

Abstracts

LuaTeX: Messing around with tokens

Hans Hagen

Most TeX users only deal with (keyed in) characters and (produced) output. Some will play with boxes, skips and kerns, maybe even leaders (repeated sequences of the former). Others will be grateful that macro package writers take care of such things.

Macro writers on the other hand deal with properties of characters, like catcodes and a truckload of other codes, with lists made out of boxes, skips, kerns and penalties but even they cannot look much deeper into TeX's internals. Their deeper understanding comes from reading *The TeXbook* or even looking at the source code.

When someone enters the magic world of TeX and starts asking around a bit, he or she will at some point get confronted with the concept of tokens. A token is what ends up in TeX after characters have entered its machinery. Sometimes it even seems that one is only considered a qualified macro writer if one can talk the right token-speak. So what are those magic tokens and how can LuaTeX shed light on this?

In this presentation I will show examples of how LuaTeX turns characters into tokens. We will also pay some attention to the (un)usefulness of this.

Have no fear, MEGAPOST is here!

Taco Hoekwater

Anyone who has done serious work with MetaPost knows that it has quite a few implementation limits. It is not uncommon for moderately complicated graphics to occupy more than the amount of available internal memory, or to have macros that nest so deep that the stack is not large enough to hold them all. Also, values cannot be larger than 4095 without extra care.

MEGAPOST will alleviate these problems by a combination of dynamic data structure reallocation strategies and the use of a bigger internal storage type for numeric values.

DocScope Publisher: A large-scale project based on TeX

David Kastrup

The DocScope Publisher from QuinScope GmbH is focused on data-based publishing of input in XML form. At its core, currently L^ATeX, David Carlisle's `xmltex`, and pdfTeX are employed extensively. Current applications are the printing of financial reports and of a variety of product catalogs and online excerpts. Some of the problems occurring in large-scale, high-quality printing processes in the connection with TeX are explained, and solutions and products are shown.

Making of the TeX Collection

Manfred Lotz

In 1999, DANTE in collaboration with Lehmanns bookshop first produced a CTAN snapshot consisting of 3 CDs. In 2000 the TeX Live 5c CD-ROM was added and by 2002 the CTAN snapshot already consisted of 4 CDs.

After that, it was decided to produce a double layer DVD-9 consisting of a CTAN snapshot and the TeX Live 'live' image. The first DVD-9 was produced in 2003. In the following years the contents of the DVD was expanded to contain also ProTeXt, MacTeX, and ConTeXt.

This talk gives an overview about the problems we encountered when producing the first DVD-9 in 2003 and shows what steps were involved in creating the DVD.

Sanskrit typesetting from a user's perspective

Manfred Lotz

Sanskrit is an ancient Indian language, whose meaning to India is comparable to what Greek and Latin mean to European languages.

Sanskrit typesetting is very complicated, due to the existence of over 800 required ligatures. It will be shown what options are available to typeset Sanskrit under L^ATeX. The article focuses on the use of the packages `skt` and `devnag`. Their strengths and weaknesses will be discussed and examples will be given to enable the reader to get started easily.

Advanced mathematics features, for PDF and the Web

Ross Moore

Modern L^ATeX systems, creating PDF documents, support navigational features that can be usefully exploited to make technical documents much more usable than just an online facsimile of a traditional printed document. In this talk I will show various features that were developed specifically for an online version of a mathematics journal. These features include:

- metadata attachments to the PDF document;
- bookmarks to all (sub-)sections, figures, tables, theorems, and cross-referenced equations, etc., with use of Unicode strings for bookmarks, including the (simple) mathematical expressions that occur within section titles;
- searchability and copy/paste of mathematical expressions where the PDF browser recognises and supports embedded CMAP resources for the standard (e.g., CM and AMS) math fonts;
- draggable pop-ups of floating figures and tables;
- semi-automatic generation of hyperlinks to MathSciNet for bibliographic entries; i.e., helping build the *Reference Web*.

These features are all implementable now using pdfTeX; many work also with other PDF-aware drivers.

Links to documents using them can be found via the web pages at:

<http://www.austms.org.au/Publ/Bulletin/V72P1/>

<http://www.austms.org.au/Publ/Bulletin/V72P2/>

<http://www.austms.org.au/Publ/Bulletin/V72P3/>

Also, with the X_YTeX implementation of TeX, there is now direct support for OpenType fonts, and the possibility of typesetting both text and mathematics from a single font. The STIX fonts will be available soon. I will also show the results of work done by Will Robertson and myself, with criticism and advice from Chris Rowley, building upon the work of Jonathan Kew in extending X_YTeX to support mathematics. This could well become the basis of L^ATeX support for mathematics in the future.

Data structures in TeX

Marek Ryćko

The foundation of a programming language is support for data structures and operations to be performed on them. TeX, as a programming language, lacks most of the data structures known from other languages. I show how to design some basic data structures with appropriate operations and how to implement them in TeX's language in a very simple and efficient way. One of the structures introduced is a list of atomic elements, where atomic elements are TeX's token sequences. This uniform and clean way of using lists of elements makes TeX's programming much simpler and TeX's programs (macros) much more readable.

Polishing typesetting blocks

Marek Ryćko

It is now year 2007, 30 years since Donald Knuth started to implement TeX. During those 30 years thousands of programs, packages, styles, formats, fonts, scripts have been implemented in various languages, that support the "TeX way of thinking" about typesetting. We have a huge pool of programs that are capable of realizing lots of important tasks. But there is often a problem with connecting various programs together to easily achieve more complicated, structured goals.

TeX's approach to typesetting is essentially the possibility of using TeX and other related programs as links in chains or building blocks in higher level constructions. But parts of programs would also be very useful as building blocks. For example TeX's hyphenation algorithm would be very useful in many applications, not just in TeX itself. Similarly, the METAFONT and MetaPost algorithms (by John Hobby) for Bezier curve interpolation might be used in arbitrary 2D graphics applications. The monolithic constructs like TeX or MetaPost contain inside lots of programming pearls, but the pieces cannot be used separately. We have perfect building blocks of various kinds, but still are unable to build pyramids. Hopefully, after some polishing of the blocks and also after cutting some monoliths into smaller pieces, the task can be achieved.

Designing graphical signs and logotypes

Andrzej Tomaszewski

Participants in this workshop will have a chance to measure themselves designing graphical signs and/or logotypes.