## 实验八 Linux开发工具的使用

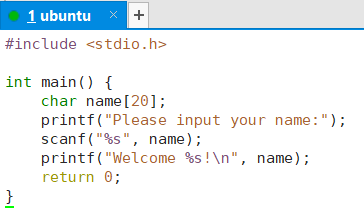
## 实验目的

（1）掌握gcc的编译方法与使用

（2）掌握Linux调试器GDB的使用

## 实验内容

1. **编译器gcc的使用**
   1. 编辑一个C语言程序文件 hello.c ，代码如下：

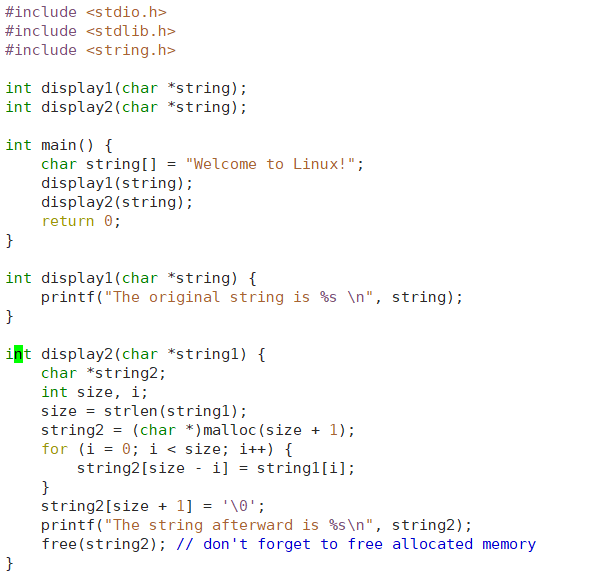


* 1. 编译文件： gcc -o hello hello.c。

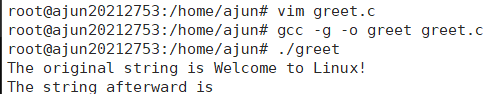


* 1. 若有错误，修改hello.c的内容，然后再次编译，直至没有错误为止。

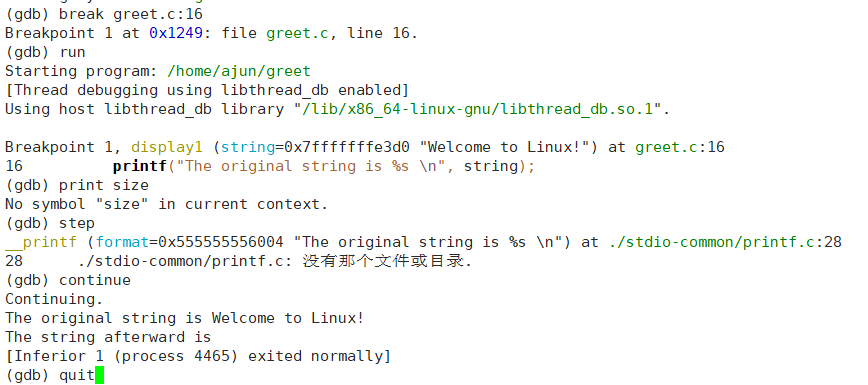
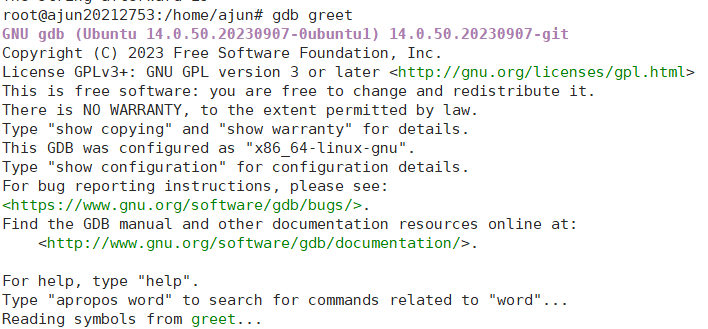
1. **使用GDB 调试程序BUG（教材12.7节）**
   1. 使用文本编辑器输入以下代码greet.c。程序试图倒序输出main 函数中定义的字符串，但结果没有显示。



* 1. 使用gcc –g 的选项编译这段代码，运行生成的可执行文件，观察运行结果。

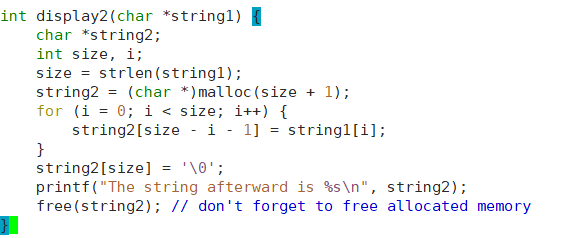


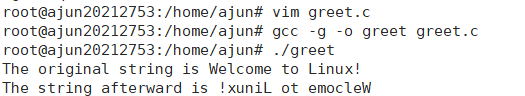
* 1. 使用gdb 调试程序，通过设置断点、单步跟踪，一步步找出错误所在。（调试过程需截图）



* 1. 纠正错误，更改源程序并得到正确的结果。

在 display2 函数中，修复内存越界错误，更改 for 循环，并将 string2[size + 1] = '\0'; 修改为 string2[size] = '\0';，以确保正确的字符串结束符位置。





## 实验总结

在本次实验中，我学习了使用GCC编译器编译C语言程序以及通过GDB调试器来定位和解决程序中的错误。通过设置断点、逐步执行程序并查看变量值，我们能够更有效地理解程序的执行流程并发现潜在的问题。这些技能对于开发和调试软件都是至关重要的，能够帮助我们提高代码质量和效率。