南京郵電大學

实验报告

（ 2024 / 2025 学年 第 一 学期）

专业 信息安全

学生姓名 潘韬

班级学号 B22041622

指导教师 王磊

Experiment3 C programming experiment

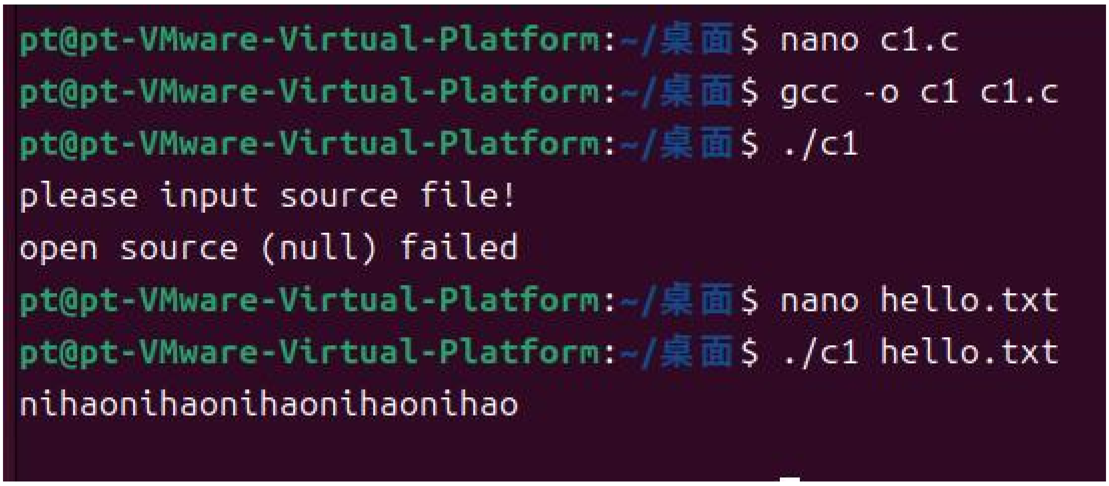
Experimental purpose:

Further use the basic syntax of C programming language in Linux system, deepen the understanding of the knowledge.

(1) Task 1

(1) Write a C program that uses standard I/O libraries to display the contents of text files. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile file.

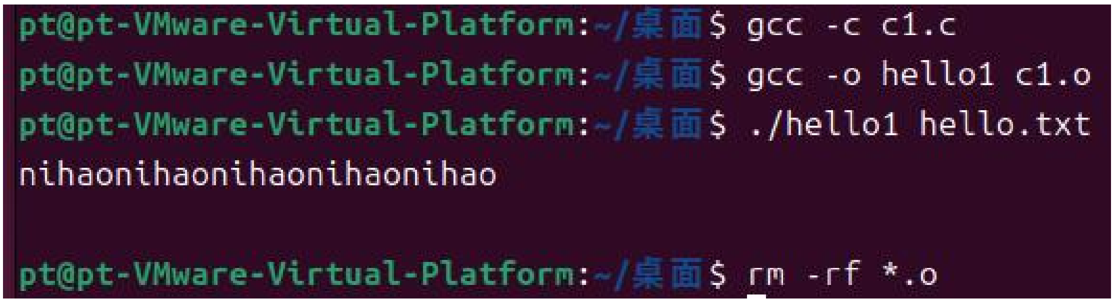
```
#include <stdio.h>
int main(int argc, char* argv[])
{
char buf[1024] = { 0 };
FILE* fp = fopen(argv[1],"r");
if (argc < 2)
{
printf("please input source file!\n");
}
if (fp == NULL)
{
printf("open source %s failed\n", argv[1]);
return -1;
}
while (fgets(buf,1024, fp))
{
printf("%s\n", buf);
}
return 0;
}
```



```
pt@pt-VMware-Virtual-Platform:~/桌面$ nano c1.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -o c1 c1.c
pt@pt-VMware-Virtual-Platform:~/桌面$ ./c1
please input source file!
open source (null) failed
pt@pt-VMware-Virtual-Platform:~/桌面$ nano hello.txt
pt@pt-VMware-Virtual-Platform:~/桌面$ ./c1 hello.txt
nihaonihaonihaonihaonihao
```

Make sure your filename is c1.c
We can use the following makefile.

```
hello1:c1.o
gcc -o hello1 c1.o
c1.o:c1.c
gcc -c c1.c
clean:
rm -rf *.o
```



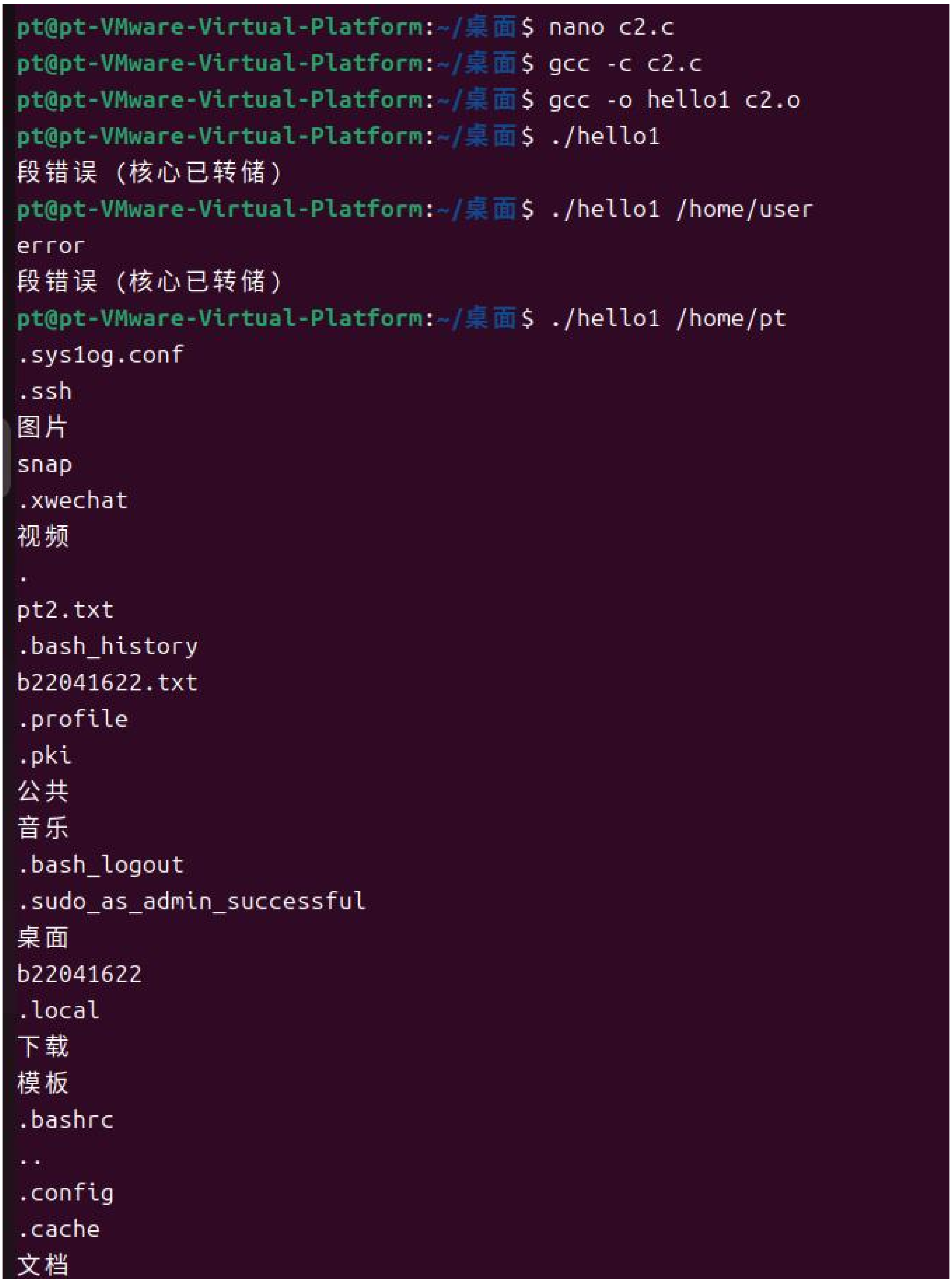
```
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -c c1.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -o hello1 c1.o
pt@pt-VMware-Virtual-Platform:~/桌面$ ./hello1 hello.txt
nihaonihaonihaonihaonihao

pt@pt-VMware-Virtual-Platform:~/桌面$ rm -rf *.o
```

(2) Task 2

(2) Write a C program that displays all the file names in the current directory. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile file.

```
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>
int main(int argc, char* argv[])
{
DIR* dirp;
struct dirent* direntp;
if ((dirp = opendir(argv[1])) == NULL) {
```

```
printf("error\n");
// exit(1);
}
while ((direntp = readdir(dirp)) != NULL)
printf("%s\n", direntp->d_name);
closedir(dirp);
// exit(0);
}
```

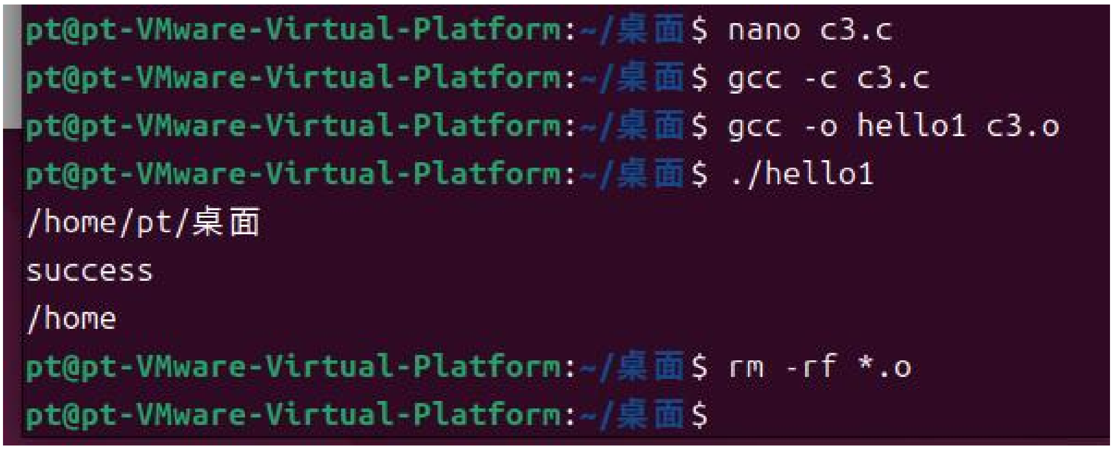


```
pt@pt-VMware-Virtual-Platform:~/桌面$ nano c2.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -c c2.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -o hello1 c2.o
pt@pt-VMware-Virtual-Platform:~/桌面$ ./hello1
段错误 (核心已转储)
pt@pt-VMware-Virtual-Platform:~/桌面$ ./hello1 /home/user
error
段错误 (核心已转储)
pt@pt-VMware-Virtual-Platform:~/桌面$ ./hello1 /home/pt
.sys1og.conf
.ssh
图片
snap
.xwechat
视频
.
pt2.txt
.bash_history
b22041622.txt
.profile
.pki
公共
音乐
.bash_logout
.sudo_as_admin_successful
桌面
b22041622
.local
下载
模板
.bashrc
..
.config
.cache
文档
```

(3) Task 3

(3) Write a C program that changes the working directory of the current process. The program is compiled
and linked by the make tool, which requires the generation of the.o file first, and then the generation of the
executable file, and the function of deleting the intermediate file (.o) in the makefile file.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(){
char buf[1024] = {0}; char buf2[1024]={0};
getcwd(buf, 1024);
printf("%s\n", buf);
if(chdir("/home")<0){
printf("error\n");
}
else
{
printf("success\n");
}
getcwd(buf2,1024);
printf("%s\n",buf2);
return 0;
}
```



```
pt@pt-VMware-Virtual-Platform:~/桌面$ nano c3.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -c c3.c
pt@pt-VMware-Virtual-Platform:~/桌面$ gcc -o hello1 c3.o
pt@pt-VMware-Virtual-Platform:~/桌面$ ./hello1
/home/pt/桌面
success
/home
pt@pt-VMware-Virtual-Platform:~/桌面$ rm -rf *.o
pt@pt-VMware-Virtual-Platform:~/桌面$
```