Loading

@ IT > Coding Edge > In 'real macro', cut the number of lines of C in half!

ツイート   0   ☐

【LLTV Report】 Dramatic Before After, Craftsmanship (middle part)

In 'real macro', cut the number of lines of C in half!

2009/09/07

"LL event" with the theme of lightweight programming language (LL: Lightweight Language) held every summer. The seventh "LLTV" was held in Tokyo and Nakano on August 29, 2009. In this article we report part of the program, "Great Improvement !! Dramatic Before After". In the first part , I reported the introduction of neta series "extension" of fortune command by Ruby and implementation of sl command by Firefox extension. In this article which becomes the middle part, we report Lisp style macros in C language and report on the skills of the artist who "repairs" the source code of the ls command to about half. In the second part , we will report on the dramatic before-after announcement that the sales management system that tends to be stagnant at the sales department was "rebuilt" with the bash command.

Problems with source code of ls written in C

The presentation entitled "Real / Macro Metaprogramming On C" was made by Mr. Shiro Kawai, known as the developer of Scheme processing system "Gauche" (Gauche). It is a "takumi" known as a super hacker covering academic informatics discussions, implementation talks of muddy hackers leading to assembler level optimization.

Mr. Kawai tried renovation using the ls command (not GNU's, but attached to FreeBSD). The source code is 1141 lines. Mr.
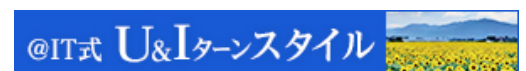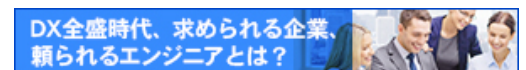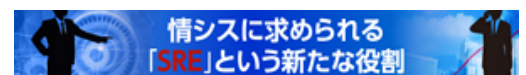
Kawai says, "It is a very straightforward and easy-to-read code, it is superior as a source code written in C", but "From the perspective of programmers who have been spoiled by using LL, you will find discontent by all means" , Pointing out poor redundancy and maintenance.
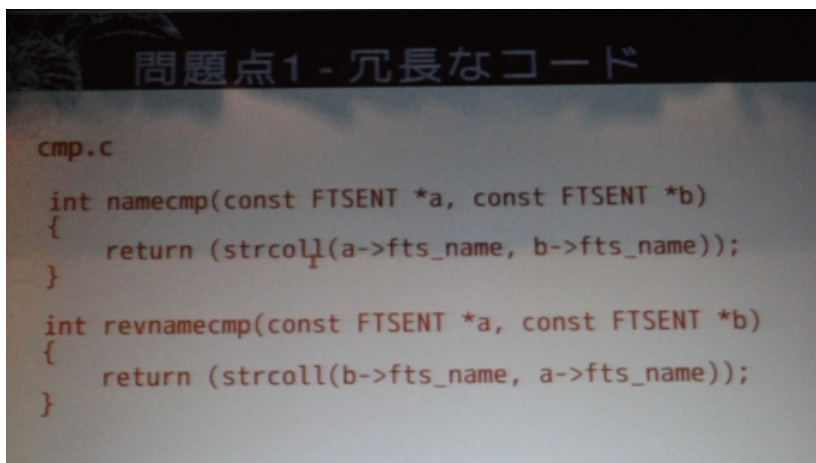
Master Takashi Kawai

For example, when displaying the file list with the ls command, you can sort in ascending / descending order. There are various things to compare by sorting, such as file names and timestamps. At this time, as comparison function passed by the function pointer, it is said that two types of comparison objects, ascending and descending, are defined in cmp.c in pairs. They have exactly the same structure except function name and operation are opposite, and there is waste "Mr. Kawai" (Mr. Kawai) "When you want to add a new sort order, you have to write two each." "If you can use LL's higher-order functions, you can create functions on the fly that act the opposite way, with functions as arguments, but I can not do that in C language." This is the limitation of C language which has poor abstraction function to create patterns with similar structure.


```
cmp.c
 int namecmp(const FTSENT *a, const FTSENT *b)
 {
     return (strcoll(a->fts_name, b->fts_name));
 }
 int revnamecmp(const FTSENT *a, const FTSENT *b)
 {
     return (strcoll(b->fts_name, a->fts_name));
 }
```

One of the problems Mr. Kawai indicated. Because C language abstraction function is poor, two similar functions are defined s eparately

Likewise, it is a problem seen in the C source code that non-essential information goes into the code. For such a loop that accesses an array using a counter in a for loop, or a loop that sequentially searches a linked list with a pointer, it is possible to describe the target as intended more strictly if the language has a high degree of abstraction. Mr. Kawai pointed out the

weakness of the abstract function of C language by showing pseudo code "I wish I could write like this".
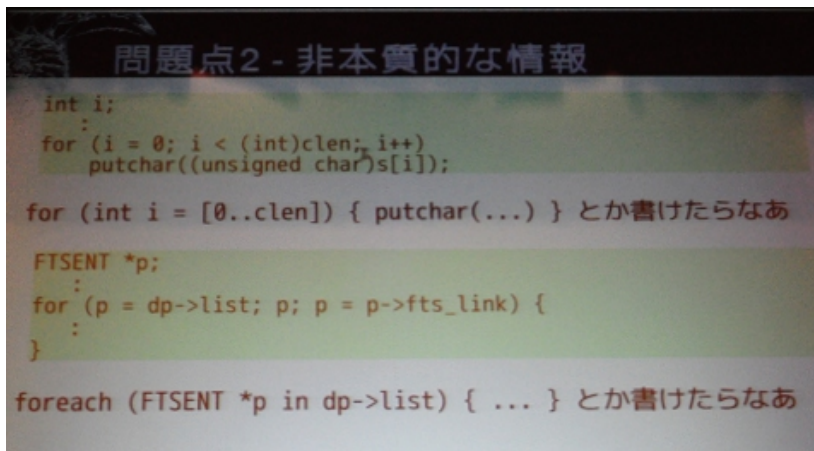
Weakness of C can not be abstracted of loop

The third problem pointed out by Mr. Kawai is that it is easy for information to be distributed in code. ls manages many options with flags. However, the code related to the flag is related to the header file, the place where the variable is declared, and the option parser of the main function, and "There are at least three modifications to add one option Needed ", pointing out the poor maintainability due to the poor abstract function.
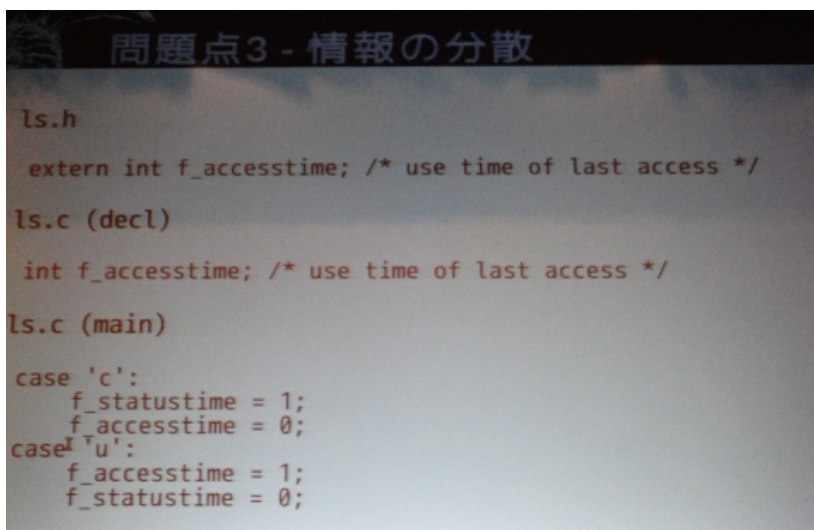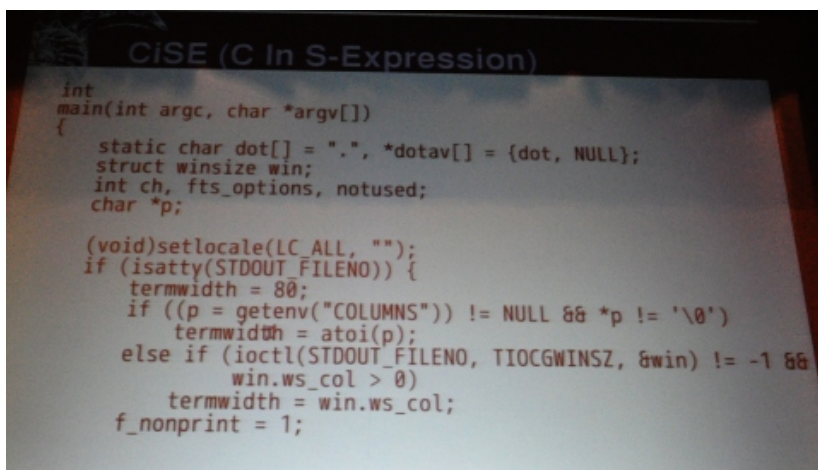

Code related to the option switch is dispersed in three places: h eader, variable declaration, option parser

Drastically renovated with genuine macros!

Source code of ls written in C. The craftsmanship put forward against the three problems "redundancy", "nonessential information", "information dispersion" is "real macro". Here, Mr. Kawai's genuine macro is not a macro like a text substitution which is processed by the C preprocessor, but a Lisp-like "macro that can convert arbitrary source code before compiling".

"The (syntax of the programming language) syntax is decorative, a great person may not know, but I guess everyone here knows it." When describing the point of renovation like this, Mr. Kawai oddly transformed the code fragment written in C into a different syntax one after another. If you replace it with a Lisp S expression of parentheses as a print statement as an example of a function call, an if statement as a control structure, "This is the same thing even if it writes like this", it is expected to unexpectedly (or as expected) A laughter blew up from her.

Tampering is a code example of C's main function. "For example, you can write such a main function like this", the common source code is cropped in its original shape, but it is converted into the source code of a mysterious face that no one has seen While being able to show off, the venue was overwhelmed by the craft 's "rough skill" and was enveloped in a big laugh. Takumi says, "Looks change only a bit, if you get used to it you will be able to read the slaslers", it's a story.
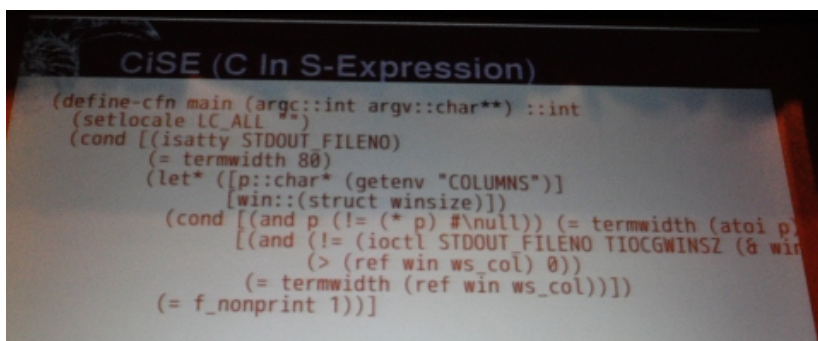

Typical code example of C indicated by Mr. Kawai


An example in which the above C code is rewritten in a form called CiSE. It says writing C in S formula

Magical-like code conversion

Although the venue was invited to laugh at the source code of the S formula, this is not a "magnificent neta" or the like, it is a serious effort that Mr. Kawai named "CiSE" (C In S - Expression) . With CiSE, you can use macros that manipulate

abstract syntax trees to generate code, without changing the preferred features such as C's ABI, libraries, semantics close to hardware. In Gauche's implementation of the Scheme processing system that emphasized practicality, it says that CiSE is used in part.

The first example shown by Mr. Kawai as an indication of the power of a macro is abstraction of a pattern that accesses array elements by incrementing the counter from 0 in a for loop. In processing such as turning a for loop according to the length of a character string, low-level processing such as non-essential temporary variables are lost from the code and initialized with 0 by a macro named "dotimes" Can also be concealed. In CiSE, it is said that temporary variable on C etc. are automatically managed by serial number, so that name collision does not occur.
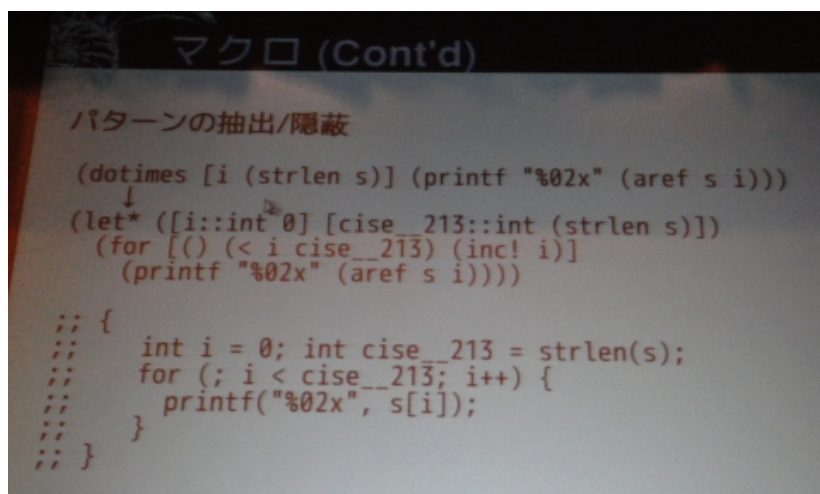

An example of using a macro to abstract a loop. The code on the top is the original. The middle is the expression expanded by macro, converting it to C language can generate the code at the bottom

In addition to this, by using macros, it is possible to eliminate repetitive patterns, organize redundant codes by combining essential processing, generate code in corresponding places on multiple files with declarative syntax like DSL, Mr. Kawai showed off abstraction techniques such as gathering the information that I was doing one after another (During this time, the dramatic Before after BGM in the hall). By clearly organizing the original functions of C, we succeeded in reducing the total number of lines of code to about half. Not only was the source code shortened, it became easier to add the same line of functions, it also added merits such as increased pattern reusability.

If you really did such a refurbishment, there is no doubt as to whether the residents of C who will maintain the ls command

will live, but there is no doubt that it is a dramatic before-after.
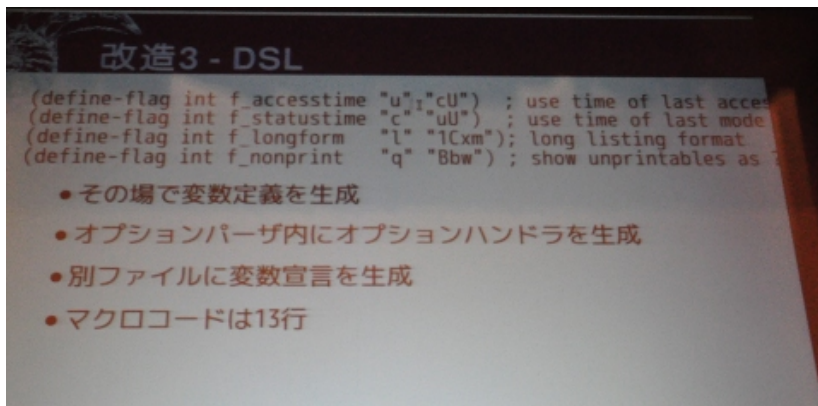
Code related to option switch processing was distributed to three places in the original C, but as for each option to be enough for one line declaration by DSL using macro

As a result of rewriting the source code of the ls command, about half the number of lines!

> **Let's use the language " " instead of being used in the language "**

Using macros generally reduces the number of lines of source code to a fraction. "The bigger the source code, the bigger the effect," Mr. Kawai said.

Mr. Kawai, who showed the power of the macro, said, "That is not to say that we should rewrite our program with macros as soon as we return today," he said. Because "Macro is a powerful drug" (Mr. Kawai). From the book "Programming Clojure" (written by Stuart Halloway) which is currently being translated Mr. Kawai quotes the phrase about the macro (Clojure is the implementation of the Lisp dialect on the JVM). It is said that there are two rules and one exception in the "macro club".

- Rule 1: Do not write macros.
- Rule 2: If it's the only way to encapsulate a pattern, write a macro
- Exceptions: If callers are easier to compare than equivalent functions, you can write macros

There are many hackers that support the power of S-expressions and macros, but it is hard to say that it is a

technique widely used in general software development. Mr. Kawai mentioned at the end as "message of the artist" as follows, including such circumstances.

"As a programmer, you are a versatile god to computers, so let 's play with the language by using the language" instead of being used in languages in honor of language restrictions. "
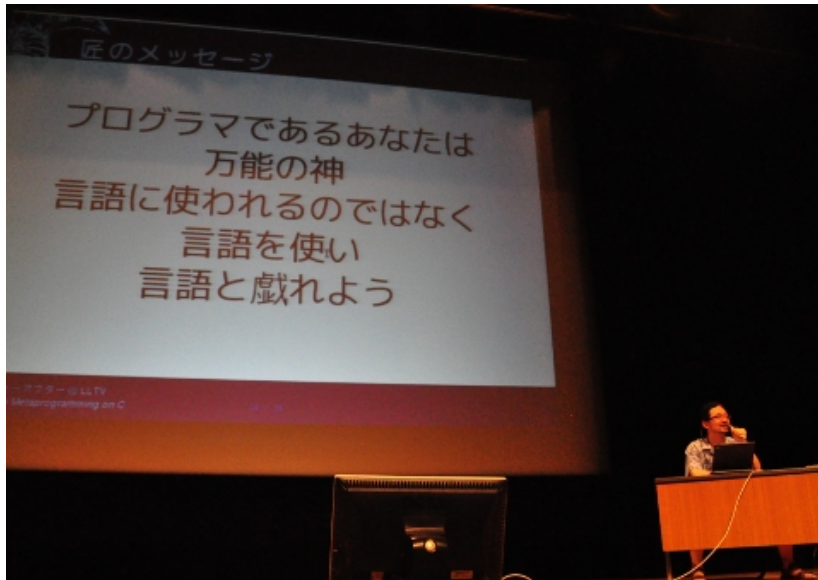
" LLTV Report Dramatic Before After, Craftsmanship (Part 2) " dramatically improves numerous business systems that were siloized and difficult to collaborate using the UNIX approach of combining bash scripts, flat text, and multistage pipe processing Report the announcement of Mr. Hiroyoshi Otomo of the USP laboratory which has renovated.

Related Links

Lightweight Language Television 2009 (LLTV)

Related article

A festival of lightweight language is held from Ruby 2.0 to my language (@ ITNews)

"Why did you make it", those who made my own language (@ ITNews)

Japan's first hacker, Mr. Wada's "Hacker temperament" (@ ITNews)

(@ IT Nishimura Ken)
Please give us information: tokuho@ml.itmedia.co.jp

TechTarget

Slackで本当に大丈夫? 大規模障害で高まる信頼性への懸念

「iOS 11」のApple純正カメラアプリで見つかった脆弱性 悪用例と対策を解説

クラウドやサーバレスが浸透、IT運用チームが実践すべきセキュリティ対策は?

「Mirai」亜種、「Wicked」ボットネットはどのようにIoTデバイスを狙うのか

Coding Edge new article

How to execute programs is searched **from the Linux kernel co de** (2017/7/20)
A series that explores the contents of the "printf ()" "main ()" funct ion used in the C language "Hello World!" Program from various a spects such as analysis and disassembling by a debugger, source co de reading and so on. In the last round, in the Linux kernel, we will explore what kind of processing is done at program startup

If you are an engineer you know what exit () is called at the en d of the C language program? (July 13, 2017)
A series that explores the contents of the "printf ()" "main ()" funct ion used in the C language "Hello World!" Program from various a spects such as analysis and disassembling by a debugger, source co de reading and so on. In this issue, we will explore the contents of exit () called at the end of the program

Basics of FileDialog operation in VBA & How to get free space of drive, size of file and time stamp (2017/7/10)
How to acquire the free space of the specified drive, file time stam p and attribute, and the basics of FileDialog / explorer operation

Farewell overtime! It is troublesome What is "Excel VBA" to e ase excel work? (July 6, 2017)
Daily occurrence "troublesome work". There is a possibility that ef ficiency can be achieved with simple programming. In this article, we introduce the VBA that can be used in "Microsoft Excel" whic h is often used in business. * Shortcut key, explanation of access k ey

@ IT TOP | Coding Edge Forum Top | Conference Room | Terms of Use | Privacy Policy | Site Map