

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：计算机与软件学院所有专业

指导教师：罗胜

报告人：詹耿羽 学号：2023193026 班级：数计

实验时间：2025.3.13

实验报告提交时间：2025.3.16

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

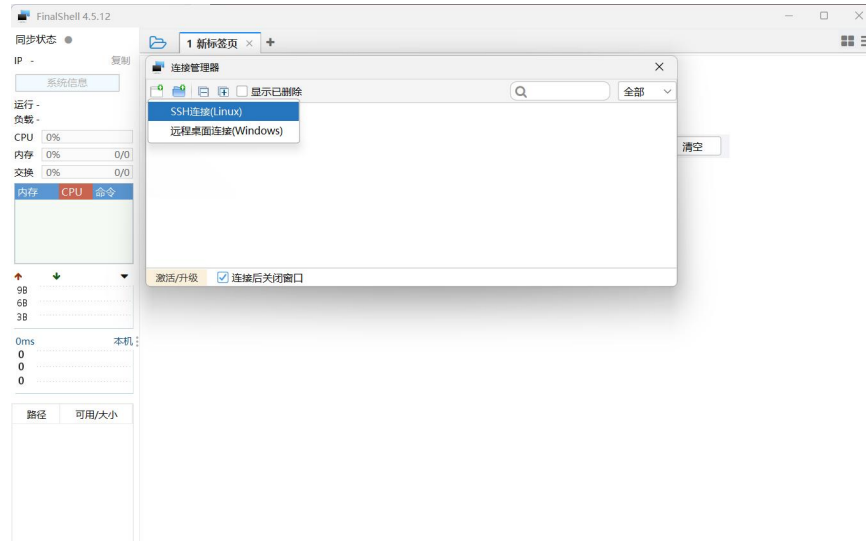
1. 个人电脑
2. Ubuntu 操作系统
3. gcc
4. gdb

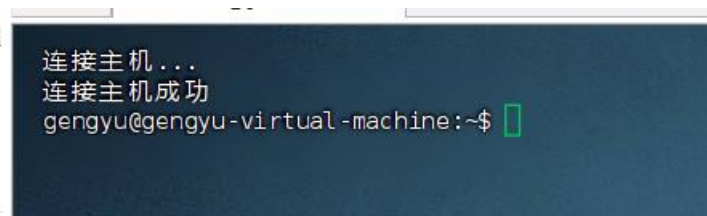
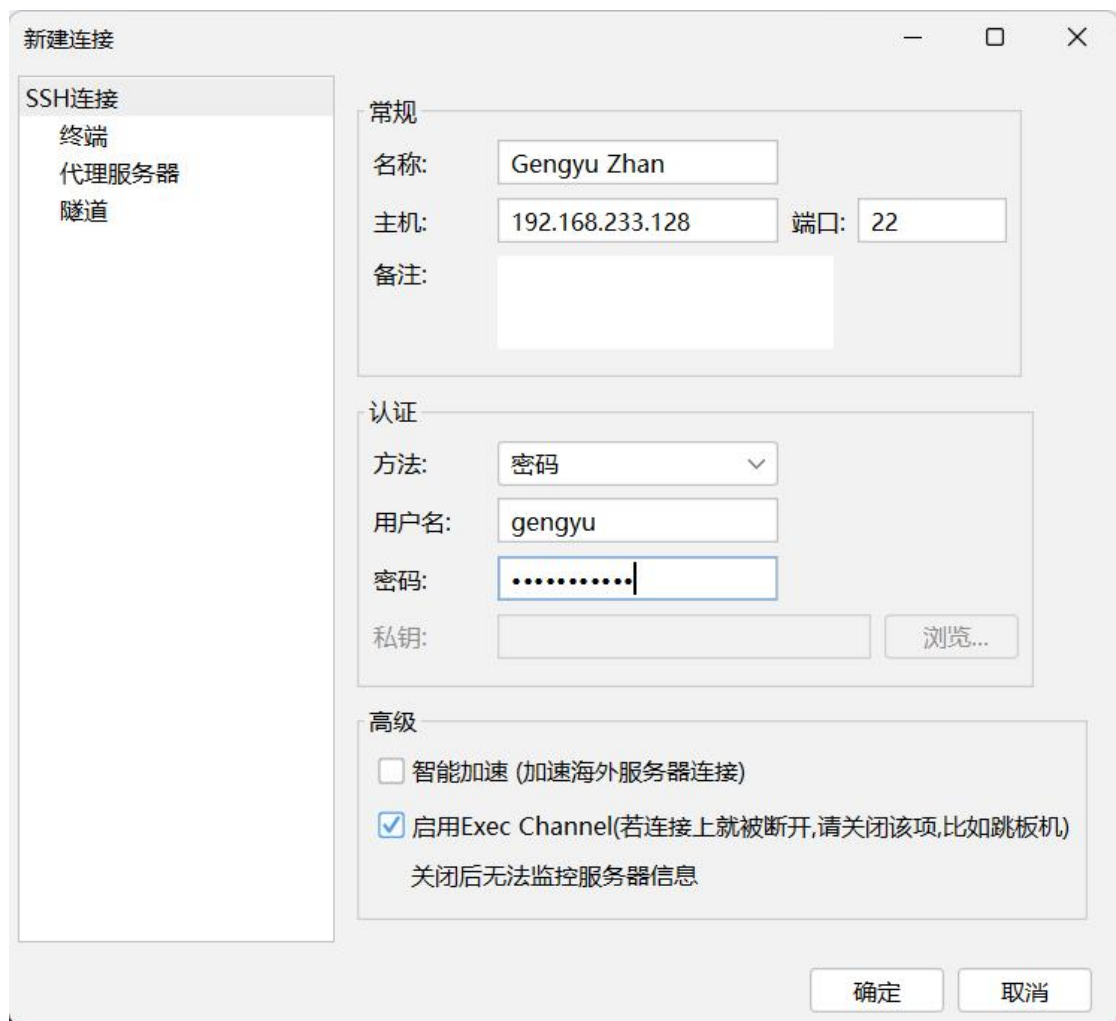
三、实验内容与步骤

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）

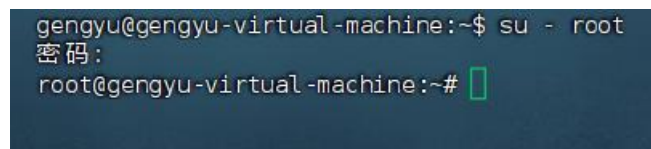
实验步骤：

- 1.1. 下面通过 Final Shell 的 SSH 连接操作 Ubuntu 的终端。





1.2.用”su - root”命令并输入密码切换到 root 用户.



1.3.使用以下命令创建一个新用户 “gengyu_2023193026”，并为其设置密码。
请注意，只有在设置了密码之后，用户才能被激活，否则将无法以该用户身份登录。

```

gengyu@gengyu-virtual-machine:~$ su - root
密码:
root@gengyu-virtual-machine:~# adduser gengyu_2023193026
正在添加用户 "gengyu_2023193026"...
正在添加新组 "gengyu_2023193026" (1001)...
正在添加新用户 "gengyu_2023193026" (1001) 到组 "gengyu_2023193026"...
创建主目录 "/home/gengyu_2023193026"...
正在从 "/etc/skel"复制文件...
新的 密码:
重新输入新的 密码:
passwd: 已成功更新密码
正在改变 gengyu_2023193026 的用户信息
请输入新值, 或直接敲回车键以使用默认值
    全名 []: gengyu_2023193026
    房间号码 []:
    工作电话 []:
    家庭电话 []:
    其它 []:
这些信息是否正确? [Y/n] y
root@gengyu-virtual-machine:~#

```

1.4.在 root 用户下可直接切换到另一个用户,无需输入密码.

```

root@gengyu-virtual-machine:~# su - gengyu_2023193026
gengyu_2023193026@gengyu-virtual-machine:~$ pwd
/home/gengyu_2023193026

```

2. 在新建的用户的 home 目录下创建子目录 gdbdebug,并进入 gdbdebug 子目录.

```

gengyu_2023193026@gengyu-virtual-machine:~$ mkdir gdbdebug
gengyu_2023193026@gengyu-virtual-machine:~$ cd gdbdebug

gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ pwd
/home/gengyu_2023193026/gdbdebug
gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$

```

2.1.用 Vim 编辑如下两个文件,即: reverse.h 和 reverse.c:

2.2.用 Vim 编辑 reverse.h.

```

gengyu_2023193026@gengyu-virtual-machine:~$ vim reverse.h

```

- 按 i 键进入插入模式 (任何键都可以使用, 不必特指 i 键盘, 这里仅以 i 键盘为例进行说明。), 输入如下代码, 声明函数 reverse()。

```
int reverse(char* str);  
  
-- 插入 -- 6,1 全部
```

- 按 ESC 键回到命令行模式,按:键进入末行模式,输入 wq 保存并退出.

```
int reverse(char* str);  
  
:wq
```

- 若出现如下提示,则表示当前用户无权限修改该文件.为方便后续操作,用如下的方法给当前用户配置 sudo 认证.

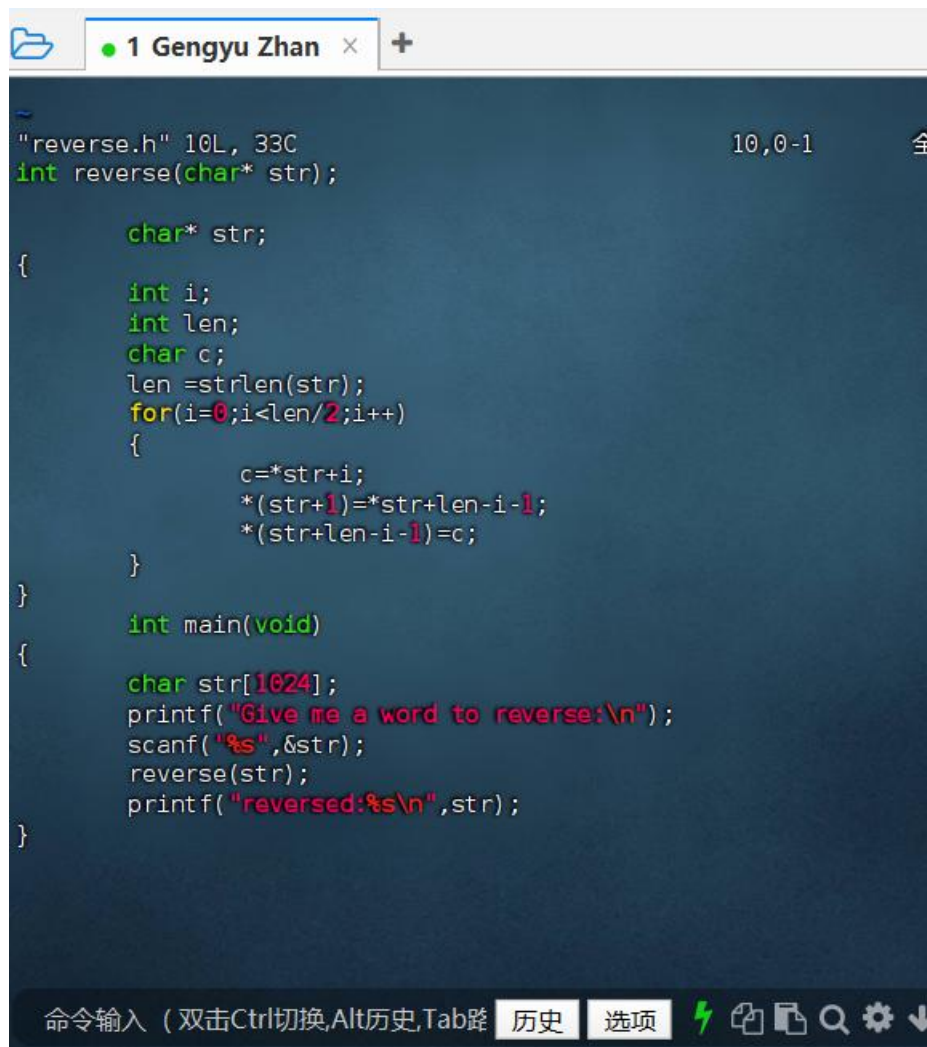
```
E45: 'readonly' option is set (add ! to override)
```

- 回到 root 用户,用 visudo 命令打开配置 sudo 认证的文件,在文件末加一

行,Ctrl+O 保存,Ctrl+X 退出.

- 回到新建用户,重新执行操作 2.2 的第二步即可.

2.3.用 Vim 编辑 reverse.c,保存并退出.




```
"reverse.h" 10L, 33C                                10,0-1    全
int reverse(char* str);

    char* str;
{
    int i;
    int len;
    char c;
    len = strlen(str);
    for(i=0;i<len/2;i++)
    {
        c=*str+i;
        *(str+1)=*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("reversed:%s\n",str);
}
```

2.4.按以下步骤编译.

- 发现 reverse.c 和 reverse.h 不在 gdbdebug 目录下,按如下步骤将其移动至 gdbdebug 目录下.



```
wangxi_2021192010@hyUbuntu:~$ ls
gdbdebug reverse.c reverse.h
wangxi_2021192010@hyUbuntu:~$ mv reverse.c gdbdebug
wangxi_2021192010@hyUbuntu:~$ mv reverse.h gdbdebug
wangxi_2021192010@hyUbuntu:~$ ls
gdbdebug
wangxi_2021192010@hyUbuntu:~$ cd gdbdebug
wangxi_2021192010@hyUbuntu:~/gdbdebug$ ls
reverse.c reverse.h
wangxi_2021192010@hyUbuntu:~/gdbdebug$
```



```
gengyu_2023193026@gengyu-virtual-machine:~$ ls
gdbdebug reverse.c reverse.h
gengyu_2023193026@gengyu-virtual-machine:~$ mv reverse.c gdbdebug
```

```
gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ ls
reverse.c reverse.h
```

```
gengyu_2023193026@gengyu-virtual-machine:~$ mv reverse.h gdbdebug
gengyu_2023193026@gengyu-virtual-machine:~$ cd gdbdebug
```

- 用如下命令编译 reverse.c。（需要事先进入到 root 装 gcc）

```
root@gengyu-virtual-machine:~# apt install gcc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
```

\$gcc -Wall reverse.c -o reverse

- 发现有下列错误信息.

```
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:23:17: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[1024]' [-Wformat=]
 23 |         scanf("%s", &str);
    |                ^~ ~~~~
```

```
reverse.c: In function 'main':
reverse.c:23:17: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[1024]' [-Wformat=]
 23 |         scanf("%s", &str);
    |                ^~ ~~~~
    |                | |
    |                | char (*)[1024]
    |                char *
reverse.c: In function 'reverse':
reverse.c:17:1: warning: control reaches end of non-void function [-Wreturn-type]
 17 | }
    | ^
```

- 修改代码至可正常编译.

```
void reverse(char* str);
```

```

#include <stdio.h>
#include <string.h>
#include "reverse.h"

void reverse(char* str) {
    char c;
    int len = strlen(str);
    for(int i = 0; i < len / 2; i++) {
        c = *str + i;
        *(str + 1) = *str + len - i - 1;
        *(str + len - i - 1) = c;
    }
}

int main() {
    char str[1024];
    printf("Give me a word to reverse:\n");
    scanf("%s", str);
    reverse(str);
    printf("REVERSED:%s\n", str);
    return 0;
}

```

```

gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ gcc -Wall revers
e.c -o reverse

```

- 运行程序并输入测试字符串,发现程序可正常运行,但输出错误.

```

Give me a word to reverse:
abcde
REVERSED:adcba

```

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

3.1. 编译时加入调试信息.

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

```

gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ gcc -g reverse.c
-o reversel

```

3.2. 启用 GDB 调试.

```

gengyu_2023193026@gengyu-virtual-machine:~/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 ./reverse1...
(gdb)

```

3.3. 键入 list, 查看源代码并根据行号/函数名设置断点.

```

(gdb) list
1      #include <stdio.h>
2      #include <string.h>
3      #include "reverse.h"
4
5      void reverse(char* str) {
6          char c;
7          int len = strlen(str);
8          for(int i = 0; i < len / 2; i++) {
9              c = *str + i;
10             *(str + 1) = *str + len - i - 1;
(gdb) list
11             *(str + len - i - 1) = c;
12         }
13     }
14
15     int main() {
16         char str[1024];
17         printf("Give me a word to reverse:\n");
18         scanf("%s", str);
19         reverse(str);
20         printf("REVERSED:%s\n", str);
(gdb) list
21         return 0;
22     }
(gdb)

```

```

(gdb) break reverse
Note: breakpoint 1 also set at pc 0x555555551d9.
Breakpoint 2 at 0x555555551d9: file reverse.c, line 7.
(gdb) break 10
Breakpoint 3 at 0x55555555202: file reverse.c, line 10.
(gdb) break 11
Breakpoint 4 at 0x55555555222: file reverse.c, line 11.
(gdb) info break
Num      Type           Disp Enb Address              What
1        breakpoint    keep y   0x0000555555551d9 in reverse at reverse.c:7
          breakpoint already hit 1 time
2        breakpoint    keep y   0x0000555555551d9 in reverse at reverse.c:7
3        breakpoint    keep y   0x000055555555202 in reverse at reverse.c:10
4        breakpoint    keep y   0x000055555555222 in reverse at reverse.c:11
(gdb) █

```

3. 4. 观察变量值并作分析, 推测错误.

```

Give me a word to reverse:
abcde

Breakpoint 1, reverse (str=0x7fffffffdf30 "abcde") at reverse.c:7
7          int len = strlen(str);
(gdb) c
Continuing.

Breakpoint 3, reverse (str=0x7fffffffdf30 "abcde") at reverse.c:10
10          *(str + 1) = *str + len - i - 1;
(gdb) c
Continuing.

Breakpoint 4, reverse (str=0x7fffffffdf30 "aecde") at reverse.c:11
11          *(str + len - i - 1) = c;
(gdb) c
Continuing.

Breakpoint 3, reverse (str=0x7fffffffdf30 "aecda") at reverse.c:10
10          *(str + 1) = *str + len - i - 1;
(gdb) c
Continuing.

Breakpoint 4, reverse (str=0x7fffffffdf30 "adcda") at reverse.c:11
11          *(str + len - i - 1) = c;
(gdb) c
Continuing.
REVERSED:adcba
[Inferior 1 (process 145603) exited normally]
(gdb) █

```

3. 5. 发现错误后退出调试并修改代码.

- 按下 quit, 退出。

```
(gdb) quit
```

- 附上 gdb 另外的使用说明:

```
(gdb) help
List of classes of commands:

aliases -- Aliases of other commands.
breakpoints -- Making program stop at certain points.
data -- Examining data.
files -- Specifying and examining files.
internals -- Maintenance commands.
obscure -- Obscure features.
running -- Running the program.
stack -- Examining the stack.
status -- Status inquiries.
support -- Support facilities.
tracepoints -- Tracing of program execution without stopping the program.
user-defined -- User-defined commands.
```

- 最终调整代码:

```
void reverse(char* str) {
    char c;
    int len = strlen(str);
    for(int i = 0; i < len / 2; i++) {
        c = *(str + i);
        *(str + i) = *(str + len - i - 1);
        *(str + len - i - 1) = c;
    }
}
```

四、实验结果

编译并运行 reverse.c 后, 发现程序能够顺利编译, 运行结果正确, 实验任务圆满完成。

```
gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ gcc -Wall reverse.c -o reverse
gengyu_2023193026@gengyu-virtual-machine:~/gdbdebug$ ./reverse
```

```
Give me a word to reverse:
abcde
REVERSED: edcba
```

五、实验总结与体会

在这次的实验过程中, 我首次尝试在 Linux 操作系统上进行操作, 这对我来说是一个全新的体验。在实际操作的过程中, 我遇到了一些挑战, 比如对 Linux 系统中的各种指令不够熟悉, 以及对用户权限的管理方面也存在一些困惑。面对

这些困难，我并没有放弃，而是通过查阅大量的相关资料和文档，逐步克服了这些难题。通过这次实验，我不仅熟练掌握了 Linux 系统中一些常用的指令，还加深了对 Linux 用户权限管理的理解。此外，我还学习了如何使用 Vim 编辑器来编辑 C 语言的源文件，并且掌握了如何通过命令行来编译和运行 C 程序。更进一步，我学会了如何在命令行环境下使用 gdb 调试工具来设置断点，并进行程序调试。这次实验让我对各种集成开发环境（IDE）的底层工作原理有了更加清晰的认识。实际上，我意识到各种 IDE 的功能无非是将原本需要手动输入的命令行操作自动化，并且通过图形化界面的方式更加直观地展示给用户，从而提高开发效率。

指导教师批阅意见:

成绩评定：

指导教师签字:

年 月 日

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。