|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 1502760001-07  题目类型 实验1 | |  |  |  | | --- | --- | --- | | **得分** | **教师签名** | **批改日期** | |  | 冯禹洪 |  | |

**深 圳 大 学 实 验 报 告**

**课 程 名 称： 计算机系统(2)**

**实验项目名称： 实验环境配置与使用**

**学 院： 计算机与软件学院**

**专 业： 软件工程（腾班）**

**指 导 教 师： 冯禹洪**

**报告人： 黄亮铭 学号： 2022155028 班级： 腾班**

**实 验 时 间： 2024年04月12日**

**实验报告提交时间： 2024年04月26日**

**教务处制**

**一、 实验目标：**

熟悉Linux上C程序的编译和调试工具，包括以下内容：

1. 了解Linux操作系统及其常用命令

2. 掌握编译工具gcc的基本用法

3. 掌握使用gdb进行程序调试

**二、实验环境与工件**

**1.**个人电脑

2. Linux 操作系统

3. gcc

4. gdb

**三、实验内容与步骤**

1. 根据实验一：实验环境配置与使用.ppt熟悉Linux基本操作（P.1 – P.28）,然后根据以下过程创建用户：用户名为学生名称加学号，如**吴坤汉**，学号**2015170297**，则该用户名为wukunhan\_2015170297。按照1.1~1.3完成并截图，截图需要有运行的命令及其结果。另外：**后面的题目必须在该新建用户下完成。（30分）**

实验步骤：

* 1. 首先切换为超级用户

*$su*

如果出现Authentication failure的情况，则可以用`sudo passwd root`来给root重设密码：

**1.1任务截图**

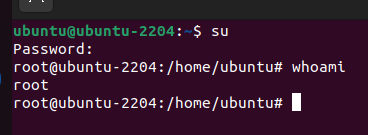


图1：切换为超级用户

* 1. 参考以下命令创建新用户, 设置新建用户的密码，注意：只有设置了密码才能激活用户，否则无法以该用户身份登录

*#adduser wukunhan\_2015170297*

**1.2任务截图**

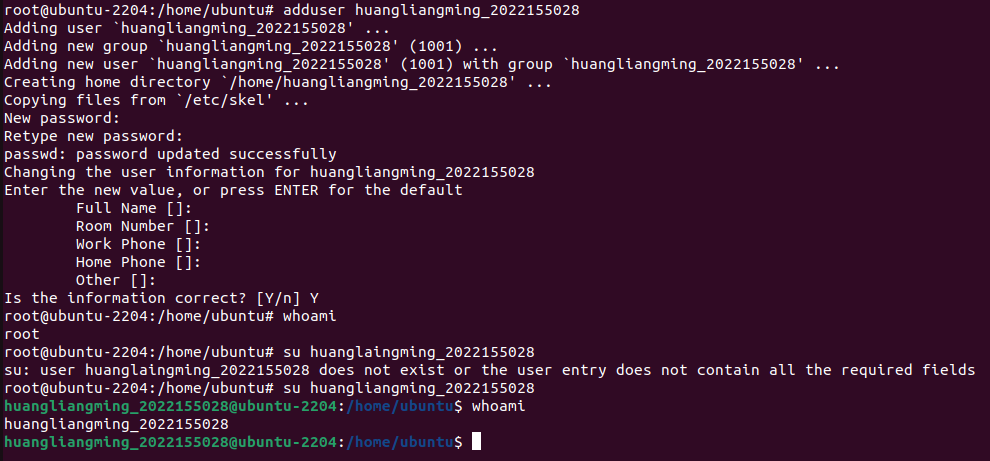
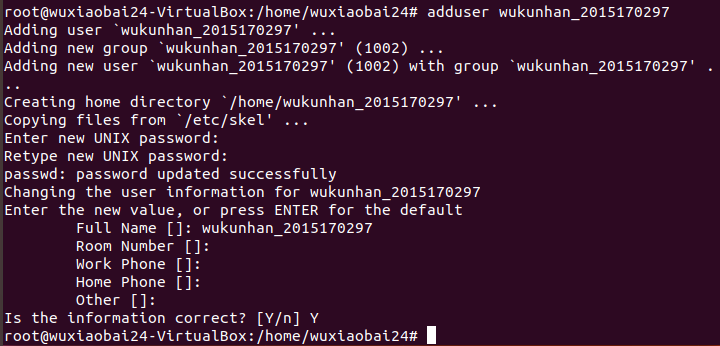


图2：新建用户



* 1. 注销当前用户，并以新建的用户身份登录，登录后运行 $ whoami，并进行截图；

**1.3任务截图**

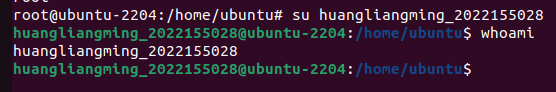


图3：以新建用户身份登录

1. 新建用户主目录下创建子目录：**gdbdebug**, 并进入gdbdebug 子目录。将过程和结果截图。（10分）

**任务2截图**

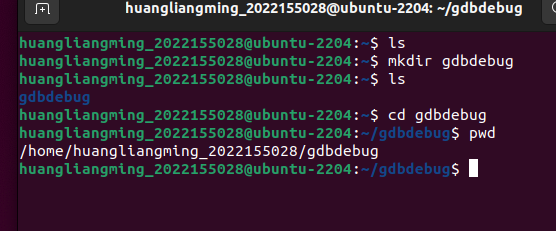


图4：创建并进入子目录

1. 使用vi编辑以下两个文件并编译和运行，截图（30分）
   1. 编辑reverse.h

**任务3.1截图**

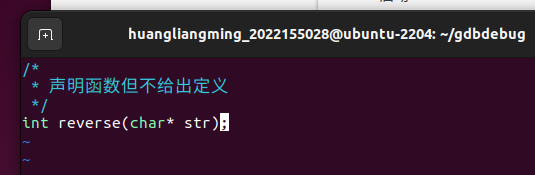


图5：头文件

* 1. 编辑reverse.c

**任务3.2截图**

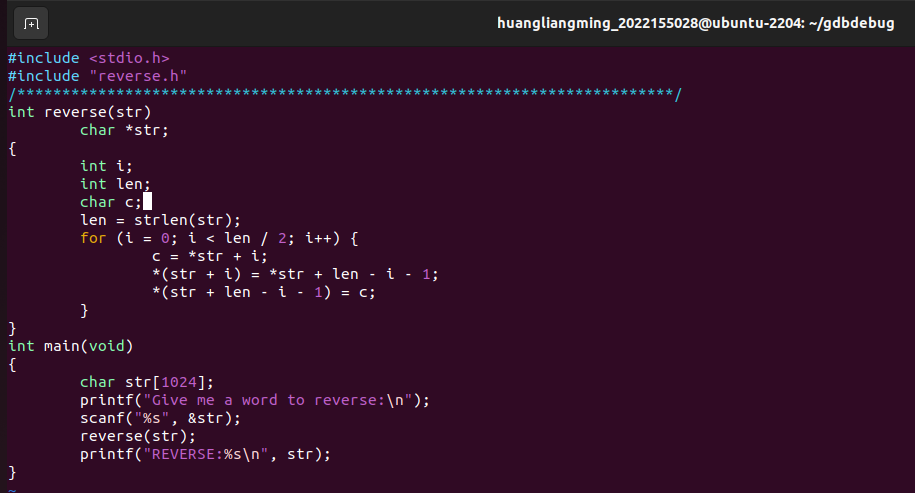


图6：C语言源文件

* 1. 按以下步骤编译，如有警告信息，请修改代码至无警告信息

$gcc –Wall reverse.c –o reverse

**任务3.3截图（修改前）**

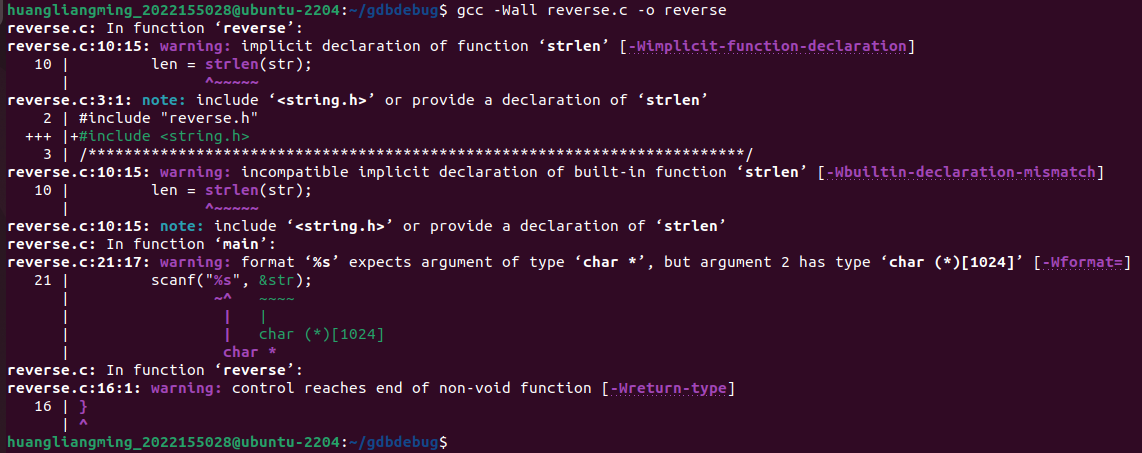


图7：源文件警告信息

**警告信息分析：**阅读警告信息，发现①我们在使用库函数strlen的时候没有引用string.h这个库；②我们在使用scanf的时候，参数类型不匹配。

**更改：**我们首先在reverse.c文件中引入string.h这个库，然后将scanf的参数&str更改为str。重新编译后发现警告信息已经消失，证明修改正确。

**任务3.3截图（修改后）**

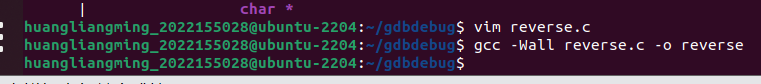


图8：无报错（修改后）

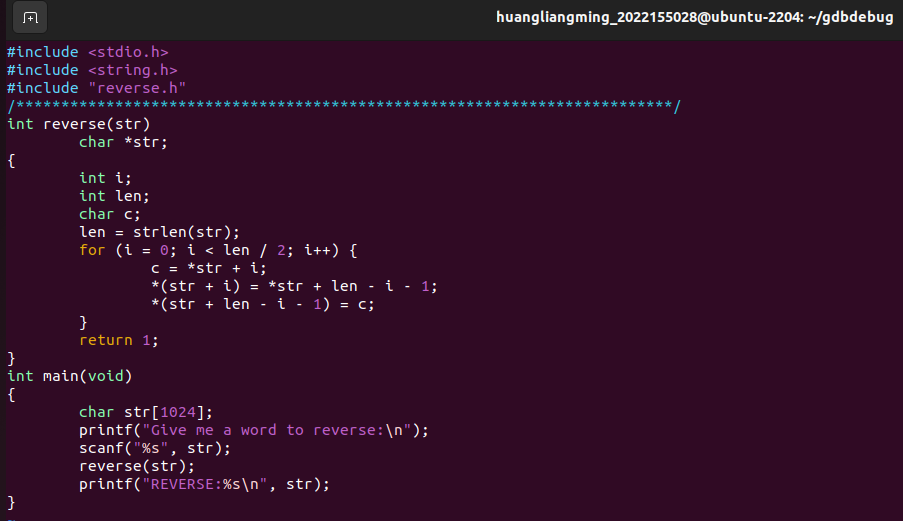


图9：修改后的代码

* 1. 运行程序

$./reverse

**任务3.4截图（输入hello world！）**

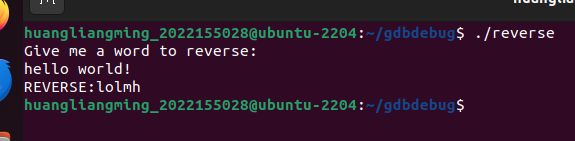


图10：运行程序

1. 按照以下过程调试并修正reverse.c,请参考过程截图。（30分）
   1. 编译时加入调试信息

**$gcc -g reverse.c -o reverse1**

**任务4.1截图**



图11：运行命令

* 1. 启用GDB调试

**任务4.2截图**

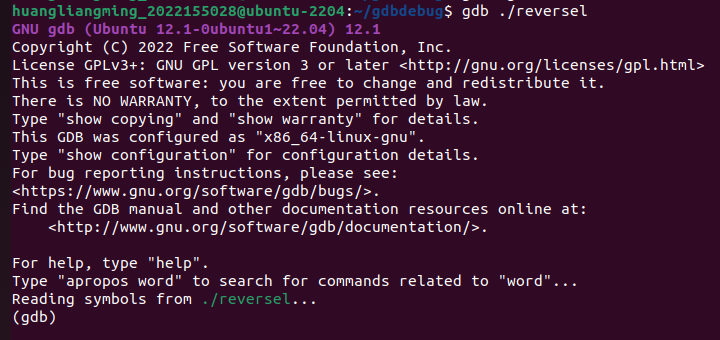


图12：启动GDB调试

* 1. 键入list,查看源代码并根据行号/函数名设置断点（断点位置仅供参考，不需雷同）

**任务4.3截图**



图13：键入list

**设置断点：**我选择在源代码的13、14、15和25行设置断点（即变量str出现的并且可能出现错误的位置），一共4个。

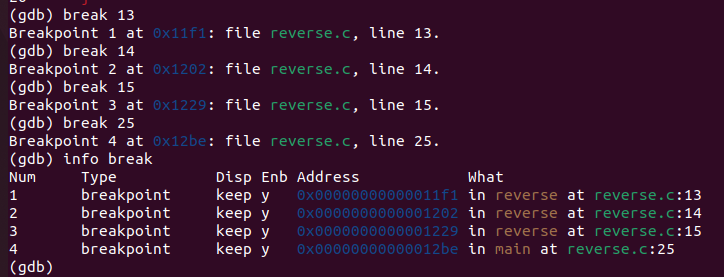


图14：设置断点

* 1. 观察变量值，并作分析，推测错误（过程仅供参考，不需雷同）

**任务4.4截图：**在前五次步进时，str的值均正确，但是在第六次时出现了意外值f，在第七次时出现了意外值e，因此合理推测错误可能存在于第五次或者第六次步进时程序运行的代码中。

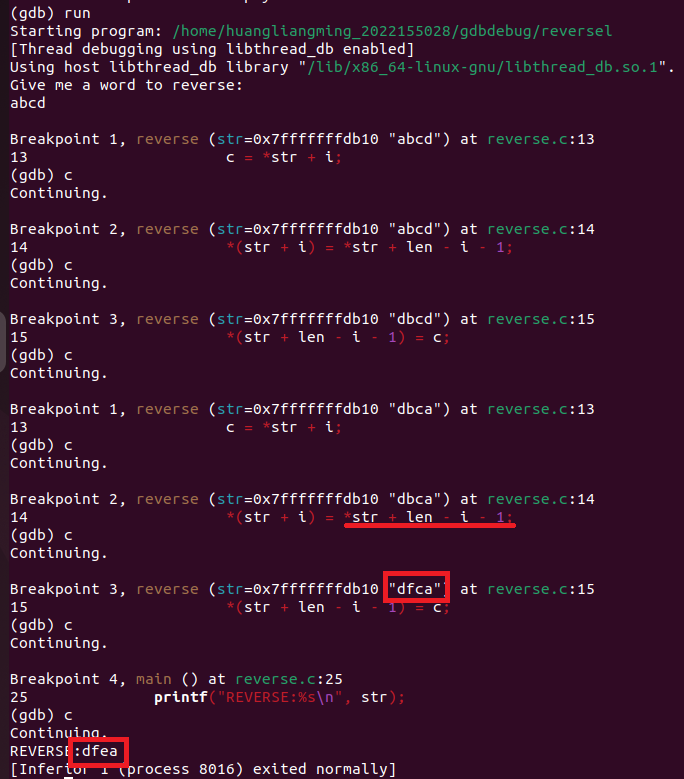


图15：断点调试过程

* 1. 修正程序并运行

**任务4.5截图**



图16：修正后程序代码

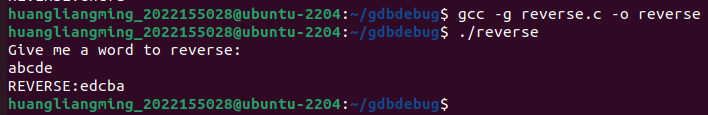


图17：修正后程序运行结果

**四、实验结果**

1. 成功新建用户，并在新用户主目录下创建了子目录gdbdebug；
2. 使用vim完成了在gdbdebug目录下编写reverse.c和reverse.h文件；
3. 根据警告信息，将程序内容一一修改，使程序编译时不再报任何信息，即成功编译；
4. 根据提示，对程序设置断点，使用gdb进行debug操作，使程序达到我们预期的效果。

**五、实验总结与体会**

1. 通过本次实验，我掌握了linux命令行的基本操作；
2. 我学习了在linux下的编程方法；
3. 我熟悉了如何使用gdb对代码进行调试，使代码产生的结果符合我们的预期；

|  |
| --- |
| **指导教师批阅意见：**  **成绩评定：**  指导教师签字：  年 月 日 |
| 备注： |