

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）

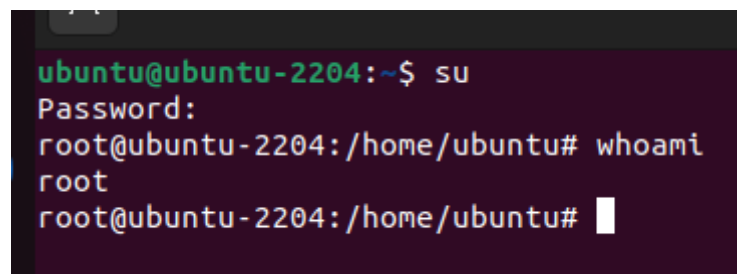
实验步骤：

- 1.1. 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



```
ubuntu@ubuntu-2204:~$ su
Password:
root@ubuntu-2204:/home/ubuntu# whoami
root
root@ubuntu-2204:/home/ubuntu#
```

图 1：切换为超级用户

- 1.2. 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

```
root@ubuntu-2204:/home/ubuntu# adduser huangliangming_2022155028
Adding user 'huangliangming_2022155028' ...
Adding new group 'huangliangming_2022155028' (1001) ...
Adding new user 'huangliangming_2022155028' (1001) with group 'huangliangming_2022155028' ...
Creating home directory '/home/huangliangming_2022155028' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for huangliangming_2022155028
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
root@ubuntu-2204:/home/ubuntu# whoami
root
root@ubuntu-2204:/home/ubuntu# su huangliangming_2022155028
su: user huangliangming_2022155028 does not exist or the user entry does not contain all the required fields
root@ubuntu-2204:/home/ubuntu# su huangliangming_2022155028
huangliangming_2022155028@ubuntu-2204:/home/ubuntu$ whoami
huangliangming_2022155028
huangliangming_2022155028@ubuntu-2204:/home/ubuntu$
```

图 2：新建用户

```
root@wuxiaobai24-VirtualBox:/home/wuxiaobai24# adduser wukunhan_2015170297
Adding user 'wukunhan_2015170297' ...
Adding new group 'wukunhan_2015170297' (1002) ...
Adding new user 'wukunhan_2015170297' (1002) with group 'wukunhan_2015170297' .
..
Creating home directory '/home/wukunhan_2015170297' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for wukunhan_2015170297
Enter the new value, or press ENTER for the default
  Full Name []: wukunhan_2015170297
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
root@wuxiaobai24-VirtualBox:/home/wuxiaobai24#
```

- 1.3. 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；

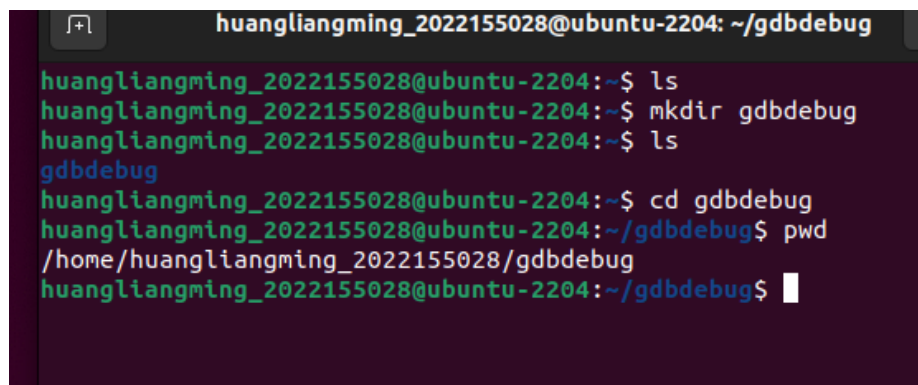
1.3 任务截图

```
root@ubuntu-2204:/home/ubuntu# su huangliangming_2022155028
huangliangming_2022155028@ubuntu-2204:/home/ubuntu$ whoami
huangliangming_2022155028
huangliangming_2022155028@ubuntu-2204:/home/ubuntu$
```

图 3：以新建用户身份登录

2. 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



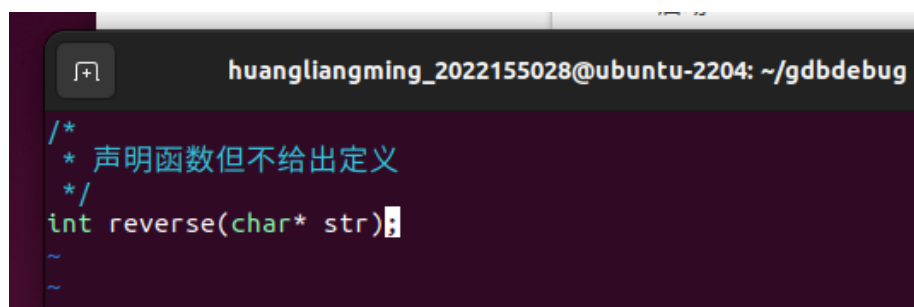
```
huangliangming_2022155028@ubuntu-2204: ~/gdbdebug
huangliangming_2022155028@ubuntu-2204:~$ ls
huangliangming_2022155028@ubuntu-2204:~$ mkdir gdbdebug
huangliangming_2022155028@ubuntu-2204:~$ ls
gdbdebug
huangliangming_2022155028@ubuntu-2204:~$ cd gdbdebug
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ pwd
/home/huangliangming_2022155028/gdbdebug
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

图 4：创建并进入子目录

3. 使用 vi 编辑以下两个文件并编译和运行，截图（30 分）

3.1. 编辑 reverse.h

任务 3.1 截图

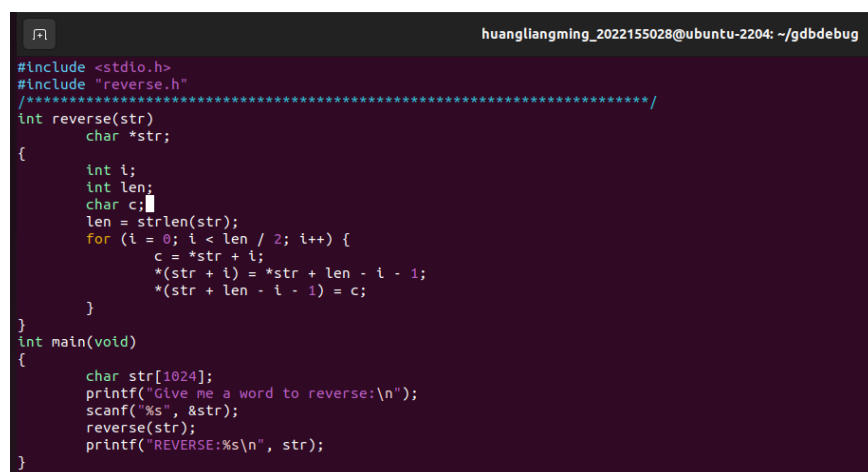


```
/*
 * 声明函数但不给出定义
 */
int reverse(char* str);
```

图 5：头文件

3.2. 编辑 reverse.c

任务 3.2 截图



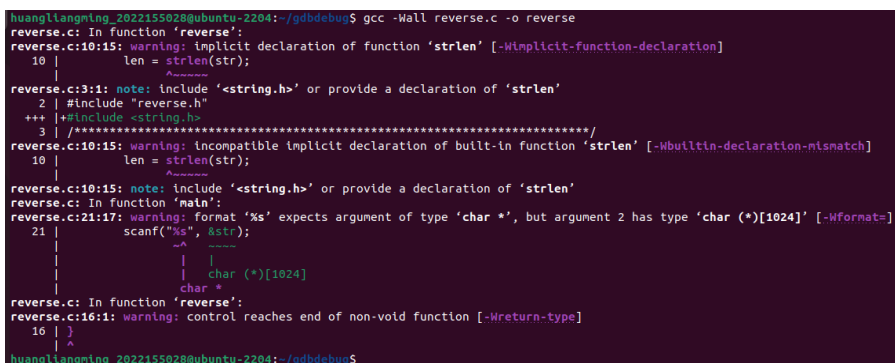
```
#include <stdio.h>
#include "reverse.h"
/*****
int reverse(str)
    char *str;
{
    int i;
    int len;
    char c;
    len = strlen(str);
    for (i = 0; i < len / 2; i++) {
        c = *str + i;
        *(str + i) = *str + len - i - 1;
        *(str + len - i - 1) = c;
    }
}
int main(void)
{
    char str[1024];
    printf("Give me a word to reverse:\n");
    scanf("%s", &str);
    reverse(str);
    printf("REVERSE:%s\n", str);
}
```

图 6：C 语言源文件

3.3. 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）



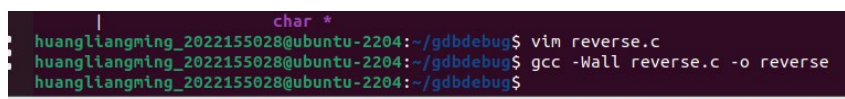
```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ gcc -Wall reverse.c -o reverse
reverse.c: In function 'reverse':
reverse.c:10:15: warning: implicit declaration of function 'strlen' [-Wimplicit-function-declaration]
 10 |     len = strlen(str);
    |               ^~~~~~
reverse.c:3:1: note: include '<string.h>' or provide a declaration of 'strlen'
  3 | /*****
    | 2 | #include "reverse.h"
    | +++ #include <string.h>
    | 3 | /*****
reverse.c:10:15: warning: incompatible implicit declaration of built-in function 'strlen' [-Wbuiltin-declaration-mismatch]
 10 |     len = strlen(str);
    |               ^~~~~~
reverse.c:10:15: note: include '<string.h>' or provide a declaration of 'strlen'
reverse.c: In function 'main':
reverse.c:21:17: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[1024]' [-Wformat=]
 21 |     scanf("%s", &str);
    |                ^~
    |                |
    |                char (*)[1024]
    |                char *
reverse.c: In function 'reverse':
reverse.c:16:1: warning: control reaches end of non-void function [-Wreturn-type]
 16 | }
    | ^
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

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。

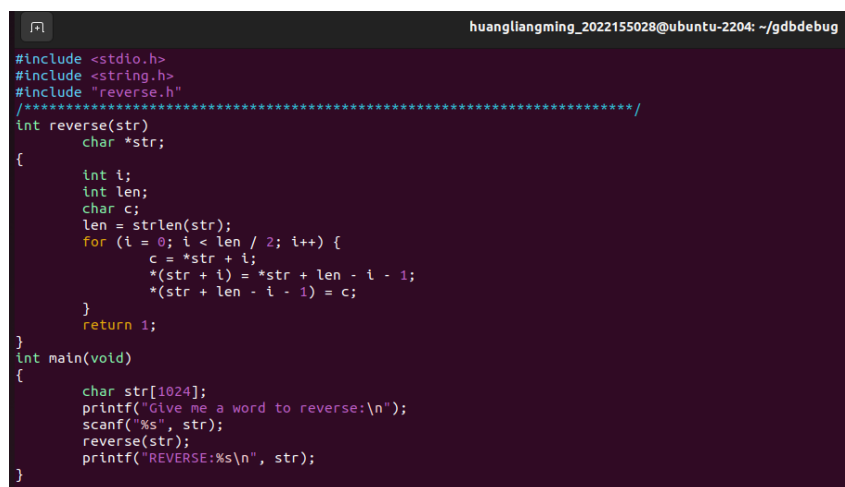
更改：我们首先在 `reverse.c` 文件中引入 `string.h` 这个库，然后将 `scanf` 的参数 `&str` 更改为 `str`。重新编译后发现警告信息已经消失，证明修改正确。

任务 3.3 截图（修改后）



```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ vim reverse.c
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ gcc -Wall reverse.c -o reverse
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

图 8：无报错（修改后）



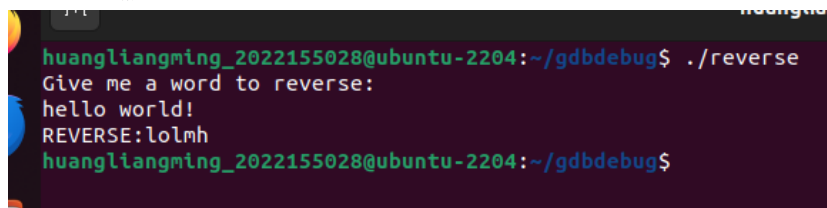
```
huangliangming_2022155028@ubuntu-2204: ~/gdbdebug
#include <stdio.h>
#include <string.h>
#include "reverse.h"
/*****/
int reverse(str)
    char *str;
{
    int i;
    int len;
    char c;
    len = strlen(str);
    for (i = 0; i < len / 2; i++) {
        c = *str + i;
        *(str + i) = *str + len - i - 1;
        *(str + len - i - 1) = c;
    }
    return 1;
}
int main(void)
{
    char str[1024];
    printf("Give me a word to reverse:\n");
    scanf("%s", str);
    reverse(str);
    printf("REVERSE:%s\n", str);
}
```

图 9：修改后的代码

3.4. 运行程序

```
$. ./reverse
```

任务 3.4 截图（输入 hello world!）



```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ ./reverse
Give me a word to reverse:
hello world!
REVERSE:lolmh
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

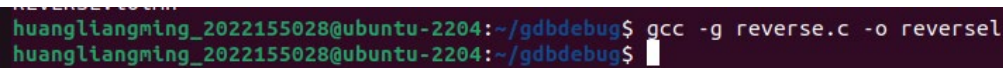
图 10：运行程序

4. 按照以下过程调试并修正 reverse.c, 请参考过程截图。(30 分)

4.1. 编译时加入调试信息

```
$gcc -g reverse.c -o reversel
```

任务 4.1 截图



```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ gcc -g reverse.c -o reversel
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

图 11: 运行命令

4.2. 启用 GDB 调试

任务 4.2 截图



```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ gdb ./reversel
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./reversel...
(gdb)
```

图 12: 启动 GDB 调试

- 4.3. 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



```
(gdb) list
6           char *str;
7       {
8           int i;
9           int len;
10          char c;
11          len = strlen(str);
12          for (i = 0; i < len / 2; i++) {
13              c = *str + i;
14              *(str + i) = *str + len - i - 1;
15              *(str + len - i - 1) = c;
(gdb) list
16          }
17          return 1;
18      }
19      int main(void)
20      {
21          char str[1024];
22          printf("Give me a word to reverse:\n");
23          scanf("%s", str);
24          reverse(str);
25          printf("REVERSE:%s\n", str);
(gdb) list
26      }
(gdb)
```

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。

```

(gdb) break 13
Breakpoint 1 at 0x11f1: file reverse.c, line 13.
(gdb) break 14
Breakpoint 2 at 0x1202: file reverse.c, line 14.
(gdb) break 15
Breakpoint 3 at 0x1229: file reverse.c, line 15.
(gdb) break 25
Breakpoint 4 at 0x12be: file reverse.c, line 25.
(gdb) info break
Num      Type             Disp Enb Address            What
1        breakpoint      keep y   0x00000000000011f1 in reverse at reverse.c:13
2        breakpoint      keep y   0x0000000000001202 in reverse at reverse.c:14
3        breakpoint      keep y   0x0000000000001229 in reverse at reverse.c:15
4        breakpoint      keep y   0x00000000000012be in main at reverse.c:25
(gdb)

```

图 14：设置断点

4.4. 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。

```

(gdb) run
Starting program: /home/huangliangming_2022155028/gdbdebug/reversel
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Give me a word to reverse:
abcd

Breakpoint 1, reverse (str=0x7fffffffdb10 "abcd") at reverse.c:13
13      c = *str + i;
(gdb) c
Continuing.

Breakpoint 2, reverse (str=0x7fffffffdb10 "abcd") at reverse.c:14
14      *(str + i) = *str + len - i - 1;
(gdb) c
Continuing.

Breakpoint 3, reverse (str=0x7fffffffdb10 "dbcd") at reverse.c:15
15      *(str + len - i - 1) = c;
(gdb) c
Continuing.

Breakpoint 1, reverse (str=0x7fffffffdb10 "dbca") at reverse.c:13
13      c = *str + i;
(gdb) c
Continuing.

Breakpoint 2, reverse (str=0x7fffffffdb10 "dbca") at reverse.c:14
14      *(str + i) = *str + len - i - 1;
(gdb) c
Continuing.

Breakpoint 3, reverse (str=0x7fffffffdb10 "dfca") at reverse.c:15
15      *(str + len - i - 1) = c;
(gdb) c
Continuing.

Breakpoint 4, main () at reverse.c:25
25      printf("REVERSE:%s\n", str);
(gdb) c
Continuing.
REVERSE:dfea
[Inferior 1 (process 8016) exited normally]

```

图 15：断点调试过程

4.5. 修正程序并运行

任务 4.5 截图

```
#include <stdio.h>
#include <string.h>
#include "reverse.h"
/*****
int reverse(str)
    char *str;
{
    int i;
    int len;
    char c;
    len = strlen(str);
    for (i = 0; i < len / 2; i++) {
        c = *(str + i);
        *(str + i) = *(str + len - i - 1);
        *(str + len - i - 1) = c;
    }
    return 1;
}
int main(void)
{
    char str[1024];
    printf("Give me a word to reverse:\n");
    scanf("%s", str);
    reverse(str);
    printf("REVERSE:%s\n", str);
}
~
```

图 16：修正后程序代码

```
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ gcc -g reverse.c -o reverse
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$ ./reverse
Give me a word to reverse:
abcde
REVERSE:edcba
huangliangming_2022155028@ubuntu-2204:~/gdbdebug$
```

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;

<p>指导教师批阅意见：</p> <p>成绩评定：</p>	<p>指导教师签字：_____</p> <p>年 月 日</p>
--	--------------------------------------

<p>指导教师批阅意见：</p> <p>成绩评定：</p>	<p>指导教师签字：_____</p> <p>年 月 日</p>
--	--------------------------------------

<p>指导教师批阅意见：</p> <p>成绩评定：</p>	<p>指导教师签字：_____</p> <p>年 月 日</p>
--	--------------------------------------

<p>指导教师批阅意见:</p> <p>成绩评定:</p>	<p>指导教师签字:</p> <p>年 月 日</p>
--	---------------------------------

备注:	
-----	--

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。