

Upozornení: Tento příběh slouží k vysvětlení podstaty systému maker v Lispu. Příběh je pouhou fikcí a nikdy se nestal.

The Story of Junior Programmer Mac

Once upon a time, long ago, there was a company of Lisp programmers. It was so long ago, in fact, that Lisp had no macros. Anything that couldn't be defined with a function or done with a special operator had to be written in full every time, which was rather a drag. Unfortunately, the programmers in this company—though brilliant—were also quite lazy. Often in the middle of their programs—when the tedium of writing a bunch of code got to be too much—they would instead write a note describing the code they needed to write at that place in the program. Even more unfortunately, because they were lazy, the programmers also hated to go back and actually write the code described by the notes. Soon the company had a big stack of programs that nobody could run because they were full of notes about code that still needed to be written.

In desperation, the big bosses hired a junior programmer, Mac, whose job was to find the notes, write the required code, and insert it into the program in place of the notes. Mac never ran the programs—they weren't done yet, of course, so he couldn't. But even if they had been completed, Mac wouldn't have known what inputs to feed them. So he just wrote his code based on the contents of the notes and sent it back to the original programmer.

With Mac's help, all the programs were soon completed, and the company made a ton of money selling them—so much money that the company could double the size of its programming staff. But for some reason no one thought to hire anyone to help Mac; soon he was single-handedly assisting several dozen programmers. To avoid spending all his time searching for notes in source code, Mac made a small modification to the compiler the programmers used. Thereafter, whenever the compiler hit a note, it would e-mail him the note and wait for him to e-mail back the replacement code. Unfortunately, even with this change, Mac had a hard time keeping up with the programmers. He worked as carefully as he could, but sometimes—especially when the notes weren't clear—he would make mistakes.

The programmers noticed, however, that the more precisely they wrote their notes, the more likely it was that Mac would send back correct code. One day, one of the programmers, having a hard time describing in words the code he wanted, included in one of his notes a Lisp program that would generate the code he wanted. That was fine by Mac; he just ran the program and sent the result to the compiler.

The next innovation came when a programmer put a note at the top of one of his programs containing a function definition and a comment that said, "Mac, don't write any code here, but keep this function for later; I'm going to use it in some of my other notes." Other notes in the same program said things such as, "Mac, replace this note with the result of running that other function with the symbols x and y as arguments."

This technique caught on so quickly that within a few days, most programs contained dozens of notes defining functions that were only used by code in other notes. To make it easy for Mac to pick out the notes containing only definitions that didn't require any immediate response, the programmers tagged them with the standard preface: "Definition for Mac, Read Only." This—as the programmers were still quite lazy—was quickly shortened to "DEF. MAC. R/O" and then "DEFMACRO."

Pretty soon, there was no actual English left in the notes for Mac. All he did all day was read and respond to e-mails from the compiler containing DEFMACRO notes and calls

to the functions defined in the DEFMACROs. Since the Lisp programs in the notes did all the real work, keeping up with the e-mails was no problem. Mac suddenly had a lot of time on his hands and would sit in his office daydreaming about white-sand beaches, clear blue ocean water, and drinks with little paper umbrellas in them.

Several months later the programmers realized nobody had seen Mac for quite some time. When they went to his office, they found a thin layer of dust over everything, a desk littered with travel brochures for various tropical locations, and the computer off. But the compiler still worked—how could it be? It turned out Mac had made one last change to the compiler: instead of e-mailing notes to Mac, the compiler now saved the functions defined by DEFMACRO notes and ran them when called for by the other notes. The programmers decided there was no reason to tell the big bosses Mac wasn't coming to the office anymore. So to this day, Mac draws a salary and from time to time sends the programmers a postcard from one tropical locale or another.