SOAP BOX ANNUAL REPORT

Monroy Lopez, Ivan

2009

Monroy Lopez, Ivan

Soap box annual report

‘\" t
.\"
.\" You may distribute under the terms of the GNU General Public
.\" License as specified in the file COPYING that comes with the
.\" man-db distribution.
.\"
.\" Sat Nov 27
.\"
.pc
.TH print soapbox "2009-11-28" "1.0" "print"
.SH manuals, man pages
.I according to the mmm, software has two faces, both as important. the code
.I speaks to the machine, and the documentation ``tells its story to the human
.I user.’’

The following is also from the fanzine I’m talking about. It’s a perhaps convoluted way of saying that there’s a difference between the documents that keep track of the changes in a piece of software over time, and the documents that show how a piece of software is used, even if they sometimes get lumped together. Texinfo is just another documentation format that some people seem to prefer.

.I in the gnu coding standards, the terms documentation and manual practically
.I mean the same thing. specifically, a manual is a texinfo file, and
.I documentation includes manuals and other things like NEWS files and change
.I logs. in practice, the latter term practically takes the place of the former.
.I regarding unix man pages, gnu suggests that -if adequate- help2man be used to
.I extract a man page from the texinfo file. unlike man pages, gnu manuals should
.I have a ``coherent topic’’. as an example, the coding standards note that diff
.I and diff3 are both covered in a single manual, whereas there are two man
.I pages, one for each command.

I think that it’s funny that they write implementation does not equal documentation. I guess that sometimes it’s hard to take distance:

.I the people of gnu distinguish between the way in which a program was built,
.I and the way in which it’s used. they warn about modelling the documentation
.I after the software:

.B programmers tend to carry over the structure of the program as the
.B structure for its documentation. but this structure is not necessarily
.B good for explaining how to use the program [...] learn to notice when
.B you have unthinkingly structured the documentation like the
.B implementation, stop yourself, and look for better alternatives.

.I the coding standards advise authors to approach users pedagogically, thinking
.I about ``the concepts and questions that a user will have in mind when reading

.I it.'' what's more, the manuals that they write should admit two types of .I reading: tutorial and reference. it's interesting to note that by tutorial they .I mean something that a user may want to read straight through.

in this essay, code is the textual aspect of computer technology that may be loaded up on a text editor and easily changed. i'm consciously avoiding any discussion on the subject of text editors — jot down your edits on a piece of paper and then write them to newfile with echo. what's important to me is that this be easily accomplished. with enough time and energy, anyone could write interesting code. the best project would be if my grandmother took the time to re-write the linux kernel from scratch, and if she kept a record of her reflections about code. as much as i like pierre menard, this is not feasible. the processes of code should be manageable without the need of resorting to too much external technical support. this means that code is relative. what's code for some will not be code for others.

when it comes down to it, this means that code is text written in one of the computer languages. code is the active practice of altering half-understood text files. code are the static characters that silently stare back at you, and that will not even you give the illusion that you're double-guessing a machine. it's always evident that someone else was there before you, and that that person was sloppy. it's just a matter of playing along and locating those three characters in a text file that will make all the difference on whether your computer can display postscript or whether it will keep that as a secret to itself.

my one year experience as a software writer has not been so nice, but really you should ask someone else with more experience. i've only worked a little as a freelancer (whatever that means) on some random websites. i've been learning on the job (whatever that means).

there are probably other ways of saying it, but it's also fine to say that i've been bored. i don't know why. there have been projects i'm not ashamed of, but most of the time it hasn't been like that. i wish i could say i was young i needed the money but i'm not even that young anymore. health insurance and that type of stuff is a work of fiction. my life is only as long as my computer's life.

it's depressing to see that some cliches are true. the offices of lab rats typing looking for the key of the cookie, waiting to be rewarded with a cookie, and kept awake with coffee. i could have passed out in front of the computer and no one would have noticed. the feeling that no one knows what they're doing. the obscure files that you can find inside of some computers. what i would do is open them for as long as i'd stand it, close my eyes, press some random keys, close them and try to forget about it. there's determinacy in software insofar as you can work like this and nothing breaks.

i wish i could live in a mountain and contemplate my mark-up. i wish it was, but it can't be and it isn't only about the code. if it was only about the code, i wouldn't need to go out of my room. i don't know what it's about, and i don't know what i'm talking about. this is the point where i start to have problems finishing my sentences. likewise, this type of media work is complete nonsense. it's absurd that i can't find a job, and that i can't hold on to the jobs that i hate.

after one year of writing software, it's good to go back to the first things i wrote. maybe there is something of crawling into my bed and pulling up the covers about it. probably not. i'm just happy. it was great to have imagined

an introductory perl manual where the examples and exercises deal with generative groff mark-up.