

# Work

by Jeffreys Copeland and Haemer



MICHAEL AVETO

*"I beheld the wretch—the miserable monster whom I created."*

— *Frankenstein*, Mary Wollstonecraft Shelley

*"How much easier it is to be critical than to be correct."*

— Benjamin Disraeli

## Odds and Ends

**A**h, May. We can't help but think of the late Bill Rotsler's cartoon cat sitting in a window distracted by a butterfly with the caption: "If cats had a longer attention span, they could rule the world." Just so we don't compete with the short attention span engendered by spring fever, we'll be covering a set of topics we've had kicking around in the attic for a while, none of which are enough to fill a complete column. Thus, we present you with our Franken-column.

But first, we found reader reaction to our February column, "Differences Among Women," educational (*Sun-Expert*, Page 38, <http://sw.expert.com/C9/SE.C9.FEB.99.pdf>).

### Differences Among Correspondents

Sometimes, life imitates that simple harmonic motion experiment from freshman physics. When we wrote a column on technology and reading, "A Short History of Reading" (*SunExpert*,

November 1998, Page 58, <http://sw.expert.com/C9/SE.C9.NOV.98.pdf>), we were surprised that the first two notes we received about it were both from women. We used this as a springboard for our February column. (As you know, there's sufficient publication offset that our observations and counter-observations occur in three-month waves.)

The level of reader interest in that column was higher than we'd anticipated. We must have struck a nerve—or a pair of nerves, as it turns out.

One reader, Pete Kernan, now has a Web page about four-tuples, <http://theory2.phys.cwru.edu/~pete/sequence.html>. There is also a related entry, A045794, in the "On-Line Encyclopedia of Integer Sequences," <http://www.research.att.com/~njas/sequences/index.html> (look for "Haemer," "Copeland" or "1 1 1 3 3 4 9").

We promised to report on the sex ratio of the responses to our column, and here it is: Within a one-month

period, we got 61 pieces of email from 34 unsolicited readers. Of these, nine respondents were women (including Ann Janssen, one of the correspondents to the November column) and 25 were men. The correspondents even included the husband-and-wife team of Shelly Shumway and Arthur Smith.

One (male) reader, Sal Mamone, sent us a pointer to some statistics he gathered about sex differences among his computer science students (see "Empirical Study of Motivation in a Entry-Level Programming Course," ACM SIGPLAN Notices, March 1992). We're not sure Sal's statistics completely apply—he was teaching COBOL and we think that puts an entirely different skew into the results—but they are interesting.

All the responses were interesting and gratifying, but what jumped out at us was the sexual dimorphism. Women sent email saying, "Interesting column, here's my opinion"; men sent email saying, "Interesting column, here's my code/math." We suspect that we could

write a Perl script to sort the responses by sex. One woman sent a technical response (containing math/code); three men sent nontechnical responses. The two sexes sent identical percentages of cross-dressed mail.

But we've still received no responses from Antarctica.

## Monopolies and You

It should be apparent by now that we're open-source bigots. We firmly believe in open systems with commodity hardware and, for the most part, nonproprietary software. But there are forces in the world that disagree with us. The largest of those is currently (and probably still will be, by the time you read this) on trial for violations of antitrust laws. We speak, of course, about Microsoft Corp.

We won't go into detail about the trial because whatever we say will be out-of-date by the time this is in print, but we'll note some interesting reactions:

- Amid the calls to break Microsoft into various slices, Perl consultant and author Tom Christiansen has suggested a different solution: He'd rather see the government make all of Microsoft's source code subject to the GNU Public License.

- An IBM Corp. spokesman has suggested that being sued for antitrust violations will destroy Microsoft. After all, he reasons, once IBM ran into antitrust trouble—a lawsuit that lasted for eight years, from the last day of the Johnson administration to the first day of the Carter administration—the company spent all its time consulting with lawyers about its plans rather than making new ones. We aren't sure if lawyers had anything to do with IBM's stupidity about the PC market and relative hardware pricing; that's what actually brought the former largest computer company in the world to the brink of death.

- There's a movement afoot from Linux users to get Microsoft and the hardware vendors to refund their license fees. In general, the Linux community buys commodity hardware but never boots the installed versions of Windows that are pre-installed on the machines. Open-source advocate Eric Raymond led a protest march over this issue at Microsoft's Silicon Valley offices in February. (See <http://www.netcraft.com.au/geoffrey/toshiba.html> for another example.)

- If Microsoft is broken up, we expect the century's first forced corporate breakup will be instructive. When Standard Oil was dismembered by the U.S. Supreme Court in 1911, conventional wisdom was that John D. Rockefeller's fortune would suffer. Quite the contrary, he was three times as rich within five years.

- Our guess is that no matter what Judge Thomas Penfield Jackson rules at the trial itself—which we expect will be against Microsoft in some form—Microsoft will appeal the verdict. The applicable appeals court has already demonstrated its computer illiteracy in its infamous “the browser is part of the operating system because Microsoft says so” decision. This means that all bets are off on the final outcome.

## Off By One

We've tripped over a variety of off-by-one errors in our time. In fact, we've complained about some of these before in this column. How do they show up and how do we prevent them?

Some examples of obfuscated code, and the fixes for them, may be instructive.

Taking our cue from Disraeli, in October 1996 (see “The Date Class, Part 2,” *RS/Magazine*, Page 32) we provided an example complete with `fix`, of the `%U` and `%W` specifiers to the `date` command and the `strftime()` interface. These two specifiers return the week number; in the case of `%W`, it's the number of weeks beginning on Sunday since January 1 of the current year. In many (nay, most) implementations, these are calculated incorrectly. Given a populated `tm` structure and the realization that the number of weeks since the beginning of the year is the same as the number of Sundays, it's pretty easy to calculate:

```
sun_week (tm)
struct tm *tm;
{
    int lastsun = tm->tm_yday -
        tm->tm_wday;
    return (lastsun+7)/7;
}
```

On the other hand, we have been known to get things wrong from time to time. A while back, we built a routine to overwrite a section of a file with nulls. Because the files could be large, we wanted the program to print a status bar to tell us how far along it was. It could print a dot for each block it wrote, but it would be far more effective to print a line of fixed length and then add a dot for each 5% of the write completed.

The code for writing the blocks is pretty obvious:

```
fprintf(stderr, "-20s (%07ld) ",
    filename, size);
/* insert [set up for status bar] here */
while( size > 0L )
{
    if( size >= BUFSIZ )
        write(fp,nullbuf,BUFSIZ);
    else
        write(fp,nullbuf,size);
    size -= BUFSIZ;
    /* insert [show status] here */
}
```

But how do we print the status? Our first cut looked something like this:

```
#define REPORT 20
/* set up for status bar */
osize = size;
nn = size / REPORT;
cnt = nn * (REPORT-1);

...

/* show status */
```

```
while( size < cnt )
{
  cnt -= nn;
  fprintf(stderr, ".");
}
```

But this, of course, results in an incorrect bar length if the file size is less than 20, or if rounding makes the initial value of `cnt` an odd number.

The correct code is as follows:

```
#define REPORT 20
/* set up for status bar */
osize = size;
nn = REPORT;

...

/* show status */
while( nn > 0 && size < (osize*nn/REPORT) )
{
  nn--;
  fprintf(stderr, ".");
}
```

An equally odd calculation occurs in the TeX macros for Ronald L. Graham, Donald E. Knuth and Oren Patashnik's *Concrete Mathematics: A Foundation for Computer Science*, Second Edition (published by Addison-Wesley Publishing Co., 1994, ISBN 0-201-55802-5). TeX provides the time of day in minutes since midnight. (We'll leave alternate implementations as an exercise for the reader.) Converting that to the traditional *hours:minutes* format requires a bit of fiddling. Usually, we use the following:

```
\def\formattedtime{\hrs = \time
  \divide \hrs by 60
  \mins = \time
  \divide \mins by 60
  \multiply \mins by -60
  \advance \mins by \time
  \number \hrs
  : \ifnum \mins < 10 0\fi \number \mins
}
```

On the other hand, we did a bit of head scratching over the following fragment from the *Concrete Mathematics*' macros before the inevitable "aha!":

```
\def\hours{\count0=\time
  \divide\count0 by60 % find the o'clock
  \multiply\count0 by40
  \advance\count0\time % convert to hhmm
  \advance\count0 10000
  \expandafter\gobbleone\number\count0\relax
}
\def\gobbleone1{}
```

The calculation of `time` divided by 60 multiplied by 40 provides 40 multiplied by the hours. Because the number of minutes since midnight already contains the hour multiplied by 60, this has the effect of leaving the hours multiplied by 100 in the result. Thus, we are left with hours multiplied by 100, plus minutes. Adding 10,000 guarantees that there is a leading zero, if necessary. Unfortunately, it's preceded by a leading one; fortunately, that character is eaten by `gobbleone` in a bit of TeX macro legerdemain.

## HTML and troff

Let's change gears now. By virtue of our being open-source bigots, we are also in favor of open formats. This means proprietary documents produced by the likes of Microsoft Word and Excel make us see various shades of red. (OK, they make Haemer see red. Copeland is color-blind, so he just sees a darker shade of gray.) It also means we really like markup languages such as `troff` and HTML—in fact, we generally write this column in the former and then convert it to the latter.

There are a number of tricks we could use for this conversion, including a variety of public-domain conversion tools. But we do something that may not be as obvious: We convert our `troff` source to HTML by running it through `nroff` with a special macro package.

This all came to mind a few weeks ago when Softway Systems colleague John McMullen was converting a variety of `troff` documentation to online Web pages and asked for some assistance. We won't show you the whole macro package, but just some interesting pieces.

Our replacement for the `-mm` list macros had been as follows:

```
.\ " ===== LISTS
.de AL \ " numbered list
.nr list_type 1
<OL>
..
.de BL \ " bullet list
.nr list_type 2
<UL>
..
.de LE
.if \n[list_type]=1 <OL>
.if \n[list_type]=2 <UL>
.nr list_type 0
..
.de LI
.if \n[list_type]=1 <LI>
.if \n[list_type]=2 <LI>
..
```

John pointed out that we didn't support nested lists, and supplied the replacement code shown in Listing 1; which you'll note actually has comments in it. (For ease of reading, `@br` is a macro that replaces the `br` directive in `troff`; `br` itself becomes a macro that produces an HTML `<BR>` tag.)

It's not possible to provide a macro to handle every event-

## Listing 1. Replacement Code

```

.\ " ===== LISTS
.\ " When we enter a new list, we prepend the
.\ " correct termination tag to the string
.\ " list_end. When we end a list, we use that
.\ " string as the argument list to the .LE
.\ " macro, print the first argument and redefine
.\ " the string If the string length is zero,
.\ " we know there's a problem.
.de AL \ " numbered list
.@br
<OL>
.ds list_end "</OL> \\[list_end]
..
.\ " we could specify bullets versus dashes
.\ " (HTML 3.2) but it's not a vital issue in my
.\ " experience, but with .AL people care.
.de BL \ " bullet list
.@br
<UL>
.ds list_end "</UL> \\[list_end]
..
.de DL \ " dash list
.BL
..
.de end_list
.ie \[n[.]]=0 \{
. tm ".LE: List ending without being in a list
.\}
.el \{
\[\$1
.shift
.rm list_end
.ds list_end "\[\$@"
.\}
..
.de LE
.@br
.end_list \\[list_end]
.if "\[\$1"1" <P>
..
.de LI
.@br
<LI>
..

```



ality in our text, so the HTML macros define the following:

```
.ds HTML@Printing xx
```

Because `groff` provides a way to test the existence of a string, `.if dHTML@Printing...`, we can provide different coding for the `troff` and HTML versions. For example,

```
.ds rr re\*'sume\*'

```

```
.if d HTML@Printing .ds rr r&eacute;sum&eacute;
```

And because most usage of the `HTML@Printing` flag is related to accents, we finally wrote an accent filter:

```

#! /usr/local/bin/perl -p
# Accent filter for -mm to HTML conversion.
# Note this only works for valid combinations.

s/([AEIOUaeiou])\\*\/&#x0026;$1uml;/g;
s/([AEIOUaeiou])\\*\/&#x0026;$1luml;/g;
s/([AEIOUaeiou])\\*\/&#x0026;$1grave;/g;
s/([AEIOUaeiou])\\*\/&#x0026;$1acute;/g;
s/([AEIOUaeiou])\\*\/&#x0026;$1circ;/g;
s/([ANOano])\\*\/&#x0026;$1tilde;/g;
s/([Cc])\\*\/&#x0026;$1cedil;/g;
s/\\(AE\/&#x0026;AElig;/g;
s/\\(ae\/&#x0026;aelig;/g;

```

This nicely converts input such as

```

U\*:ber, u\*:ber,
ha\^*t, nin\*~o,
fac\*,ade, \(\aeon

```

into

```

&Uuml;ber, &uuml;ber,
h&acirc;t, ni&ntilde;o,
fa&ccedil;ade, &aelig;on

```

for printing as

```
Über, über, hât, niño, façade, æon
```

We leave it as an exercise for the reader to fill in the other interesting `troff` special characters with HTML/8859-1 escape sequences, such as inverted exclamation points and the common fractions.

Next time, we'll write a review of I18N tricks and techniques. By the time you read that, the Microsoft trial may be in appeal, all of your off-by-one bugs may be gone and you may have finished converting your `troff` documents to HTML.

Until then, happy trails. 🐦

---

*Jeffrey Copeland* (copeland@alumni.caltech.edu) lives in Boulder, CO, and works at Softway Systems Inc. on UNIX internationalization. He spends his spare time rearing children, raising cats and being a thorn in the side of his local school board.

*Jeffrey S. Haemer* (jsh@usenix.org) works at QMS Inc. in Boulder, CO, building laser printer firmware. Before he worked for QMS, he operated his own consulting firm and did a lot of other things, like everyone else in the software industry.

Note: The software from this and past Work columns is available at <http://alumni.caltech.edu/~copeland/work>.