

## An algorithm for indentation processing of program style

Zhenyu Wang

Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

ruangongwwk@163.com

---

### Abstract

Aiming at the problem that students are in poor alignment when programming on the computer, an algorithm for making indentation of C program code is proposed in this paper. First, the C program code that has not been processed by the indentation is acted as the input, and static analysis is performed to clarify the basic structure of the program. Then, the code is stored in units of carriage return characters by the algorithm and the count of the number of braces is calculated. Finally, the code after the indentation is output. This algorithm is written with JAVA language in this paper, and a C-code for specific problem is experimentalized on eclipse. The code is indented by this algorithm and the sorted code is output. The experimental results show that the algorithm can perform the indentation processing on C programs, which is helpful for cultivating students' good programming style.

### Keywords

Program indentation;Static analysis;Programming style.

---

## 1. Introduction

With the improvement of the "Internet and education" era, the application of Internet technology to teaching practice has become a research hotspot. Programming ability is a basic skill of computer science students, and the degree of mastery and goodness has an important influence on the study and employment of later professional courses[1]. The programming course is responsible for the training of programming skills. Whether students go to work in the enterprise or continue to pursue a master's degree, the programming ability will have a profound impact on their future development. However, in actual teaching, both teachers and students focus on the algorithm design and execution accuracy of the program, the problem of programming style is ignored by them. In large software development, if programmers fail to develop a good programming style, the poor program readability will caused by them, and the increase maintenance costs and workload will have a bad impact on the software[2-3]. In particular, poor indentation habits directly lead to program errors in Python programming. It is necessary for the teachers of the programming course to impart the basic knowledge of the programming style to the students at the beginning of the student learning program design, so that the students pay attention to the style problems during programming, and constantly review and improve the programming style in the programming practice, and gradually develop a good programming habits. The most effective way to cultivate students' good programming style is trying to guide students to write a mounting number of program code and emphasis them the importance of the programming style. This requires teachers to timely and accurately check the problems in the student's programming style and give students appropriate guidance and advice.

Many scholars have done research on programming style. In paper[4], several programming teachers at the University of Pittsburgh have summarized a set of programming style guidelines including the layout of the indentation and code after several years of teaching experience. For pascal programming

style research, an evaluating Pascal programs software designed by Tom Schorsch of American is used to check style errors. In paper [5], a good programming style rule for annotations, codes, indentations and so on is proposed by the author. In addition, the online program evaluation system used in the ACM annual organization competition, which is a common dynamic test-based program evaluation system, has strict restrictions on the submitted code, and has high efficiency on program running time, memory usage, and execution efficiency requirements.

Based on the research, an algorithm for the indentation of C program code is designed in this paper, which is written in JAVA language. For a C program with poor indentation, after the algorithm is typeset, a good indentation procedure can be obtained.

## 2. Preliminaries

Good programming style is very important for the readability and maintenance of the program. As shown in Figure 1, the left side is the C block with poor indentation, and the right side is the block after the indentation.

<pre>void main(void) { int a; scanf("%d",&amp;a); if (a&gt;0) printf ("%d is a positive number",a); else printf ("%d is a negative number",a);}</pre>	<pre>void main(void) {     int a;     scanf ("%d",&amp;a);     if (a&gt;0)         printf ("%d is a positive number",a);     else         printf ("%d is a negative number",a); }</pre>
---	---

Fig.1 C code comparison

This is a simple C program that judges positive and negative numbers. Obviously, the code alignment on the left side is poor in indentation, and the basic structure of the program cannot be clarified. Once the number of nested layers of the compound statement is large, the readability will become very bad. The program on the right has a good alignment and a clear structure. Even with complicated programs, the structure level is very obvious. Some good programming style rules are given below.

(1)The indentation style adopts "hanging" Allman style, which means "{" and "}" are aligned up and down. It is visually easy to match, and easier for beginners to debug the program. This is also the indentation mode adopted by the windows system. The following is an Allman indented code:

```
#include <stdio.h>
void main()
{
    int n,sum=0;
    n=1;
    while (n<=200)
    {
        sum=sum+n;
        n++;
    }
}
```

(2)For if, for, while, switch and other statements, in the alignment, according to the principle of windows programming style, the left and right curly braces should each occupy a row and a column, which is easy to read.

(3)One line of code only does one thing, which means only one statement per line. The advantage of this is that it is easier to find syntax errors when the compiler indicates that a line has an error with a line number. If there are multiple statements in a row, the line number does not indicate which statement has an error.For instance,the code below is better than the code ‘int a,b,x;x=a+b;’

```
int a,b,x;
x=a+b;
```

(4)Annotations are a bridge between programmers and future readers. Proper annotations can help the reader understand the program and provide clear guidance for testing and maintenance at a later stage. It must be noted that comments are by no means optional.For one thing, annotations should be accurate, understandable, and prevent ambiguity. Wrong annotations are not only unhelpful but harmful.For another, when the code is long, especially when it is multiple nested, it should be annotated at the end of some paragraphs for easy reading. The proper annotation is shown in Fig.2

<pre>//introduction of the function:change two value //input papameter:p1,p2 //output parameter: //return value  void swap(int* p1,int* p2) {     int t;     a=*p1;     *p1=*p2;//change *p1 and *p2     *p2=a; }</pre>	<pre>//main function void main() {     int m,n;     printf ("Input m,n:");     scanf ("%d %d",&amp;m,&amp;n);     /*actual parametrer is the address of m,n*/     swap(&amp;m,&amp;n);     printf ("sorted:%d %d\n",m,n) }</pre>
---	--

Fig.2 The proper annotation

(5)C operators and expressions are flexible, and many problems that are difficult to handle in other languages can be subtly handled with C langugae. However, C language do not specify the order of evaluation of subexpressions in expressions, such as (i+ +)+(i+ +)+(i+ +) expression is prone to ambiguity or is incomprehensible to the reader of the program. But if is written as ‘a= i+ + ; b= i+ + ; c= i+ + ; d= a+ b+ c;’, it is easy to understand, and it will not be different due to different systems.

(6)Blank lines can make the program code naturally divided into multiple paragraphs, each of which represents the corresponding meaning. A C program is a function of a unit, each function represents a module, so different functions should be blank and add annotation.The Blank line is showed in Fig.3.

```
void fun1()
{
    ...
}
//blank line
void fun2()
{
    ...
}
//blank line
...
```

Fig.3 blank line

### 3. Algorithm Design

This section designs an algorithm for the indentation processing of C programs for the indentation rules described above. The basic flow of the algorithm is shown in Fig.4.

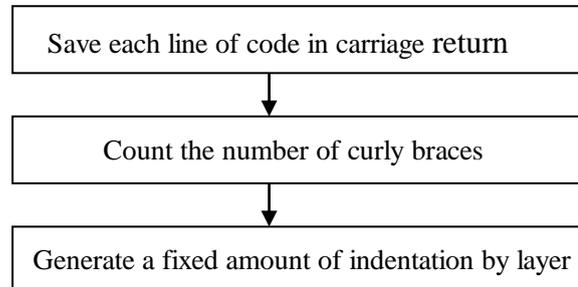


Fig.4 basic flow

The algorithm is described with JAVA language below:

```

private String RearrangeCode(String code) //RearrangeCode is used to indent the code
{
    int begin=-1,end,i,j,k=0,rowCount=0,braceNum=0;
    String space,tmp;
    String[] message;//message is used to save the code by row
    for (i=0;i<code.length();i++)
        if (code.charAt(i) == '\n')
            rowCount++;
    message=new String[rowCount];
    for (i=0;i<code.length()-1;i++)
        if (i == 0 || code.charAt(i) == '\n')// Find the location of the first carriage return
        {
            for (j=i+1;code.charAt(j) != '\n';j++)// Find the location of the last carriage return
                ;
            end=j;
            message[k]=code.substring(begin+1, end);
            k++;
            begin=end;
            i=end-1;
        }

    for (i=0;i<message.length;i++)
    {
        tmp=message[i];// Keep a copy of the current information
        if (tmp.equals("{}"))
            braceNum--;
        space="";
        for (j=1;j<=4*braceNum;j++)
            space=space+" ";// Generate a fixed number of indents
        message[i]=space+message[i];// Increase space indentation
        if (tmp.equals("{"))
            braceNum++;
    }
    code="";
  
```

```
for (i=0;i<message.length;i++)
    code=code+message[i)+"\n";
return code;
}
```

The main reasons for describing the algorithm with JAVA are: (1)JAVA supports dynamic array allocation, which is convenient for flexible processing; (2)JAVA comes with many system functions, such as substring, which can save code compared with C program.

#### 4. Experiment

This section tests the algorithm on eclipse and reads a piece of code that has not been processed by the indentation. It should be noted that the syntax in the code is not considered, so the statements in the code use s1, s2.....,instead. This test reads a while loop code that has not been processed by the indentation. After the algorithm is processed, the resulting indented code is shown in Fig.5 and Fig.6.

```
Code without indentation:
void main()
{
s1
s2
while(exp1)
{
s3
s4
s5
}
s6
s7
s8
}
```

Fig.5 code without indentation

```
Code with indentation:
void main()
{
    s1
    s2
    while(exp1)
    {
        s3
        s4
        s5
    }
    s6
    s7
    s8
}
```

Fig.6 code with indentation

It can be found from the experimental results that the original C program code that has not been indented has become a code that is convenient reading,clear hierarchy and frame correctly after

sending to the algorithm designed in this paper. For the existing program evaluation system, there is often no programming style detection module, but it is only for the correctness and time limit of the program. If the algorithm is applied to existing program evaluation software, the function aspect can also be expanded. After all, the correctness of the program is only one of them. Good programming style is also an important point of attention for students in programming learning.

## 5. Conclusion

This paper designs an algorithm for the indentation processing of C programs. For the C program code that has not been processed for indentation, saving each line of code by the carriage return, and counting the number of curly braces, and generating a fixed amount of indentation to achieve the finishing and layout of the code. The proposed algorithm is tested through eclipse and obtains the result is obtained. The algorithm is of great significance for reducing the burden of manual evaluation, realizing automatic evaluation of programs, and cultivating students' good programming style. For teachers, it is also possible to reduce the amount of evaluation and focus on deeper teaching. In today's information age, the algorithm can improve teaching efficiency and reduce the burden of teaching. Teachers can also use the system to evaluate the programming style of the program code submitted by students, and analyze the more concentrated programming style problems in student homework, and summarize and teach in a targeted programming style. In the follow-up research, the following points will be analyzed in depth, (1) Correcting the students' bad programming style procedures, which means prompting the students where the programming style is inappropriate; (2) Comparing the algorithm with the program evaluation technique. In combination, the programming style is also used as a scoring point; (3) Realizing the evaluation and processing of programming styles for various programming languages such as JAVA and Python. (4) Combined with the knowledge map, the algorithm and software are applied to the students' ordinary learning process. The students listen to the teacher in the classroom, and at the same time, they can learn through the software when out of the class. If they encounter errors, they can be corrected in time. Knowledge can also be continuously consolidated and reviewed.

## Acknowledgements

The Graduate Innovation Fund in Xi'a University of Posts and Telecommunications under grant(102-602080002).

## References

- [1] Wenguang He, Ke Zhou, Gangqiang Xiong. Experimental Teaching Reform and Practice in Programming Courses [J]. Laboratory research and exploration, 2016, 35(6):163-165.
- [2] Schorsch T . Cap: An Automated Self-Assessment Tool To Check Pascal Programs For Syntax, Logic And Style Errors[J]. ACM SIGCSE Bulletin, 2002, 27(1).
- [3] Junze Han, Liangde Zhang, Dongsheng Liu. Design and Implementation of C/C++ Programming Style Annotation System[J]. Journal of Inner Mongolia University (Natural Science Edition), 2010, 41(6).
- [4] Poole B J , Meyer T S . Implementing a set of guidelines for CS majors in the production of program code[J]. ACM SIGCSE Bulletin, 1996, 28(2):43-48.
- [5] Zhibin Ren, Longfei Liu. Discussion on the Writing Style of C++/C Program Source Code[J]. Journal of Weinan normal university, 2002, 7.