지만 받아서 실행하기 때문에 훨씬 가볍고 빠름

<https://github.com/xu-cheng/latex-action/issues/13>

데모: GitHub에 푸시하면 문서를 조판

./github/workflows/tectonic.yml

```
name: 'Build LaTeX Document'
on:
  push:
jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout
        uses: actions/checkout@v3
      - uses: wtfjoke/setup-tectonic@v2
        with:
          github-token: ${{ secrets.GITHUB_TOKEN }}
      - name: Run Tectonic
        run: tectonic -X build
      - name: Upload pdf
        uses: actions/upload-artifact@v3
        with:
          name: index
          path: build/default/default.pdf
```

인공지능: ChatGPT/GitHub Copilot

글글글

일반적으로 글을 쓰는 것은 고된 노동

글글글

일반적으로 글을 쓰는 것은 고된 노동

- \LaTeX 으로 문서를 작성하는 더욱 큰 노동

- ▶ 단순히 내용이 아니라 모양에 대해서도 신경 써야 함
- ▶ 보기 좋은 떡이 먹기도 좋다 ↔ 빛 좋은 개살구

글글글

일반적으로 글을 쓰는 것은 고된 노동

- \LaTeX 으로 문서를 작성하는 더욱 큰 노동

- ▶ 단순히 내용이 아니라 모양에 대해서도 신경 써야 함
- ▶ 보기 좋은 떡이 먹기도 좋다 ↔ 빛 좋은 개살구

- 긴 매크로들을 입력하기 위해 필요한 손재주?

- ▶ 단축기 숙달 (Vim-fu, Emacs-fu, VSCode-fu(?!), ...)
- ▶ 스니펫 활용
- ▶ \TeX 매크로 정의

글글글

일반적으로 글을 쓰는 것은 고된 노동

■ \LaTeX 으로 문서를 작성하는 더욱 큰 노동

- ▶ 단순히 내용이 아니라 모양에 대해서도 신경 써야 함
- ▶ 보기 좋은 떡이 먹기도 좋다 ↔ 빛 좋은 개살구

■ 긴 매크로들을 입력하기 위해 필요한 손재주?

- ▶ 단축기 숙달 (Vim-fu, Emacs-fu, VSCode-fu(?!), ...)
- ▶ 스니펫 활용
- ▶ \TeX 매크로 정의
- ▶ 조수에게 도움 받기

글글글

일반적으로 글을 쓰는 것은 고된 노동

■ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 으로 문서를 작성하는 더욱 큰 노동

- ▶ 단순히 내용이 아니라 모양에 대해서도 신경 써야 함
- ▶ 보기 좋은 떡이 먹기도 좋다 ↔ 빛 좋은 개살구

■ 긴 매크로들을 입력하기 위해 필요한 손재주?

- ▶ 단축기 숙달 (Vim-fu, Emacs-fu, VSCode-fu(?!), ...)
- ▶ 스니펫 활용
- ▶ $\text{T}_{\text{E}}\text{X}$ 매크로 정의
- ▶ [조수에게 도움 받기](#)

■ $\text{T}_{\text{E}}\text{X}$ 을 쓰다가 질문이 생겼을 때?

- ▶ Google-fu (Bing-fu(?!))
- ▶ 로컬 고수에게 질문
- ▶ KTUG, $\text{T}_{\text{E}}\text{X}$ StackExchange, ...

글글글

일반적으로 글을 쓰는 것은 고된 노동

■ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 으로 문서를 작성하는 더욱 큰 노동

- ▶ 단순히 내용이 아니라 모양에 대해서도 신경 써야 함
- ▶ 보기 좋은 떡이 먹기도 좋다 \leftrightarrow 빛 좋은 개살구

■ 긴 매크로들을 입력하기 위해 필요한 손재주?

- ▶ 단축기 숙달 (Vim-fu, Emacs-fu, VSCode-fu(?!), ...)
- ▶ 스니펫 활용
- ▶ $\text{T}_{\text{E}}\text{X}$ 매크로 정의
- ▶ [조수에게 도움 받기](#)

■ $\text{T}_{\text{E}}\text{X}$ 을 쓰다가 질문이 생겼을 때?

- ▶ Google-fu (Bing-fu(?!))
- ▶ 로컬 고수에게 질문
- ▶ KTUG, $\text{T}_{\text{E}}\text{X}$ StackExchange, ...
- ▶ [조수에게 도움 받기](#)

내용 ↔ 모양

내용 ↔ 모양

여러 자연어 및 프로그래밍 관련 작업에 적합한 ChatGPT는 두 마리 토끼를 다, GitHub Copilot은 국조적인 “자동 완성” 수준에서 도움.

대형 언어 모델 (Large Language Model)

- ChatGPT: OpenAI GPT-3.5/GPT-4
 - ▶ GPT-3: 1750억, GPT-4: 1조
 - ▶ 플러그인 스토어 등장
- GitHub Copilot: OpenAI Codex (GPT-3)

시나리오: 세미나 발표

1. 탐색

- ▶ 브레인스토밍
- ▶ 문헌 조사

시나리오: 세미나 발표

1. 탐색

- ▶ 브레인스토밍
- ▶ 문헌 조사

2. 구체화

- ▶ 주제 선정
- ▶ 개요 작성

시나리오: 세미나 발표

1. 탐색
 - ▶ 브레인스토밍
 - ▶ 문헌 조사
2. 구체화
 - ▶ 주제 선정
 - ▶ 개요 작성
3. 자료 수집 및 가공

시나리오: 세미나 발표

1. 탐색
 - ▶ 브레인스토밍
 - ▶ 문헌 조사
2. 구체화
 - ▶ 주제 선정
 - ▶ 개요 작성
3. 자료 수집 및 가공
4. 발표 자료 작성

ChatGPT 데모: 개요 작성

GPT-4 + Plugin Prompt Perfect

perfect I need to prepare a seminar for a LaTeX workshop. The topic is "TeX in a modern computing environment: engine, automation, and AI". For the engine, I will introduce the use of Tectonic; for the automation, GitHub actions; for AI, LLM like you, ChatGPT and GitHub Copilot. Can you provide me an outline? The seminar is about 30 minutes long.

ChatGPT 데모: 모르는 패키지 사용

GPT-4 + Plugin Ask Your PDF

I would like to express a BNF expression of a simple call-by-value lambda calculus using simplebnf LaTeX package. Here is the manual for it: <https://mirror.kakao.com/CTAN/macros/latex/contrib/simplebnf/simplebnf-doc.pdf>

Expressions	e	$::=$	v	variable
			$\lambda v . e$	abstraction
			$e_1 e_2$	application

GitHub Copilot 데모: 명령어 자동 완성

```
% centered image  
\begin{center}  
  \includegraphics[width=\linewidth]{copilot.png}  
\end{center}
```