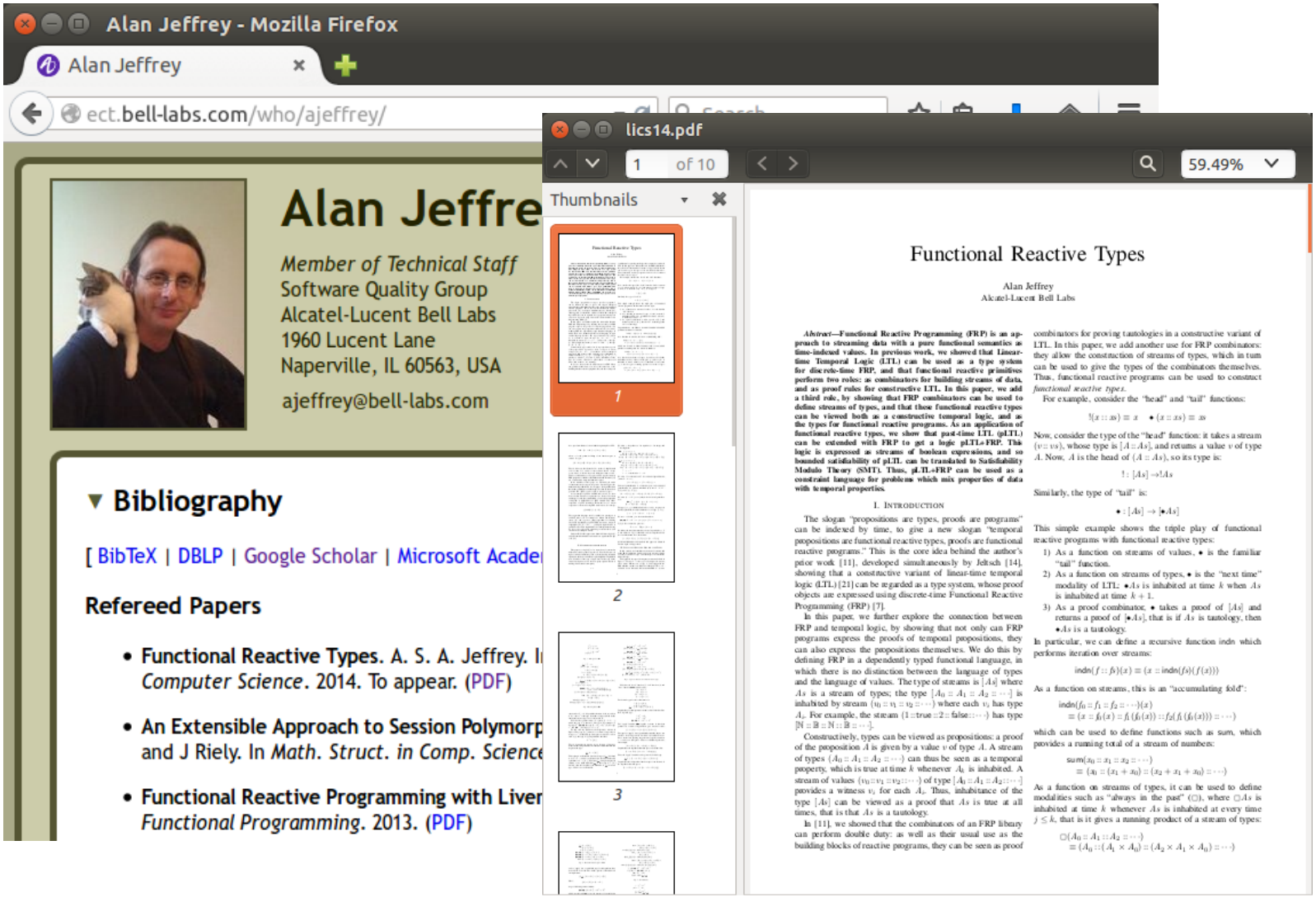
TOOLS FOR WRITING A RESEARCH PAPER

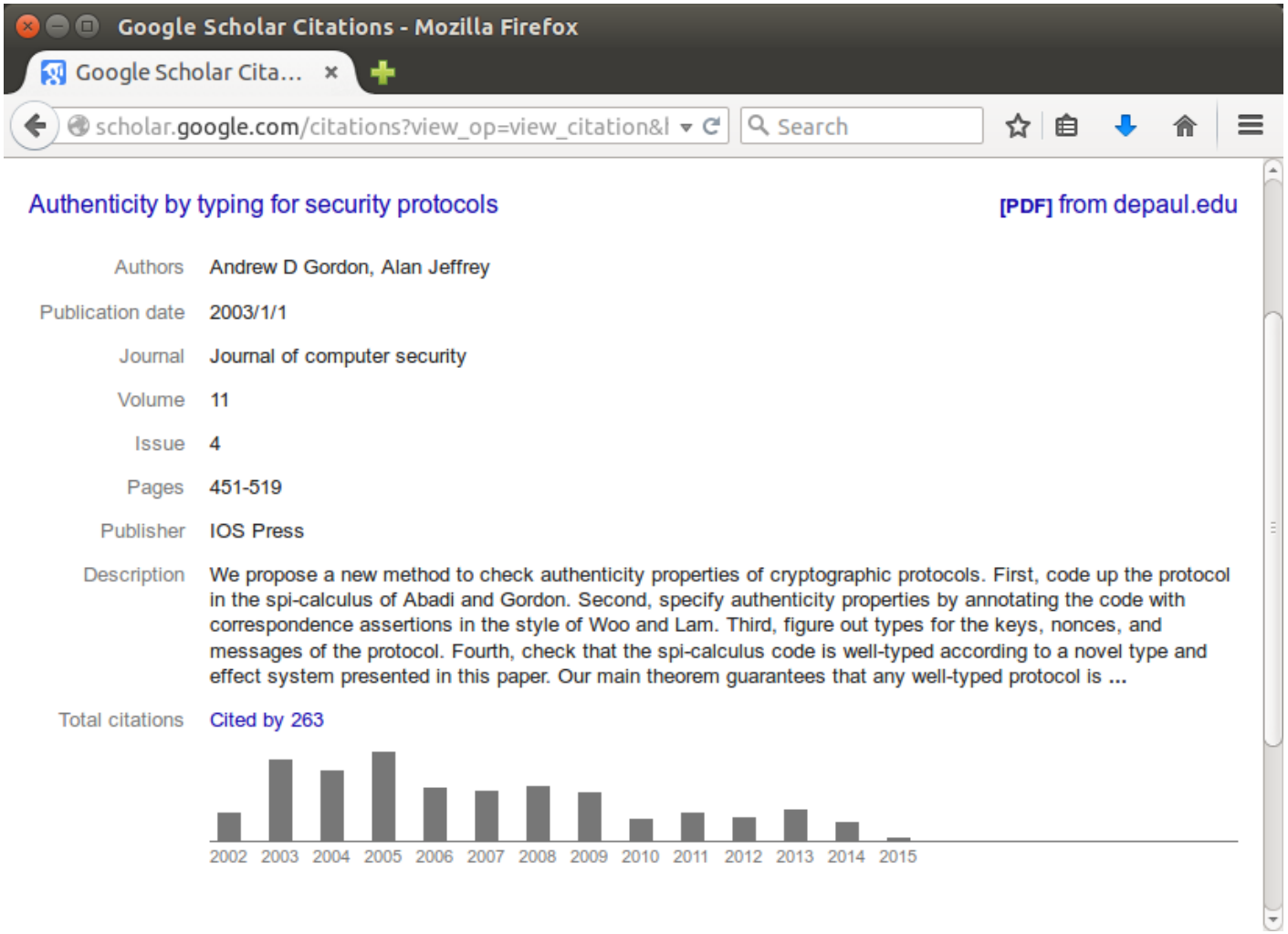
Alan Jeffrey (Bell Labs)

University of Illinois at Chicago, January 2015

PDF FILE: THE ATOMIC UNIT OF PUBLICATION

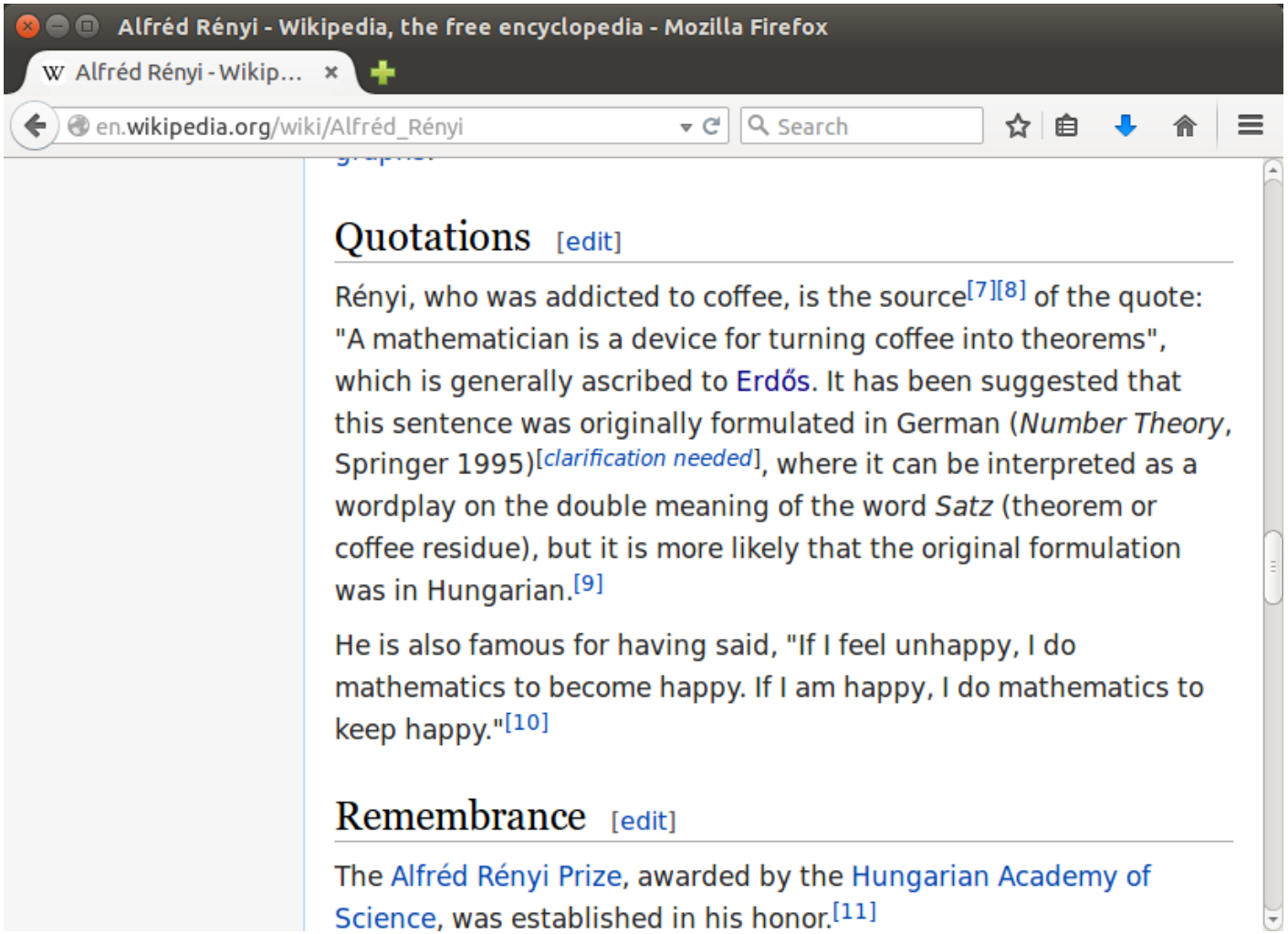


PDF FILE: THE ATOMIC UNIT OF PUBLICATION



PDF files aren't just read by humans...

PDF FILE: THE ATOMIC UNIT OF PUBLICATION



A computer scientist is a device for turning coffee into PDF files.

TOOLS

My toolset (YMMV):

- Google Drive (cloud storage)
- Git (source control)
- Evince (PDF viewer)
- Emacs (text editor)
- Make (build management)
- LaTeX(document preparation)
- BibTeX(bibliography)
- Beamer (presentation)

TOOLS

My toolset (YMMV):

- Google Drive (cloud storage)
- Git (source control)
- Evince (PDF viewer)
- Emacs (text editor)
- Make (build management)
- LaTeX(document preparation)
- BibTeX(bibliography)
- Beamer (presentation)

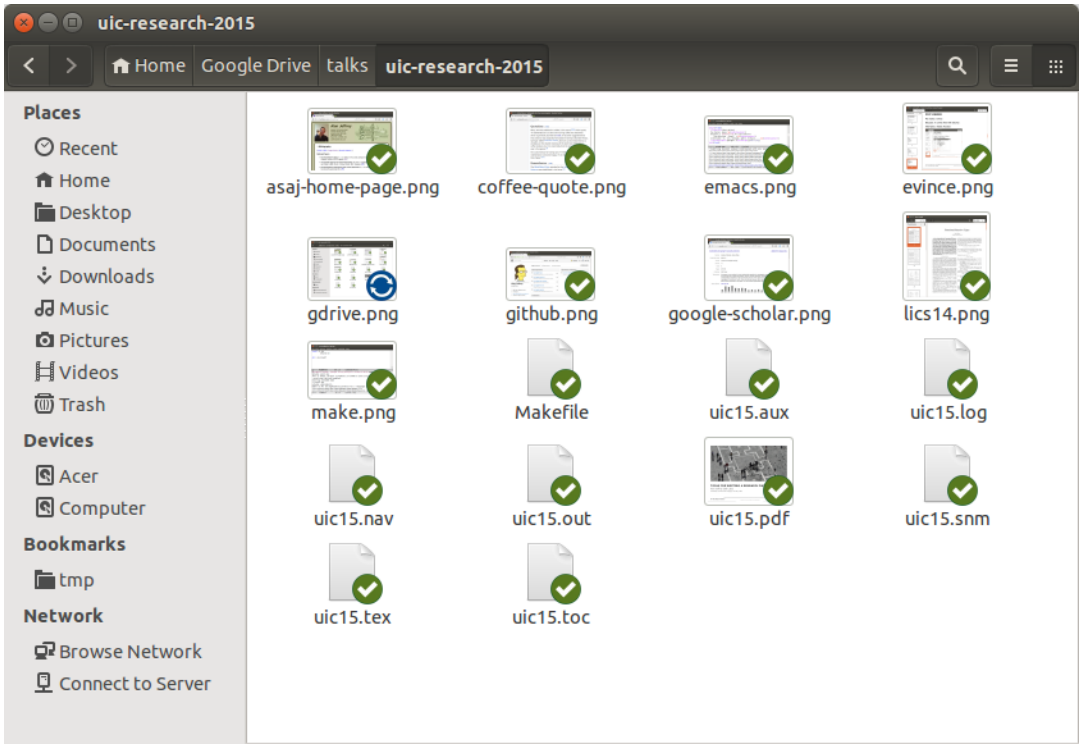
Note: many of these are tools for maintaining software.

CLOUD STORAGE

My choice: Google Drive

Because: $10/TB/mo, clients

Alternates: DIY, Amazon, MyDrive, ...

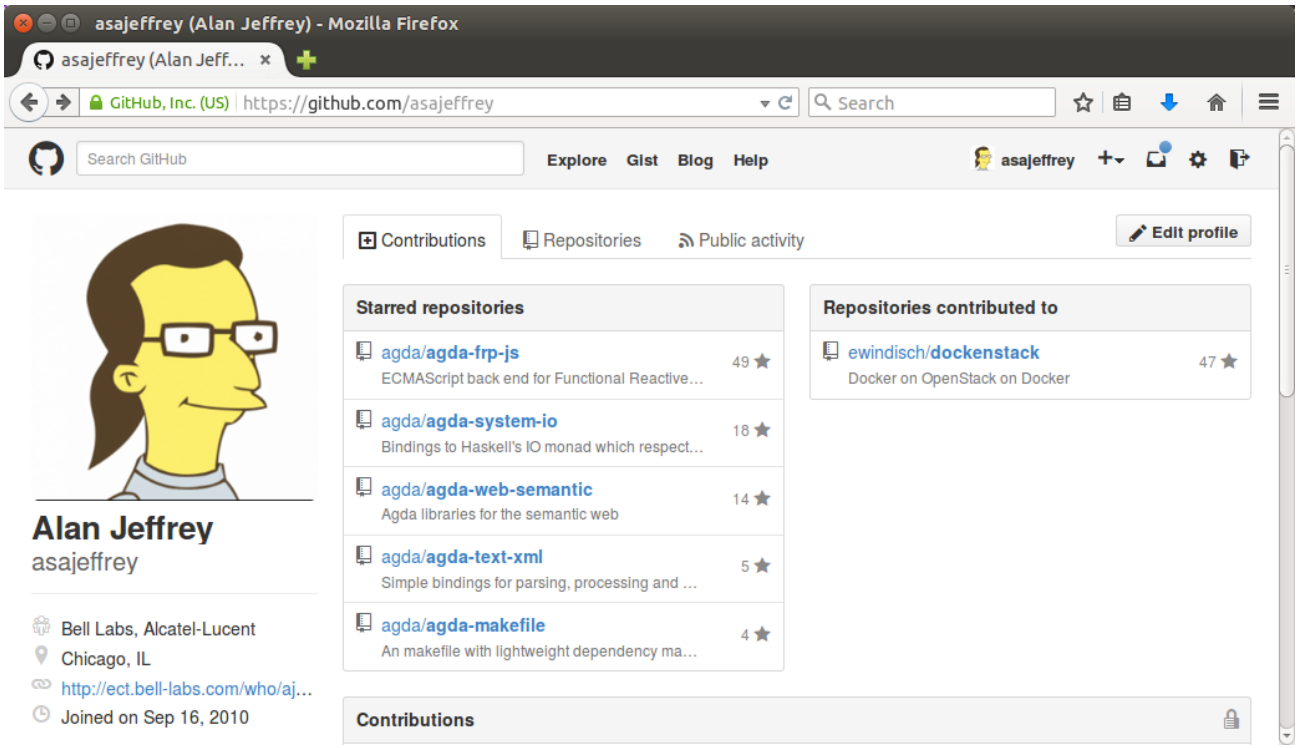


SOURCE CONTROL

My choice: git

Because: github, distributed source control

Alternates: svn, darcs

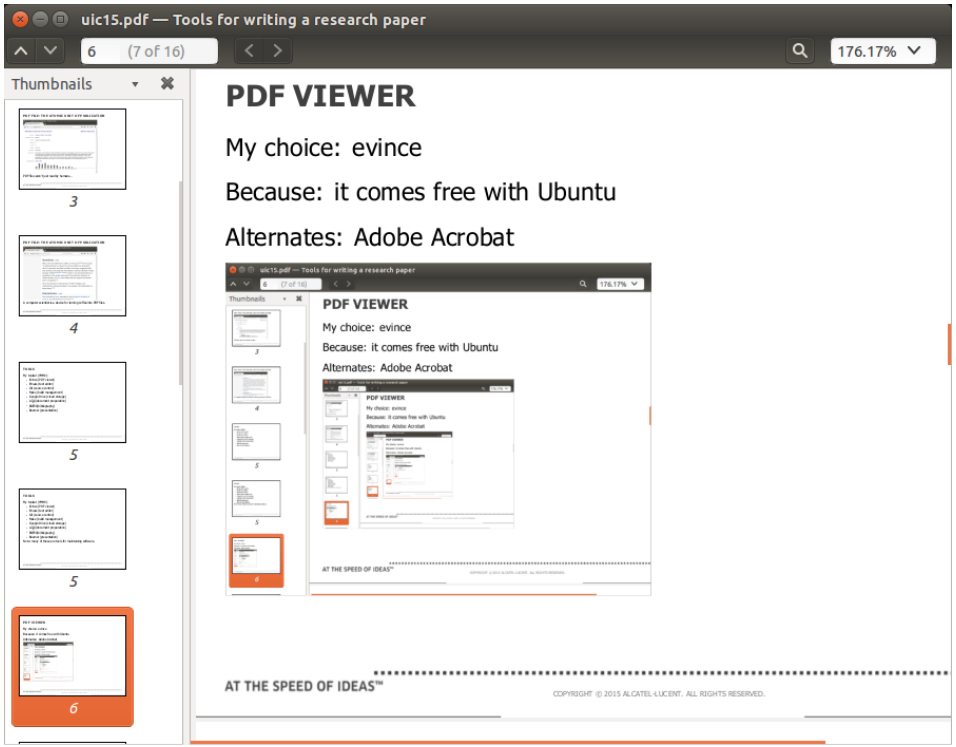


PDF VIEWER

My choice: evince

Because: it comes free with Ubuntu

Alternates: Adobe Acrobat

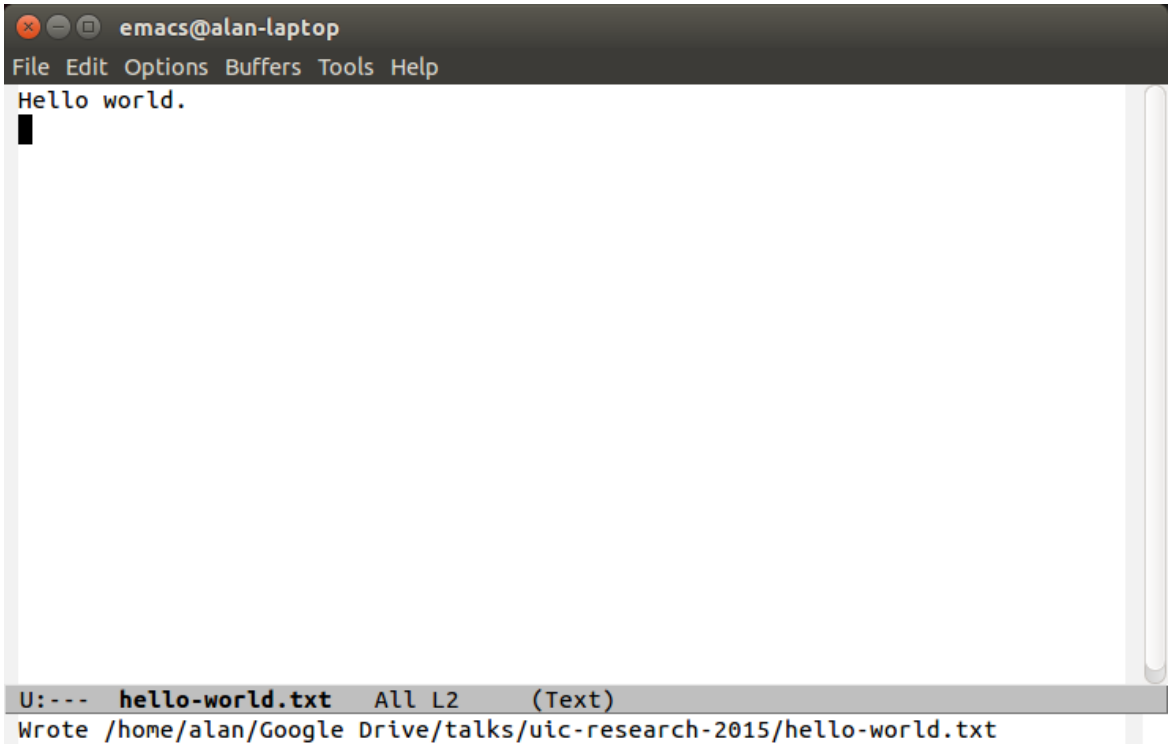


TEXT EDITOR

My choice: emacs

Because: I ♡ M-x set-controls "/usr/bin/sun --heart" t

Alternates: vi, vim, Notepad, gedit, Eclipse, ...

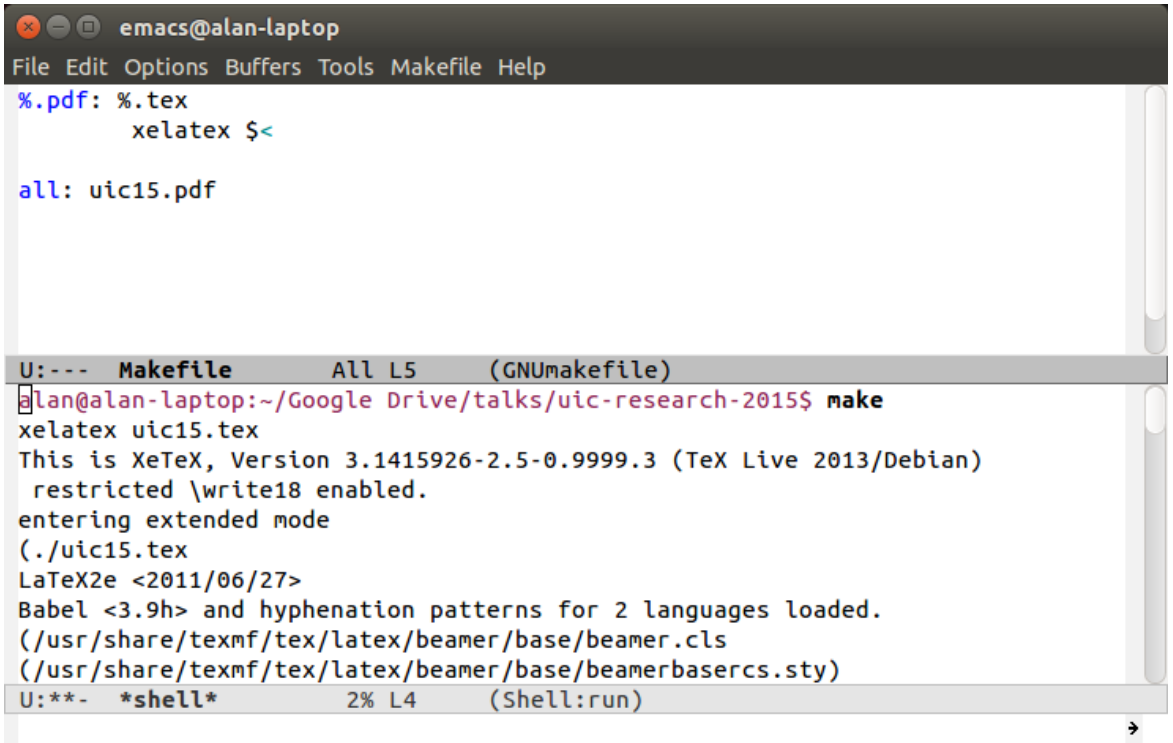


BUILD MANAGEMENT

My choice: make

Because: I am old school

Alternates: ant, maven, grunt, ...

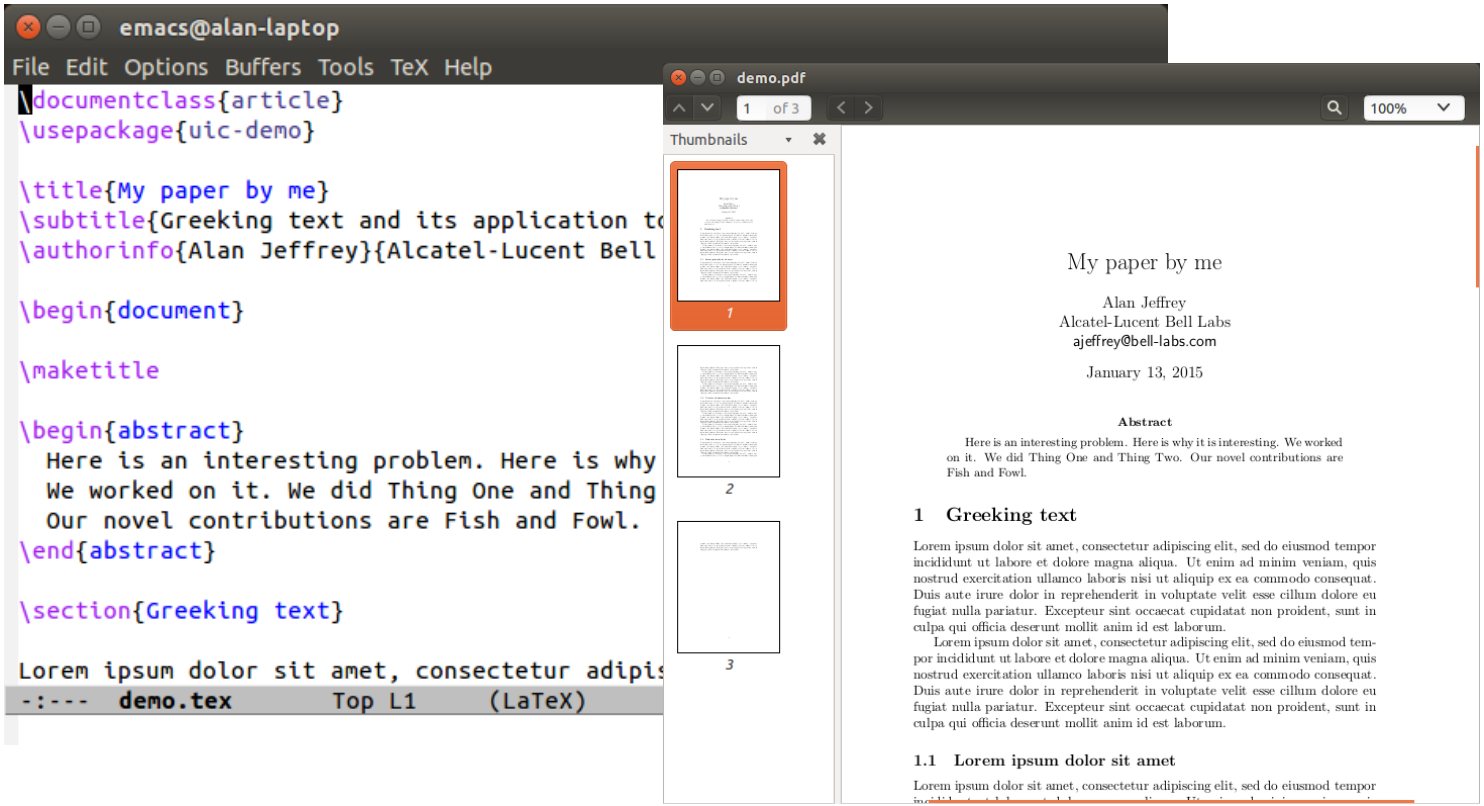


DOCUMENT PREPARATION

My choice: LaTeX

Because: I was on the dev team

Alternates: Word, LibreOffice

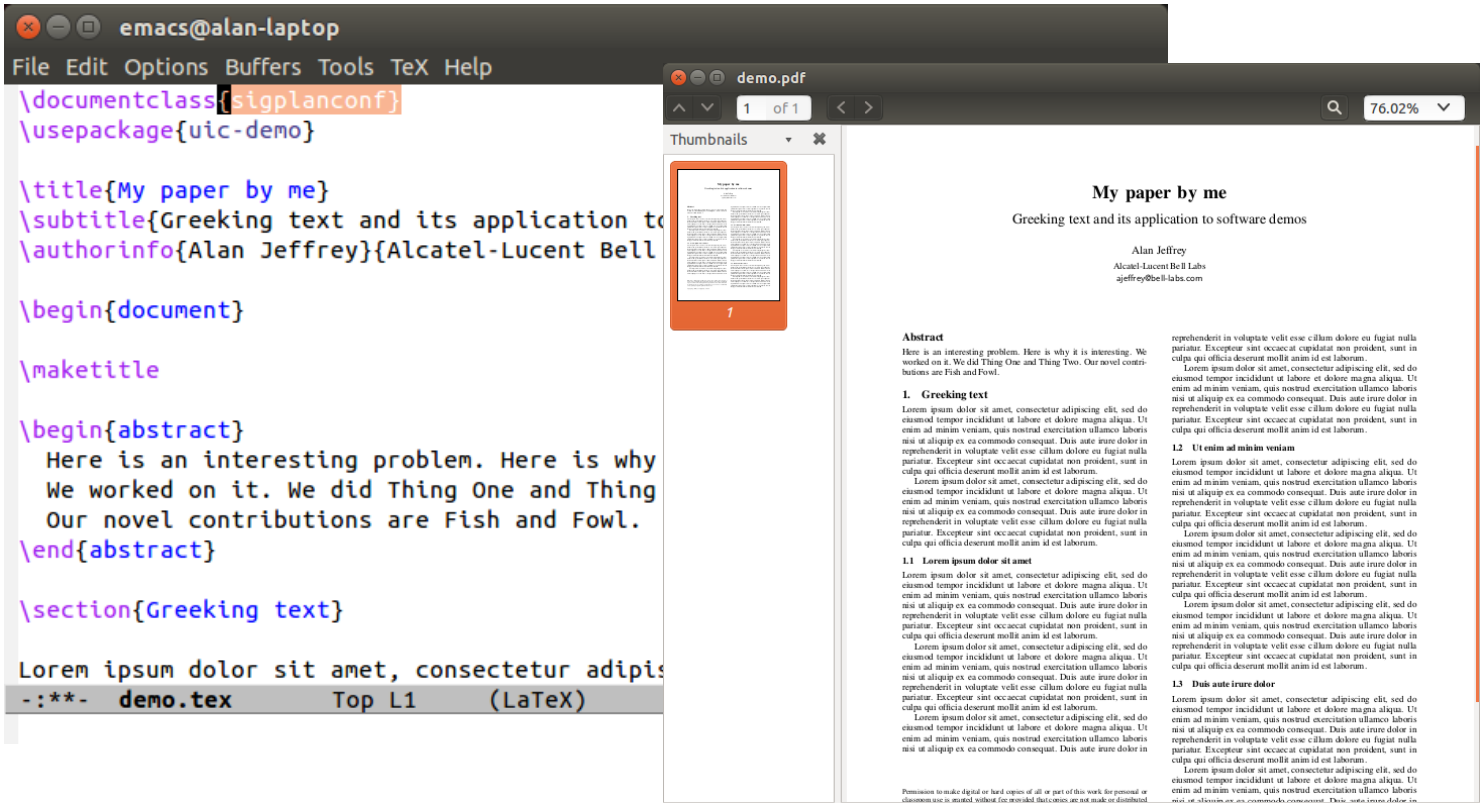


DOCUMENT PREPARATION

My choice: LaTeX

OK, not just because I was on the dev team

Supported by the high-impact venues (IEEE, ACM, Springer, Elsevier...)



DOCUMENT PREPARATION

My choice: LaTeX

OK, not just because I was on the dev team

- Not WYSIWYG
- Markup is logical structure, not visual structure
- High-quality typesetting, especially of mathematics
- Plays well with source control, build management
- Open source (before there was such a thing)
- Package infrastructure (ctan.org has 4820 packages by 2238 people)

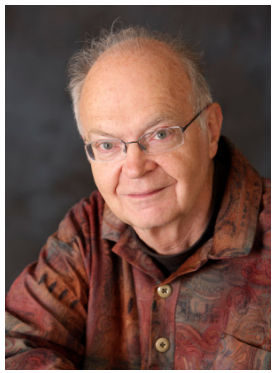
DOCUMENT PREPARATION

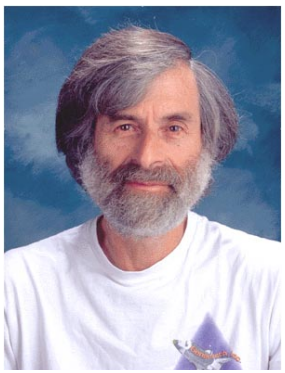
My choice: LaTeX

OK, not just because I was on the dev team

- Not WYSIWYG
- Markup is logical structure, not visual structure
- High-quality typesetting, especially of mathematics
- Plays well with source control, build management
- Open source (before there was such a thing)
- Package infrastructure (ctan.org has 4820 packages by 2238 people)

Developers include not just one, but two Turing Award winners



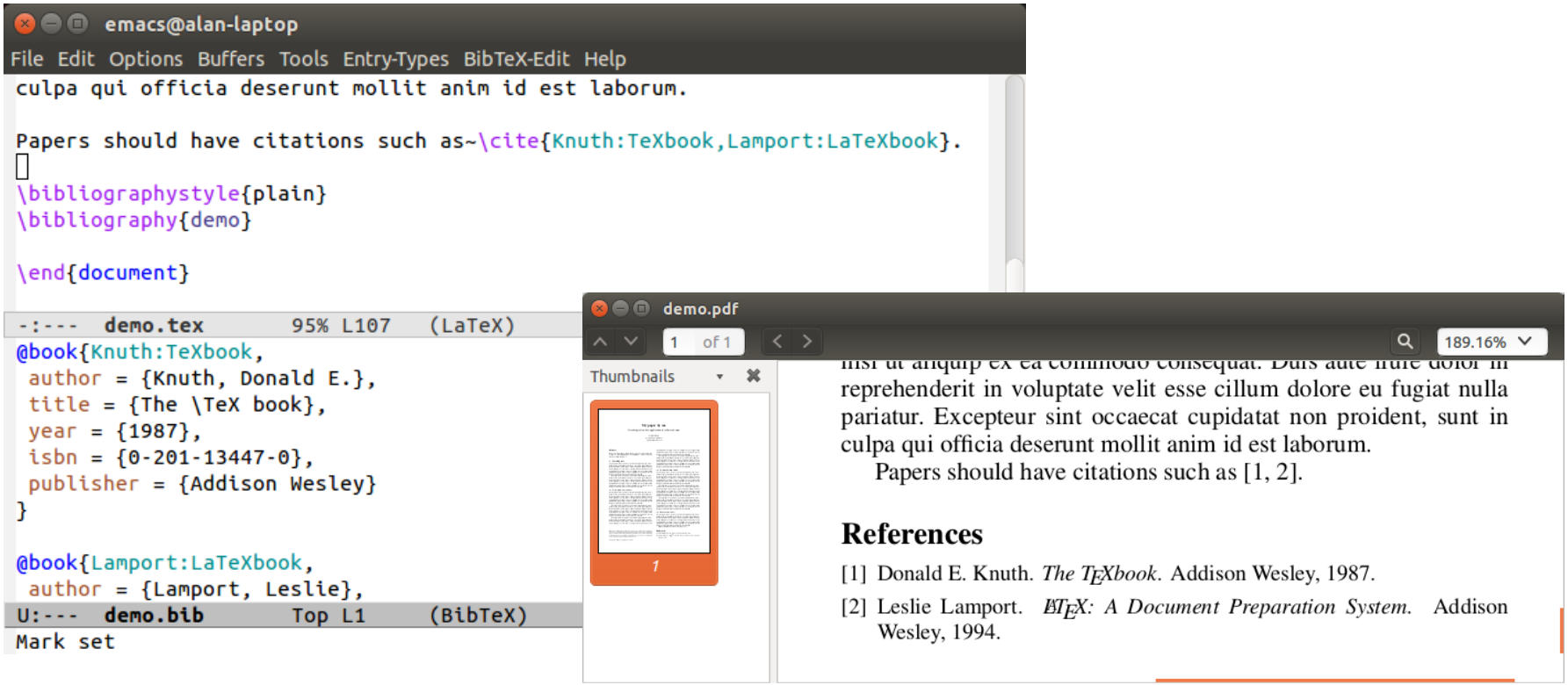


BIBLIOGRAPHY

My choice: BibTeX

Because: everyone uses it

Alternates: ?



PRESENTATION

My choice: Beamer

Because: cut-and-paste from paper to slides

Alternates: Keynote, PowerPoint, LibreOffice

emacs@alan-laptop

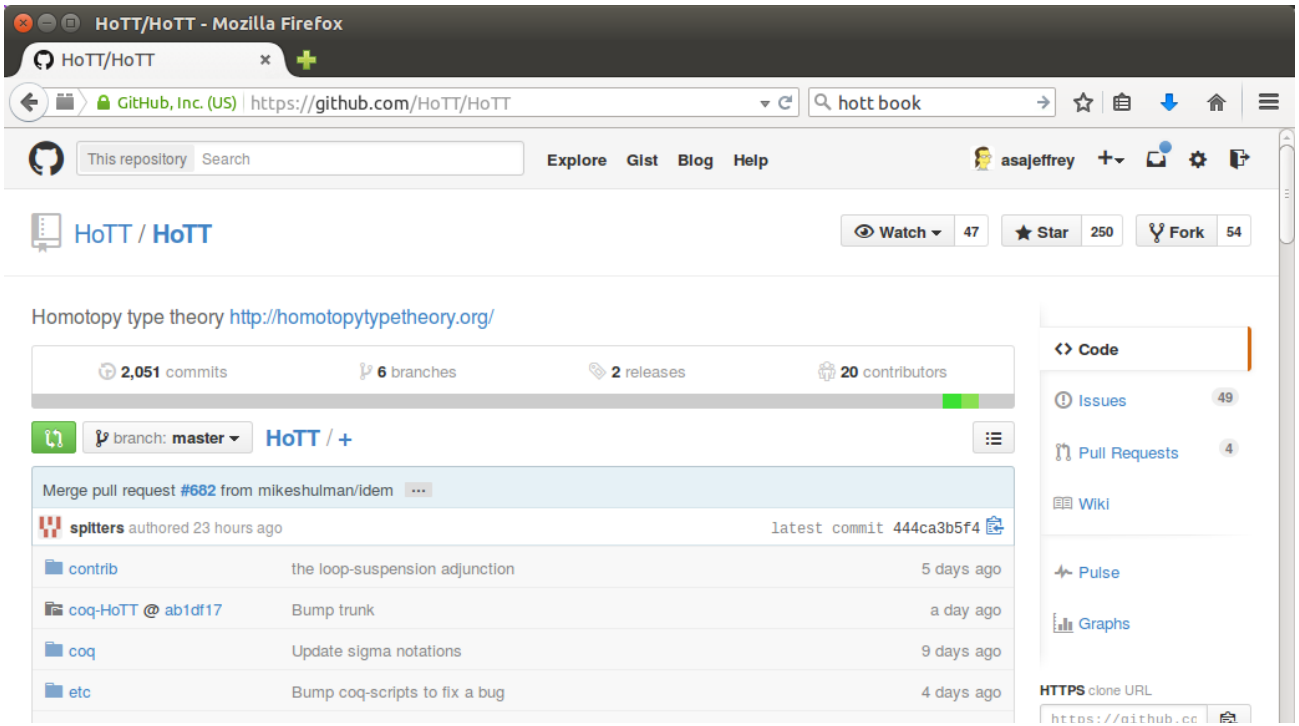
File Edit Options Buffers Tools TeX Help

```
\begin{frame}
  \frametitle{Presentation}
  My choice: Beamer \\[\medskipamount]
  Because: cut-and-paste from paper to slides \\[\medskipamount]
  Alternates: Keynote, PowerPoint, LibreOffice \\[\medskipamount]
  \includegraphics[height=.5\textheight]{beamer.png} \\[\medskipamount]
\end{frame}
```

```
-:---  uic15.tex      86% L159   (LaTeX)
<use  "alu2-footer.png" > [12] <use  "latex-sigplanconf.png" >
<use  "latex-sigplanconf-pdf.png" > <use  "alu2-footer.png" > [13]
<use  "knuth.jpg" > <use  "lamport.jpg" > <use  "alu2-footer.png" > [14]
<use  "knuth.jpg" > <use  "lamport.jpg" > <use  "alu2-footer.png" > [15]
<use  "bibtex.png" > <use  "bibtex-pdf.png" > <use  "alu2-footer.png" >
[16] <use  "beamer.png" > <use  "alu2-footer.png" > [17]
<use  "alu2-footer.png" > [18] <use  "alu2-footer.png" > [19] (./uic15.aux) )
Output written on uic15.pdf (19 pages).
Transcript written on uic15.log.
alan@alan-laptop:~/Google Drive/talks/uic-research-2015$
U:**-  *tex-shell*    Bot L10778  (TeX-Shell:run Shell-Compile)
```

CASE STUDY: THE HOTTBOOK

- Homotopy Type Theory
- From the IAS 2012-13 year on Univalent Foundations of Mathematics
- 20 authors, 600pp, mostly written in a year
- Used git (on github.com/HoTT), emacs, LaTeX, BibTeX
- Licensed under Creative Commons, hardback book is $30



SUMMARY

Treat writing like software:

- Learn to touch-type
- Think about licensing
- Use source control, build management, etc.
- Distinguish source from executable
- Aim for minimal, elegant solutions
- Have professional pride in your work
- Make your work easy to navigate
- Think about the structure of your work (balanced tree if possible)
- Consider how humans will use your work

Q&A

Tools, research, working at a research lab...?