gcc中的-w -W和-Wall选项

原创 冬冬咚咚咚 于 2015-10-30 23:20:36 发布 51102 收藏 37
分类专栏： gcc编译 文章标签： gcc

今天在看一个makefile时看到了gcc -W -Wall....这句，不明其理，专门查看了gcc的使用手册。

-w的意思是关闭编译时的警告，也就是编译后不显示任何warning，因为有时在编译之后编译器会显示一些例如数据转换之类的警告，这些警告是我们平时可以忽略的。

-Wall选项意思是编译后显示所有警告。

-W选项类似-Wall，会显示警告，但是只显示编译器认为会出现错误的警告。

在编译一些项目的时候可以-W和-Wall选项一起使用。

举个例子：

```c
#include <stdio.h>

void main()
{
        int a=1.0*4;
        return 0;
}
```

直接编译
gcc -o test_w_wall testwwall.c

```
testwwall.c: In function ‘main’:
testwwall.c:6:2: warning: ‘return’ with a value, in function returning void [enabled by default]
  return 0;                              http://blog.csdn.net/
  ^
```

只显示这一个警告，下面使用-w选项。

gcc -w -o test_w_wall testwwall.c

不会显示任何警告，直接编译成功。

gcc -Wall -o test_w_wall testwwall.c

```
testwwall.c:3:6: warning: return type of ‘main’ is not ‘int’ [-Wmain]
 void main()
      ^
testwwall.c: In function ‘main’:
testwwall.c:6:2: warning: ‘return’ with a value, in function returning void [enabled by default]
  return 0;                            http://blog.csdn.net/
  ^
testwwall.c:5:13: warning: unused variable ‘a’ [-Wunused-variable]
         int a=1.0*4;
             ^
```

显示了所有的警告，比之前不使用任何选项多出了变量a未使用这个警告，也多出了main函数的返回值不是int型。

gcc -W -o test_w_wall testwwall.c

```
testwwall.c: In function ‘main’:
testwwall.c:6:2: warning: ‘return’ with a value, in function returning void [enabled by default]
  return 0;
  ^
```

只显示了没有返回值的main函数不应该有return一个值这个警告。

gcc -W -Wall test_w_wall testwwall.c

```
testwwall.c:3:6: warning: return type of ‘main’ is not ‘int’ [-Wmain]
 void main()
      ^
testwwall.c: In function ‘main’:
testwwall.c:6:2: warning: ‘return’ with a value, in function returning void [enabled by default]
  return 0;
  ^
testwwall.c:5:13: warning: unused variable ‘a’ [-Wunused-variable]
         int a=1.0*4;
             ^
```

比单独使用-W多出了变量为使用这个警告，比-Wall选项少了一个看起来重复的main函数返回值不是int这个警告。

之前看了一篇国外程序员写的博客，说编译时不使用-W -Wall选项的是stupid的，所以编译时还是尽量带上吧。