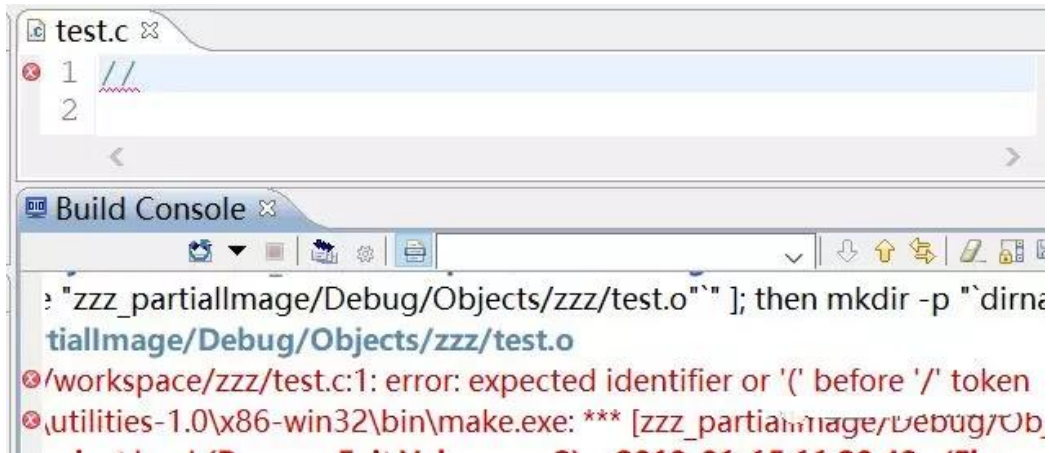


VxWorks 编译环境下对代码注释的要求

很多编码人员习惯于在单行注释时使用双斜线(//)，因为输入方便。

```
01:  
02: // this is a single-line comment  
03:  
04: /* this is another single-line comment */  
05:  
06: /*  
07:  * this is a multiline commen  
08:  */
```

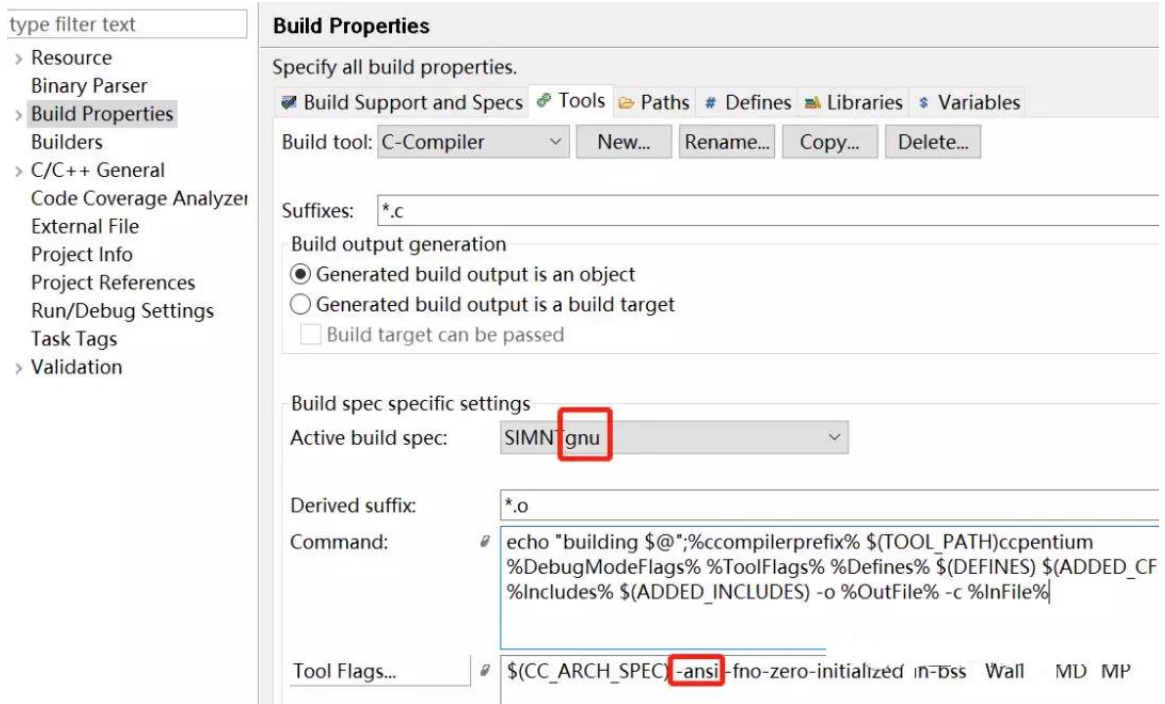
不过 VxWorks 的开发人员可能就会遇到一个问题了



上图是在 Workbench 的 DKM 工程里添加一个 c 文件，该文件中只包含一个双斜线，编译工程就报错了：expected identifier or '(' before '/' token

在 Tornado 里，也会遇到类似的问题。这是为什么呢？

因为这个工程中，IDE 使用的编译器是 gnu，编译规则默认包含了 -ansi



从 gnu 的手册中，可以看到-ansi 等同于-std=c89，而 c89 那个年代，是不支持双斜线的

iso9899:1990

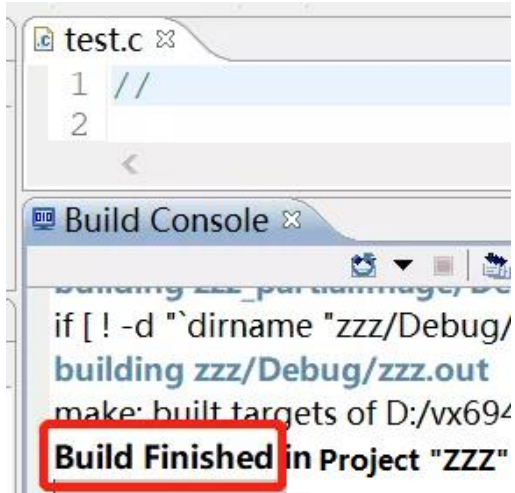
c89 The ISO C standard from 1990. 'c89' is the customary shorthand for this version of the standard.

The '-ansi' option is equivalent to '-std=c89'.

问题找到了，怎么解决呢？最简单粗暴的办法就是 -- 删了它！



再编译一下试试



搞定了!

荆溪惠子鬢鬕髻，义理钻研到粹精。只是搞定还不行，我们还要知道为什么搞定。继续看手册

gnu89 The 1990 C standard plus GNU extensions. This is the default.

原来删除-ansi后，IDE就会将规则重置为-std=gnu89，而正是gnu89对c89做了一些扩展，其中就包含了这个双斜线注释。

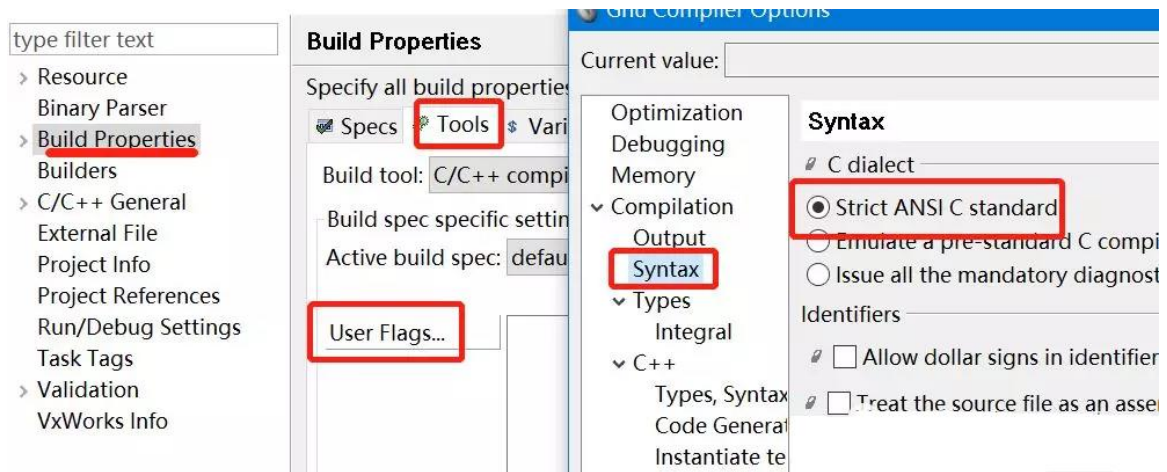
那我们直接将规则设置为gnu89，甚至c99、gnu99之类的呢？

Derived suffix:	*.o
Command:	echo "building \$@";%ccompilerprefix% \$(TOOL_PATH)ccpentium %DebugModeFlags% %ToolFlags% %Defines% \$(DEFINES) \$(ADDED_C %Includes% \$(ADDED_INCLUDES) -o %OutFile%
Tool Flags...	\$(CC_ARCH_SPEC) -std=c99 -fno-zero-initialized -Wall -MD -MP

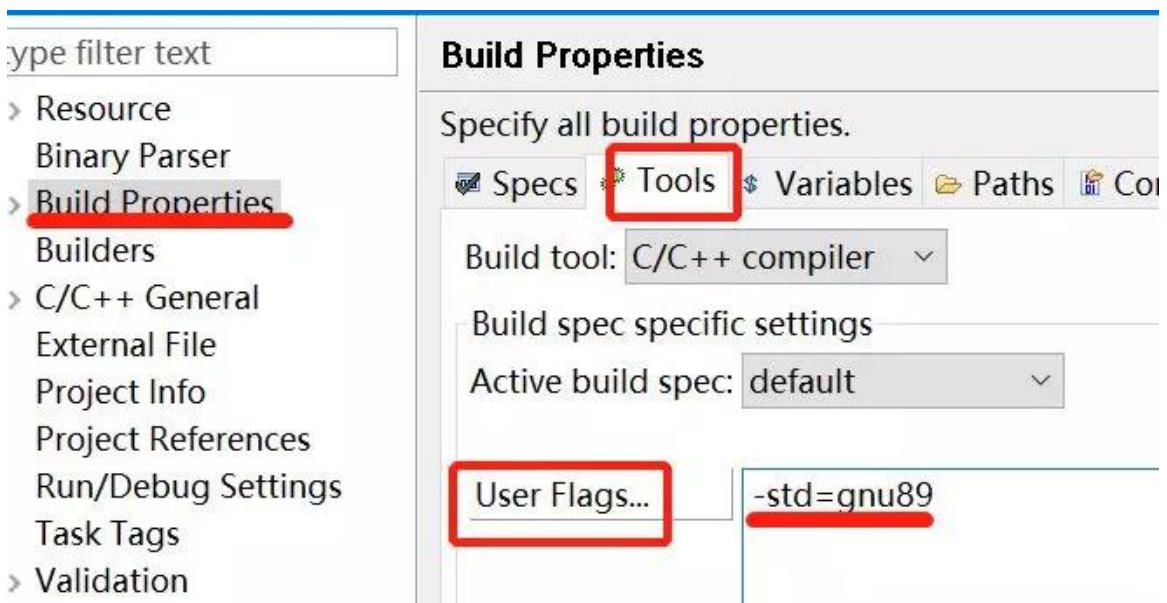
试一下，都没有问题。

不过，毕竟和c89相比，c99的变化太多，编译规则改动太大，可能会涉及到代码移植或兼容的问题，这就需要项目经理或技术负责人来统筹定制规则了。事实上很多公司都会制定自己的编码规范，其中就会包括注释的书写规则。因此我们这里就先使用gnu89吧。

刚刚验证的是DKM工程，我们再打开VIP看看。VIP中的ansi选项是用单选框形式提供的，那怎么修改呢？

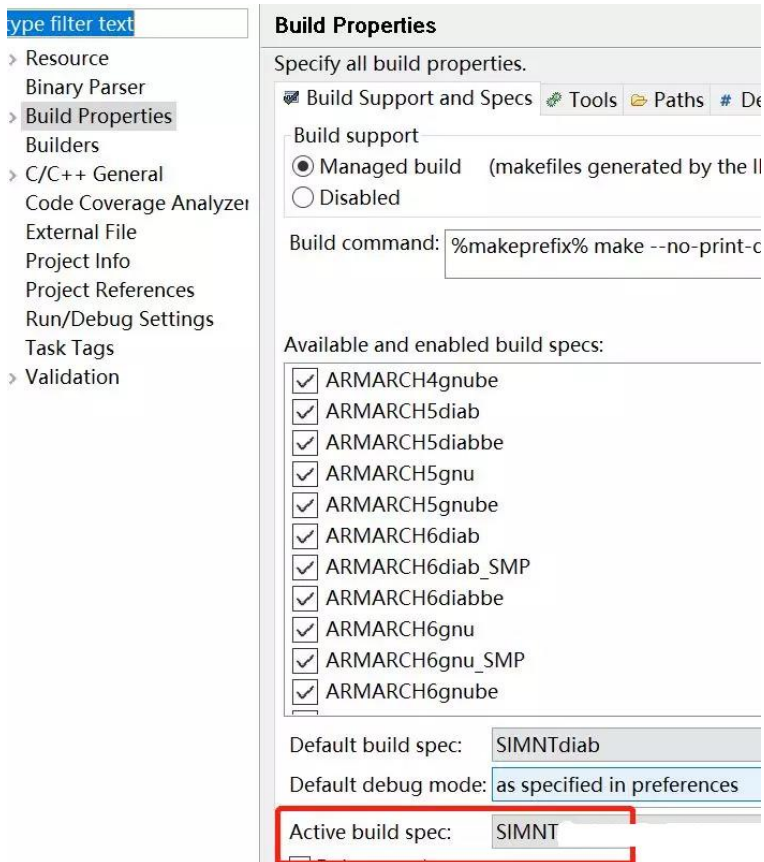


直接在 User Flags 里填写就可以了。

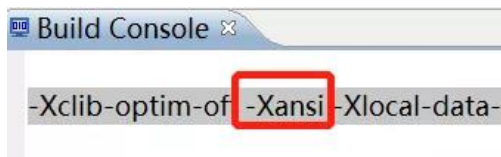


这就是 gnu 编译器编译双斜线注释的问题。

提到 gnu 编译器，你是不是又想到一种办法？对，就是用 diab 编译器来编译工程。



试一下，果然没问题。难道 diab 默认没有使用 ansi 选项？看一下 Build Console，它用的选项是 `-Xansi`



这是什么鬼？看看 diab 手册，`-Xansi` 的全称是 `-Xdialect-ansi`。原来是厂家专门做的改进，就像是 gnu89 对 c89 的改进。

`-Xdialect-ansi`

`-x7=1`

Follow the ANSI C standard with some additions.

This is the default.

Synonyms: `-Xansi`, `-Xa`.

看看它改进了哪些，果然有这一条

**C++ comments `///
好，到此为止，都搞定了！`**