

indent

格式化C语言的源文件

补充说明

indent命令可辨识C的原始代码文件，并加以格式化，以方便程序员阅读、修改等操作。

语法

```
indent (选项)(源文件)
或
indent [选项](源文件) (-o 目标文件)
```

选项

```
-bad [选项]在声明区加上空白行；
-bap [选项]添加空白行；
-bbb [选项]在注释后面添加空白行；
-bc [选项]在声明段中，如果出现逗号就换行；
-bl [选项]if [选项]或是else [选项]for等)与后面执行区段的“{”不同行，且“}”自成一格-bli<缩排格数>设置{}缩排的格数；
-br [选项]if [选项]或是else [选项]for等)与后面执行区段的“{”同行，且“}”自成一格；
-bs [选项]在sizeof之后空一格；
-c<栏数>：将注释置于程序右侧指定的栏位；
-cd<栏数>：将注释置于声明右侧指定的栏位；
-cdb [选项]注释符号自成一格；
-ce [选项]将else置于“}”[选项]if执行区段的结尾)之后；
-ci [选项]<缩排格数>：叙述过长而换行时，指定换行后缩排的格数；
-cli [选项]<缩排格数>：使用case时，switch缩排的格数；
-cp<栏数>：将注释置于else与elseif叙述右侧指定的栏位；
-cs [选项]在case之后空一格；
-d<缩排格数>：针对不是放在程序码右侧的注释，设置其缩排格数；
-di<栏数>：将声明区段的变量置于指定的栏位；
-fc1 [选项]针对放在每行最前端的注释，设置其格式；
-fca [选项]设置所有注释的格式；
-gnu [选项]使用指定的GNU格式，该参数为默认值；
-i<格数>：设置缩排的格数；
-ip<格数>：设置参数的缩排格数；
-kr [选项]指定使用Kernighan&Ritchie的格式；
-lp [选项]叙述过长而换行，且叙述中包含了括号时，将括号中的每行起始栏位内容垂直对其排列；
-nbad [选项]在声明区段后不要加上空白行；
-nbap [选项]在程序后面不添加空白行；
-nbbb [选项]在注释段后面不添加空白行；
-nbc [选项]在声明段中，即使出现逗号，也不换行；
-ncdb [选项]注释符号不自成一格；
-nce [选项]不将else置于“}”后面；
-ncs [选项]不在case后面空一格；
-nfc1 [选项]不要格式化放在每行最前端的注释；
-nfca [选项]不用格式化任何的注释；
-nip [选项]参数不要缩排；
```

- nlp[]叙述过长而换行，且叙述中包含了括号时，不用将括号中的每行起始栏位垂直对其排列；
- npcs[]在调用函数名之后，不要添加空格；
- npro[]不要读取indent的配置文件“.indent.pro”[]
- npsl[]程序类型与程序名称放在同一行；
- nsc[]注释左侧不要添加星号；
- nsob[]不用处理多余的空白行；
- nss[]若for或while区段仅有一行时，在分号前不加空格；
- nv[]不显示详细的信息；
- orig[]使用berkeley格式；
- pcs[]在调用函数名与“{”之间添加空格；
- psl[]程序类型置于程序名称的前一行；
- sc[]在每行注释左侧添加星号；
- sob[]删除多余的空白行；
- ss[]若for或swile区段仅有一行时，在分号前加上空格；
- st[]将结果显示在标准输出设备上；
- T[]数据类型名称缩排；
- ts<格数>：设置tab的长度；
- v[]显示详细的执行过程；
- version[]显示版本信息。

实例

使用indent命令将C语言源文件“test.c”中所有的sizeof后面添加一个空格，输入如下命令：

```
indent -bs /home/rootlocal/桌面/test.c
```

执行上面的命令后，用户可以打开指定的源文件查看在sizeof后面是否都添加了一个空格。由于该命令的参数非常多，所以用户可以根据实际需要选择适合的参数进行使用即可。

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:indent>

Last update: **2021/10/15 14:58**

