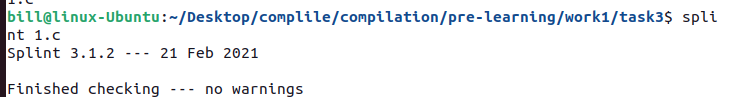
|  |
| --- |
| char firstChar1(char\* s)  {  return \*s;  } |

对于该程序，s必须指向一个确定的地址，当s为空指针或野指针时，无法返回s指向的值，因此出错

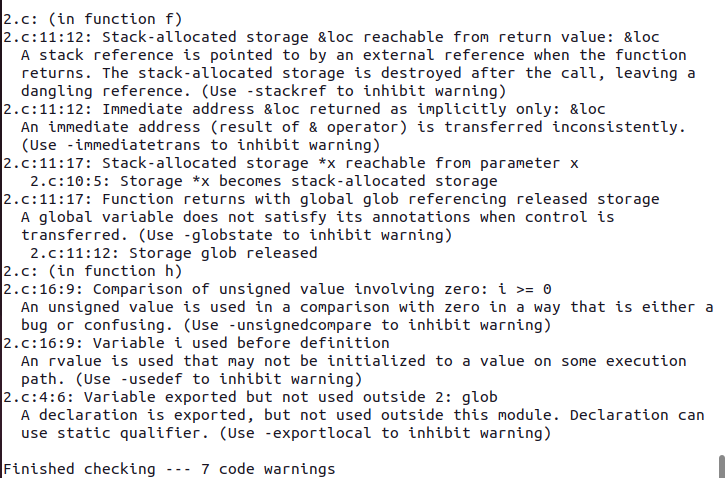


Splint检查结果提示没有问题

|  |
| --- |
| int\* glob;  int\* f(int\*\* x)  {  int sa[2] = { 0, 1 };  int loc = 3;  glob = &loc;  \*x = &sa[0];  return &loc;  }  void h(void)  {  unsigned int i;  if (i >= 0)  printf(">=0\n");  else printf("<0");  } |

对于函数f，最主要的问题是sa是临时变量，函数返回的指针指向栈中，而函数执行完即销毁栈，因此不能返回需要的结果。该地址存储的值可能会随程序执行而发生变化；另外，x也必须指向一个确切的地址

对于函数h，临时变量i没有初始化。另外，对于无符号数的大于或等于0比较，是恒成立的。



Splint检查结果提示

在f函数中：

1. 返回的指针是悬挂，并不指向一个有效的数据地址，因为它指向的数据随着函数结束而被销毁了
2. 第二个错误不太明白，可能是指向立即数的地址是隐式返回的，且求该地址的运算不是一致性的（应该是该地址具有随机性，不是每次运算都得到一样的结果）
3. \*x被函数栈帧分配存储空间，它所指向的，也是一个栈帧中的地址
4. 全局变量global指针同样被悬挂
5. 无符号数带0比较，要么是bug，要么是迷惑操作
6. I在使用前没有被定义
7. Glob是全局变量，但没有被其他模块调用，可以用静态限定符来限定