C语言程序设计大作业

代码缩进格式化工具

单 位： 信息工程学院

班 级：计算机科学与技术一班

学 号： 2016551103

姓 名： 芮安

任课教师： 谢勇

湘 潭 大 学

2017年2月

1. 题目概述
2. 概述

用C语言编写程序实现对C语言源程序文件进行代码缩进格式化；采用命令行方式；参数为带格式化的C语言源程序文件；支持多种代码风格的格式化；

1. 所要完成的任务说明

代码缩进格式化工具

|  |
| --- |
| 任务 |
| 采用命令行方式，参数为带格式化的C语言源程序文件 |
| 支持多种代码风格的格式化（风格种类具体见 <http://en.wikipedia.org/wiki/Indent_style> ） |
| 输出格式化后的文件 |

1. 完成情况

|  |  |
| --- | --- |
| 任务 | 完成情况说明 |
| 采用命令行方式，参数为带格式化的C语言源程序文件 | 命令格式实现：  Indent 源程序文件 参数 格式化后文件  通过main()函数参数的形式传入 |
| 支持多种代码风格的格式化 | 支持格式化为一下几种风格：   1. K&R and variants:   1TBS, Stroustrup, Linux kernel, BSD KNF   1. Allman 2. GNU 3. Whitesmiths 4. Horstmann 5. Pico 6. Ratliff 7. Lisp |
| 输出格式化后的文件 | 输出的文件名在命令使用过程中自己命名 |

1. 开发环境说明

OS: Ubuntu 16.04 LTS

编译器 gcc (version 5.4.0 20160609)

1. 程序分析与设计
2. 系统分析

创建函数

(1)del 将源程序文件格式化为单行

int del(int argc , char \*argv[])

argc,argv用于获取命令行源代码文件名

(2)head 将在del函数中生成的 ”ing.c” 文件中先格式化头文件和宏定义，并且生成新文件 ”h\_ing.c”

int head() 不带参数

(3)以下8个函数将 ”h\_ing.c” 格式化为与函数名对应风格的程序文件

KR(int argc, char \*argv[])

Allman(int argc, char \*argv[])

GNU(int argc, char \*argv[])

Whitesmiths(int argc, char \*argv[])

Horstmann(int argc, char \*argv[])

Pico(int argc, char \*argv[])

Ratliff(int argc, char \*argv[])

Lisp(int argc, char \*argv[])

argc,argv用于获取命令行生成新文件名

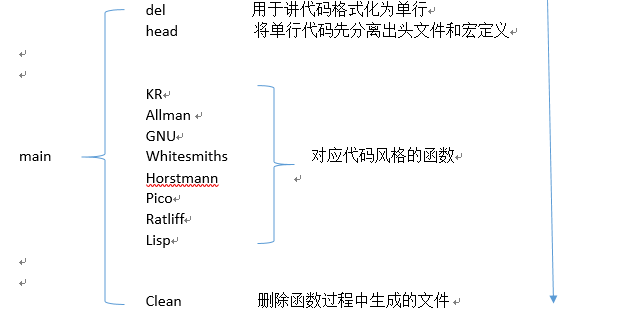
(4)clean 删除当前目录词下的 “ing.c” 和 ”h\_ing.c” 文件

void clean()

不带参数

1. 函数设计

总框架



函数解析

**del(int argc, char \*argv[])**

部分代码

int flag**=**0**; //用flag作为计数器来确定缩进的个数**

int jd\_flag**=**0**;** **//用于判断 '{' , '}' , ';' 是否在括号内**

**while((**ch**=**getc**(**fp**))** **!=** EOF**){ //对应流程图中的**2

**if(**ch**==**'('**)**

jd\_flag**=**1**;**

**else** **if(**ch**==**')' **&&** jd\_flag**==**1**)**

jd\_flag**=**0**;**

**if(** **(** **(**ch**==**' '**)** **||** **(**ch**==**'\n'**)** **||** **(**ch**==**'\r'**)** **)** **&&** jd\_flag**==**0 **){ //流程图中的**3

flag**=**0**;**

**continue;**

**}**

**else** **if(**ch**==**' ' **&&** jd\_flag**==**0**){**

flag**++;**

**if(**flag**>**1**){**

**continue;**

**}**

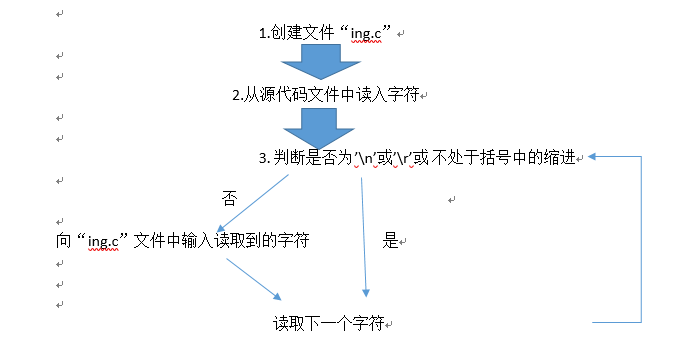
**}**

**else**

flag**=**0**;**

fputc**(**ch**,**fq**);**

**}**



**head()**

部分代码：

int flag**=**0**;**

**while((**ch**=**getc**(**fp**))** **!=** EOF**){**//2

**if(**ch**==**'#'**){**//3

flag**++;**

**if(**flag**>**1**){**

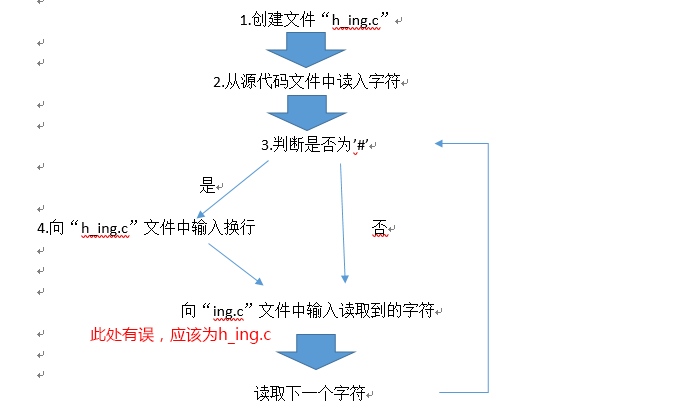
fputs**(**"\n"**,**fq**);**//4

**}**

**}**

fputc**(**ch**,**fq**);**

**}**



**各种风格函数，大致相同，以Allman为例**

**Allman(int argc, char \*argv[])**

部分代码：

//以"{","}"字符为标记

int flag**=**0**;** //用flag作为计数器来确定缩进的个数

int jd\_flag**=**0**;** //用于判断 '{' , '}' , ';' 是否在括号内

**while((**ch**=**getc**(**fp**))** **!=** EOF**){** //从文件中读入字符

**if(**ch**==**'('**)**

jd\_flag**=**1**;**

**else** **if(**ch**==**')' **&&** jd\_flag**==**1**)**

jd\_flag**=**0**;**

**if(**ch**==**'{' **&&** jd\_flag**==**0**){** //1

flag**++;**

**if(**flag**>**0**){**

fputs**(**"\n"**,**fq**);**

**for(**i**=**0**;**i**<**flag**-**1**;**i**++){**

fputs**(**" "**,**fq**);** //8

**}**

**}**

fputc**(**ch**,**fq**);**

fputs**(**"\n"**,**fq**);**

**for(**i**=**0**;**i**<**flag**;**i**++){**

fputs**(**" "**,**fq**);** //8

**}**

**continue;**

**}**

**else** **if(**ch**==**'}' **&&** jd\_flag**==**0**){** //2

flag**--;**

fputs**(**"\n"**,**fq**);**

**for(**i**=**0**;**i**<**flag**;**i**++){**

fputs**(**" "**,**fq**);**

**}**

fputc**(**ch**,**fq**);**

fputs**(**"\n"**,**fq**);**

**for(**i**=**0**;**i**<**flag**;**i**++){**

fputs**(**" "**,**fq**);** //8

**}**

**continue;**

**}**

**else** **if(**ch**==**';' **&&** jd\_flag**==**0**){** //3

fputc**(**';'**,**fq**);**

fputs**(**"\n"**,**fq**);**

**for(**i**=**0**;**i**<**flag**;**i**++){**

fputs**(**" "**,**fq**);**

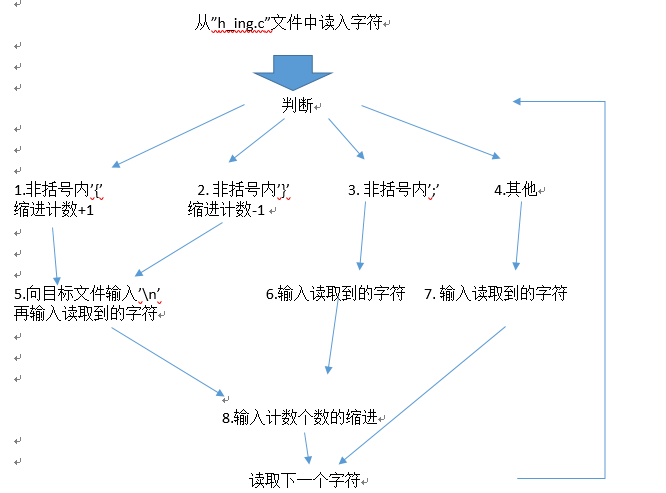
**}**

**continue;**

**}**

fputc**(**ch**,**fq**);** //4

**}**



Ps**：在Lisp和Pico 函数中，为实现向文件输出退格采用了fseek(fq,-1,SEEK\_CUR)的方式**

**clean()** 调用remove函数，删除”ing.c”和”h\_ing.c” 文件

void clean**(){**

remove**(**"ing.c"**);**

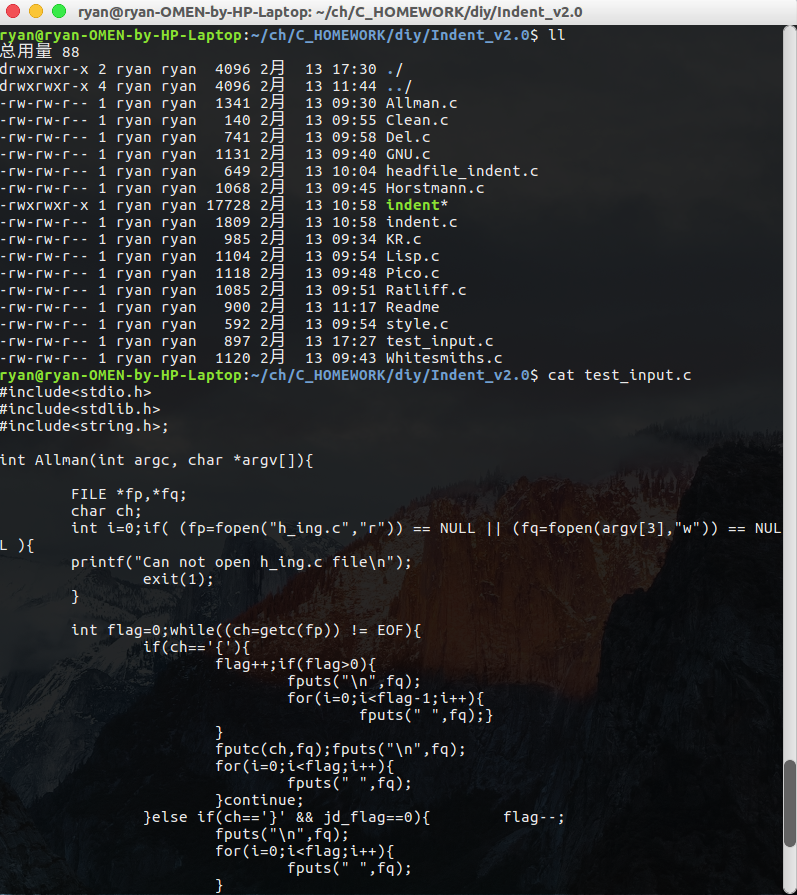
remove**(**"h\_ing.c"**);**

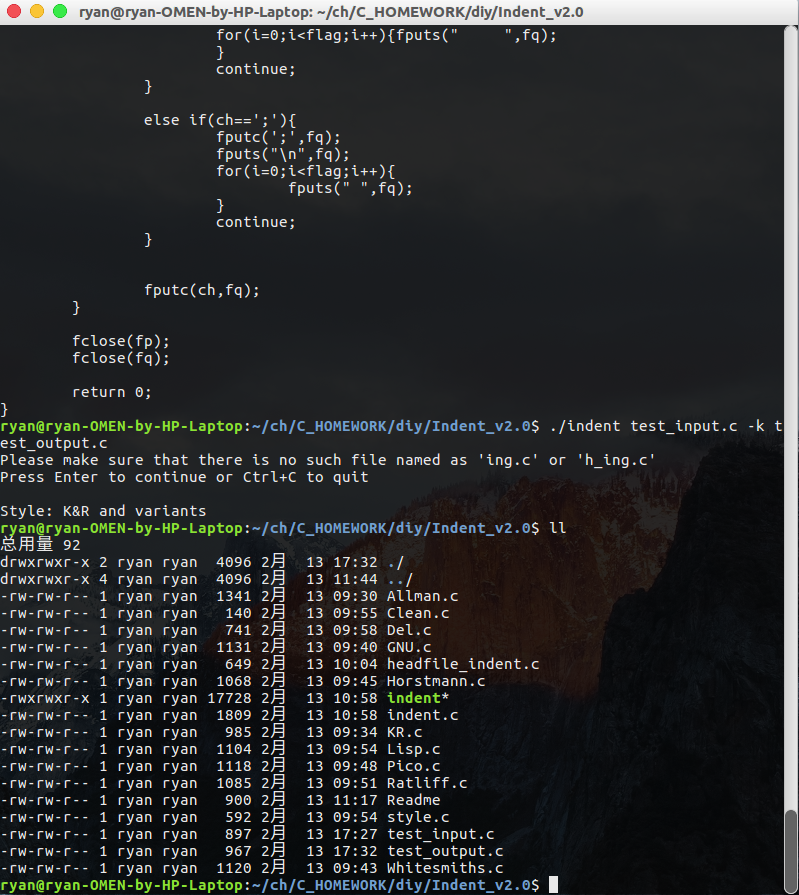
**}**

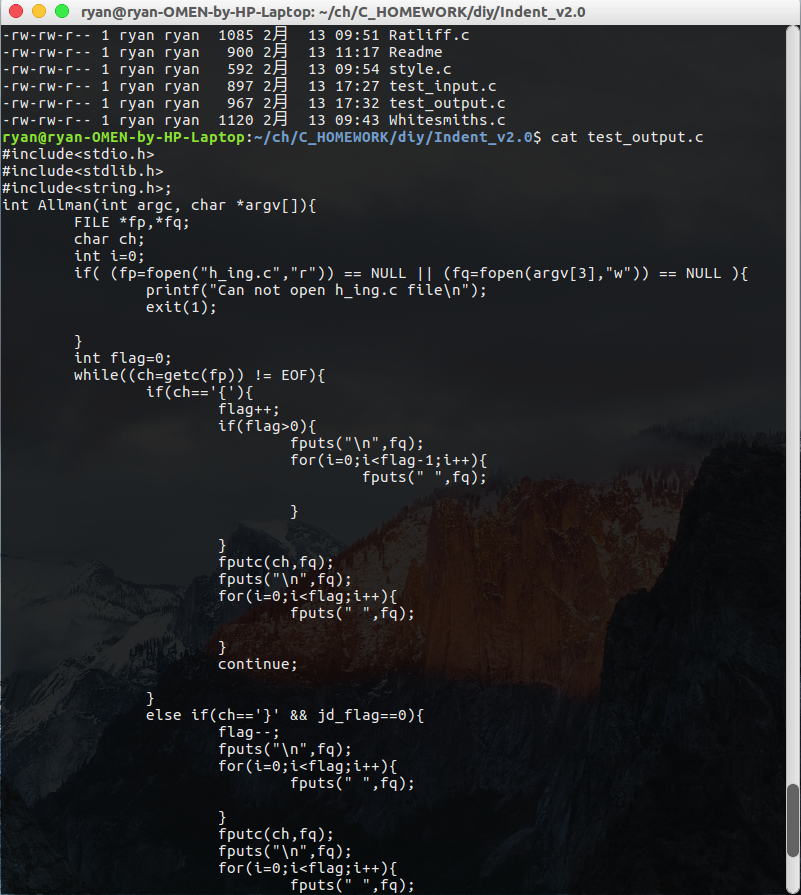
1. 测试分析
2. 测试与调试

命令行调试过程如下图

以K&R风格作为样例







1. 测试结果分析

头文件最后必须手动加上分号；

cat命令显示的缩进默认为8个空格长度，建议使用其他文本编辑器如vim或gedit打开文件查看，并将缩进设置为4空格长度；

1. 结论

此工具能够实现代码的格式化，但使用有一定条件

1. 设计体会
2. 设计过程中的问题及其解决方案

（1）命令行带参数，查阅资料可知，main函数可带参数argc,argv，并且用于获取命令行参数

（2）文件的删除，使用 remove 函数

（3）向文件输入退格，使用改变文件指针的fseek函数实现（fseek(fq,-1,SEEK\_CUR)）

1. 设计过程中的错误及其原因分析

文件名正确但是，无法打开，原因：打开文件时的模式选用错误；

向文件中输入退格时，采用fputs(“\b \b”);的方式，打开文件后显示BS BS (backspace)，原因：‘\b’退格的作用是在终端的屏幕上实现的，不能向文件中输入

1. 体会和收获

平时学习到的知识在实际开发中是远远不够的，通过查阅文档、上网搜索扩展必不可少；

要养成良好的代码风格，还要注意多写注释；