Old Software Treasuries
and
troff and friends

markus schnalke <meillo@marmaro.de>

ABSTRACT

Old software is not obsolete per se. In contrast, some software tools are still alive despite their age. Others became mostly forgotten but are still valuable helpers. No matter which group programs belong to, one should be able to use the classic Unix tools – those that are available on every Unix machine. `Ed`, `sed`, and `awk` are three of them.

`Troff`, `nroff`, and their friends (`tbl`, `eqn`, `refer`, `pic`) are also mostly vanished from the users view. But they offer high quality typesetting on the level of TeX. Man books were, and still are, typeset with their help.

This is the handout to two talks[1] [2] which introduce these tools, explain why they are the way they are, and show how to use them.

Historical background

To understand why these tools are like they are, one needs to know how computing was back then when Unix was written.

It was in the late sixties and early seventies at Bell Labs: Computers were as large as closets and people used them through terminals. Terminals consisted of a keyboard and a line printer; no screen terminals. Those line printers were quite slow, about 10–15 characters per second, thus output needed to be low to increase the working speed. The word "screen-orientation" was not even invented and editors were line editors, like `ed`.

Ed

Thompson wrote `ed` in the early seventies, based upon `qed`, which was a popular editor at that time. He equipped `ed` with the first implementation of Regular Expressions, a powerful pattern matching language.

Until the rise of screen-oriented (or visual) editors, `ed` was the primary Unix editor – the standard. Today, `vi`, `emacs`, or even graphical editors are what programmers use. Nonetheless, `ed` is still the standard. It's the only editor that resides in `/bin`. It is universally available and usable with any terminal setup.

It is definately worth and enlightening to learn `ed`.

Sed

McMahon's `sed` is a modified `ed` to be used in pipelines. This at least is how the "stream editor" started.

It is not a hacked `ed` anymore, but a stand-alone tool in the Unix toolchest which one doesn't want to miss. Today, `sed` is, besides `grep`, one of the best known helpers on a Unix system.

As `ed` and `sed` share about eighty percent of their commands, one gets two tools for the price of one.

Awk

In the late seventies, Aho, Weinberger, and Kernighan developed the pattern matching language `awk` to avoid complex text processing constructs in the shell.

`Awk` is a real programming language with variables and program flow control. It is mainly designed to write small programs with a few lines, thus it is perfectly suited for the daily needs in shell sessions.

In contrast to `perl`, `awk` is available on every Unix system. And it offers almost all of `perl`'s text processing features.

Troff and nroff

The main two document preparation program in Unix are `troff` and `nroff`.

First was `nroff` which was written by Joe Ossanna, based on `roff`, which is McIlroy's Unix version of RUNOFF. `Nroff` generates plain text output for line printers.

A few years later, when the Bell Labs got a phototypesetter, Ossanna wrote an improved version called `troff`. This one supported the new features of the phototypesetter, like different fonts, different font sizes, more characters, fine resolution movements, and the like.

After the advent of `troff`, `nroff` did not vanish, but it stayed to render plain text files, like man pages for instance. (The following text includes `nroff` when the term "`troff`" is used.)

Macro packages

Naked `troff` becomes a pain as documents grow. This problem is solved with macro packages. They encapsulate formating commands into logical entities like headings, footnotes, and paragraphs.

Using a macro package is much like programming in a high-level programming language instead of assembler. Thus it is not surprising, that macro packages are widely used. Popular ones are: ms (the standard macros), mm (the successor of ms), man (for man pages), and me.

Formating a document with the ms macros is done like this:

```
troff -ms doc.ms
```

Preprocessors

`Troff` is able to generate any possible output, but this will end up in composing everything out of small lines. This is not what users want to do by hand.

Thus a set of preprocessors were developed, which take some high-level input and convert it to low-level drawing commands for `troff`.

Popular preprocessors are:

- `tbl` which formats tabulars
- `eqn` for formulas in arbitrary complexity
- `refer` for literature references
- `pic` to generate diagrams

Preprocessors get prepended to `troff` in a pipeline, as they pass everything through which is not for themselfs:

```
tbl doc.ms | eqn | troff -ms
```

Many more preprocessors are available and it is easy to write one yourself.

This pattern of extending software is a great way to avoid complexity when adding features.

Get the software

The here presented software is all freely available. It is even available in different implementations. The GNU versions are probably best known, but are very often bloated.

The better choice are the Heirloom project tools[3] which base on original Unix source code. They all support UTF-8 and some include new features, which were always added with the Unix Philosophy in mind.

References

1. Markus Schnalke, "Old Software Treasuries," in *ChaosSeminar*, CCC Ulm, 2009-10-12. http://ulm.ccc.de/ChaosSeminar/2009/10_Softwareschaetze
2. Markus Schnalke, "troff and friends," in *ChaosSeminar*, CCC Ulm, 2009-11-09. http://ulm.ccc.de/ChaosSeminar/2009/11_troff-and-friends
3. Gunnar Ritter, *The Heirloom Tools*. http://heirloom.sf.net
4. Brian W. Kernighan and Rob Pike, *The Unix Programming Environment*, 1984.