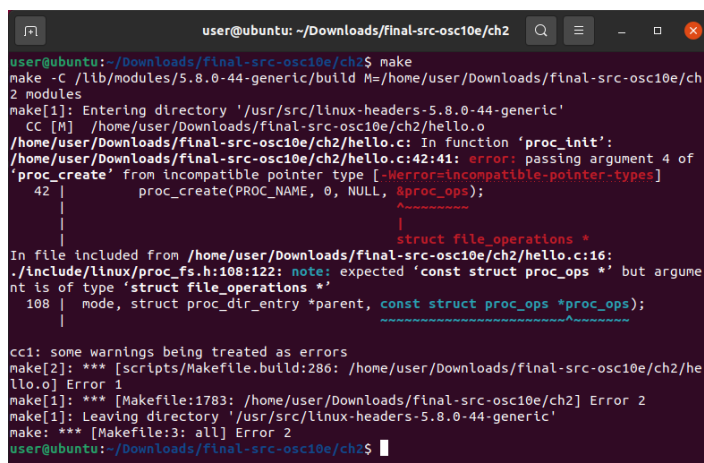


问题：



```
user@ubuntu: ~/Downloads/final-src-osc10e/ch2
user@ubuntu:~/Downloads/final-src-osc10e/ch2$ make
make -C /lib/modules/5.8.0-44-generic/build M=/home/user/Downloads/final-src-osc10e/ch2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.8.0-44-generic'
CC [M] /home/user/Downloads/final-src-osc10e/ch2/hello.o
/home/user/Downloads/final-src-osc10e/ch2/hello.c: In function 'proc_init':
/home/user/Downloads/final-src-osc10e/ch2/hello.c:42:41: error: passing argument 4 of
'proc_create' from incompatible pointer type [-Werror=incompatible-pointer-types]
42 |         proc_create(PROC_NAME, 0, NULL, &proc_ops);
    |                                         ^
    |                                         |
    |                                         struct file_operations *
In file included from /home/user/Downloads/final-src-osc10e/ch2/hello.c:16:
./include/linux/proc_fs.h:108:122: note: expected 'const struct proc_ops *' but argument
nt is of type 'struct file_operations *'
108 |     mode, struct proc_dir_entry *parent, const struct proc_ops *proc_ops);
    |                                         ^
cc1: some warnings being treated as errors
make[2]: *** [scripts/Makefile.build:286: /home/user/Downloads/final-src-osc10e/ch2/hello.o] Error 1
make[1]: *** [Makefile:1783: /home/user/Downloads/final-src-osc10e/ch2] Error 2
make[1]: Leaving directory '/usr/src/linux-headers-5.8.0-44-generic'
make: *** [Makefile:3: all] Error 2
user@ubuntu:~/Downloads/final-src-osc10e/ch2$
```

回复邮件：

同学你好：

首先，你所发送的错误信息截图的意思是：你在向 `proc_create` 函数传递参数时类型错误，第四个参数需要“`const struct proc_ops *`”类型，而你向它传递了“`struct file_operations *`”类型，导致报错。

所以，我们可以知道导致这一编译错误的原因是，你向函数传递了错误的参数，因此下一步，出于好奇，我们需要查看这个函数 `proc_create` 的定义，在 `include/linux/proc_fs.h` 的第 108 行，确实是这个类型，并不是我们的修改导致的错误。

所以接下来，我们就考虑，是否在内核版本变化过程中，函数定义出现了问题，所以我们需要查看不同版本的内核代码。我们发现在 4.14 版本中，`proc_create` 的第四个参数还是“`struct file_operations *`”类型[1]，但是在你所使用的 5.8.0 版本中修改为了“`const struct proc_ops *`”类型[2]。

综上，我们可以得出结论，本书所给的示例是基于旧版本的内核，与你所用的新版本内核在部分函数上存在变化。

最后，同学你提出的问题确实在内核开发过程中很典型，材料的变化很多时候会落后于内核的实际内容，但是这个问题本身并不是很复杂，报错信息也很充分。出于详细说明的目的，我把分析思路写的很详细，但其实这个问题的定位相当直观（甚至比你数据结构算法题 debug 还直观），所以希望你能够从上述思路中得到一些启发，并活用于下次的 debug 过程中。

当然，如果有解决不了的问题，欢迎以邮件或微信的方式向助教提问。

[1] https://elixir.bootlin.com/linux/v4.14/source/include/linux/proc_fs.h#L32

[2] https://elixir.bootlin.com/linux/v5.8/source/include/linux/proc_fs.h#L107