Selected Abstracts from TEXperience

Jan Štěpnička, Jan Šustek:
Using TEX for Organizing Vojtěch Jarník International Mathematical Competition

The University of Ostrava organizes Vojtěch Jarník International Mathematical Competition every year. Organization of the competition contains of many tasks. A large part of them is done by TEX. On the lecture we briefly present the whole process from registration of participants to putting the results on the Internet. We describe several parts which can be useful for other TEX users:

- during generation of the printing before the competition: simple loading of data from database, writing Unicode characters to a file or generating pseudorandom numbers,
- during procession of problems proposed for the competition: using a single source file for six different outputs or simple ignoring of macros and environments,
- during procession of results of the competition: loading results from spreadsheet or generation of diplomas.

It is likely that some of these parts are already implemented somewhere in the set of LATEX packages. In this case the lecture shows an alternative approach.

Zdeněk Wagner, Anshuman Pandey and Jaya Saraswati:
Development of xindy Sort and Merge Rules for Indic languages

An index is an important part of a book or a longer document. Normally in the past the index was prepared using the MakeIndex program which offered sorting according to English and German rules. A derived program called CsIndex allowed for the preparation of indices in the Czech and Slovak languages. Sorting in other languages was difficult because the algorithm was hard coded in the C program. The situation was changed in MakeIndex 3.x which is now superseded by xindy. The sorting algorithm is now defined in tables that are present in standalone modules.

This contribution shows how xindy features can be deployed in sorting Indic languages where complex scripts are used. Since the original TEX was unaware of Unicode, transliteration schemes were introduced in the past. XƎTEX is quite popular mainly among new users and the transliteration systems are still in current use. Xindy is able to handle the input in the transliteration as well as the UTF-8 encoded text. The algorithm is demonstrated in the Hindi and Marathi languages. It also shows how xindy integrates with different TEX engines.

David Březina:
Skolar – Designing a Typeface for Academic Publications
Skolar is a text serif, originally designed with scholarly and multilingual publications in mind. The typeface maintains its credibility while incorporating a subtle personal style, neither neutral nor conspicuous.

Prominent serifs and low-contrast modulation add to its robustness, and, together with a relatively large x-height, improve the typeface's readability in small sizes. This family of three weights with their respective italics and large character set is flexible enough for complex text settings and editorial work. It also becomes distinctive in bigger sizes, fitting the demands of corporate design.

There have been many practical solutions introduced in the typeface; the capitals are rather low in comparison to the ascenders. This gives the typeface even texture and more space for capital diacritical marks. The italic has a shallow angle and large counters for better readability in small print. It is easily recognized but not ostentatious, blending well with the uprights. Semibold is weighted to emphasize text blocks, where Bold is intended for word clusters. The family includes a complex set of smart arrows which can be easily keyed and infinitely combined using OpenType features. It was released with TypeTogether and it is available for web using Typekit as well.

Roman Trušník:
Typesetting *Bibliography of American Literature in Czech Translation: 2000 & 2010*
After the experience with data processing and typesetting the first part of *Bibliography of American Literature in Czech Translation* (Olomouc: Votobia 2000, 3 volumes, 1882 pages) in the late 1990s in Aldus PageMaker 5.0, new possibilities were explored before the preparation of the second part (to be published in 2010). The paper deals with the issues that had to be addressed as requirements included seamless processing of large multi-language multi-alphabet multi-font structured documents. After extensive testing (several monographs and volumes of conference proceedings, *Moravian Journal of Literature and Film*), XeLaTeX was adopted as the typesetting platform for the project.

Jan Přichystal: TeX Typesetting on Web: tex.mendelu.cz/en
This talk introduces web application TeXonWeb. This interface helps beginners to start with typesetting system TeX offering them easy to access and easy to use editor, TeX compiler, code wizards and document templates. Application could also be helpful in situations when user wants to produce high-quality document but no computer with TeX is available. Introduction to used technologies, features and future visions will also be included.

Jaroslav Hajtmar: The ScanCSV.lua Library
In computerised data processing, data stored in CSV files (Comma Separated Values) are used frequently. The presentation describes the author's library ScanCSV.lua and the method of its creation. The practical examples of its use in ConTeXt MkIV will also be demonstrated. The author shows how easy and swift is to create reports, letters, forms, certificates, invitations, business cards, double-sided cards, tables, animations, etc. using external CSV databases. Users of ConTeXt MkIV (but LuaLaTeX and LuaTeX too) can practically use data from external CSV tables in own documents through the TeX macros built by this library on-the-fly. It also means that we have this data available in an attractive, simple and natural way.

Jan Šustek: Macros That Handle Arithmetics with Big Numbers
In procedural programming languages a program calls functions with their arguments and the functions return their result. To avoid collisions, functions have their local variables. Result of a function is one of the local variables, but one can see it "nonlocally" immediately after the function call.

On the contrary, programming in TeX is based on expansions. Sequence of tokens is repeatedly expanded and when a particular token cannot be expanded, the main processor does the corresponing activity. The concept of result of a function does not make sense in TeX. If a definition, counter etc. is not local, then it is global.

We will show how it is possible, using expansions in TeX, to simulate function calls similarly as in procedural programming languages. The problem of (non)-local variables arises when one function calls another function. In this situation one can bypass the problem by a neat choice of macro names. But this is a nonnatural procedure and in the case when a function calls recursively itself it is not possible.

When one works with big numbers on computer, array is the most suitable data structure. Usual procedural programming languages know arrays. TeX primarily does not know arrays. We show one of possilble implementations of arrays in TeX. This implementation, however, is not good for sending the values between function. Hence we show another data structure and functions for conversion between different data structures. It is senseless to write that decadic expansion is not suitable.

As a demonstration of the mentioned function calls we implemented functions from number theory. The following short programs allows us to decide whether c is composite.

$$c = (2^{107} - 1)(2^{127} - 1) =$$

$$= 2760698538716225514973902344910793166845871614262060116995480300080332 9$$

```
\SET\a\POWER(2)(107)
\SET\a\SUBTRACT[\a](1)
\SET\b\POWER(2)(127)
\SET\b\SUBTRACT[\b](1)
\SET\c\ULTIPLY[\a][\b]
\ArrayToPrint\c\c
\SHOW c
\SET\r\ISCOMPOSITE(\c)
\SHOW r
\bye
```

Function `\ISCOMPOSITE` performs Fermat test calling function `\FERMAT`, which computes power in modular arithmetic calling function `\POWERMOD`, which multiples its arguments calling function `\SQUAREMOD`, which, when looking for remainder, uses division calling function `\DIVIDE`, which uses subtraction calling function `\SUBTRACT`.