Toc, a multipass framework for troff

John Ankarström

23 June 2021

Introduction

Toc is a very simple solution – in 50 or so lines – to the problem of generating tables of contents and other forms of forward references in troff documents. It consists of a shell script called *toc* and a macro package called *toc.tmac*.

The *toc* script is a wrapper around troff, passing the document through the typesetter three times. In every pass, it sets the values of a registers and a string named *te* and *tf*, respectively.

The *toc.tmac* package defines two macros, named `te` and `to`. If the *te* register is non-zero, the `te` macro hands its argument to the `tm` request, which prints it on standard error, prefixed by "(toc)". If the *tf* string is non-empty, the `to` macro invokes the `so` request, reading and inserting lines from the file specified by *tf* into the troff document.

Operation

In the first pass, the *toc* script sets *te* = 1 and *tf* = empty. This enables the `te` macro, printing its arguments on standard error. The lines prefixed with "(toc)", printed on standard error, are written to the file *$g* with the "(toc)" prefix removed.

In the second pass, *toc* sets *te* = 1 and *tf* = *$g*. This enables the `to` macro as well, inserting the contents of *$g* into the troff source. The `te` macro is still activated and accordingly prints its arguments on standard error yet another time. The output is processed like earlier and written to the file *$h*.

In the third and final pass, *toc* sets *te* = 0 and *tf* = *$h*. This disables `te`, which means that nothing is printed on standard error, but `to` is still enabled, inserting the contents of *$h* into the document.

Three is the number of passes necessary – and sufficient – for generating forward references, such as tables of contents. Two passes are not enough, as the generated references may push a referent to the next page, rendering the generated references incorrect. To account for the addition of the generated references, a third pass is needed.

Usage

Macro package

The `te` and `to` macros do not apply any formatting to or perform any processing of their input. The `te` macro prints its arguments on standard error verbatim. For example, the request

```
.te .nr &ref \n%
```

will print

```
(toc).nr &ref \n%
```

on standard error. The "(toc)" prefix is removed before the line is written to the temporary file.

When `to` is invoked, it will literally insert

```
.nr &ref 3
```

into the troff source, assuming `te` was invoked on the third page.

As such, *toc* provides the tools needed to create forward references, including tables of contents, but the exact formatting must be programmed by the user himself.

Script

The *toc* script is designed to be called instead of troff. It is not a troff preprocessor, but a wrapper around it. Any arguments given to *toc* are passed to troff (or whichever program specified in the *TROFF* environment variable).

Pre-processors can be added by setting *TROFF* to a custom shell script:

```
$ cat xroff
#!/bin/sh
... | groff "$@"
$ TROFF=./xroff toc document.t
```

Examples

Table of contents

```
.so toc.tmac
.eo
.de he
. ft B
. sp 1v
\$*
. br
. ft
. te .the \n% \$*
..
.de the
. nr _ \$1
. shift
. ta 0 \n(.luR
. tc .
\$*         \n_
. tc
. br
..
.ec
.sp |1i
.to
.\" ...
.he First heading
.\" ...
.he Second heading
```