

南京邮电大学

实验报告

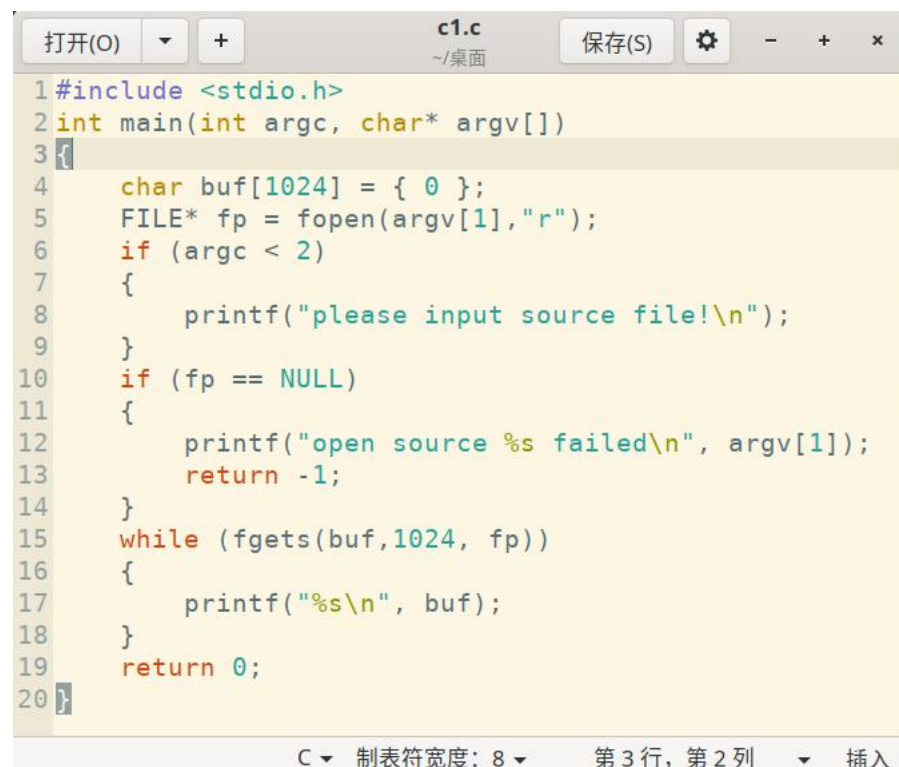
(2024/ 2025 学年 第 一 学期)

课程名称	GNU/Linux 编程			
实验名称	实验三			
实验时间	2024	年 11	月 15	日
指导单位	计算机学院 网络空间安全系			
指导教师	王磊			

学生姓名	龙海阔	班级学号	B22041012
学院(系)	计算机学院	专 业	信息安全

Task 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.

A screenshot of a C code editor window. The window has a title bar with 'c1.c' and a path '~/.桌面'. It includes buttons for '打开(O)', '保存(S)', and a settings gear icon. The code is as follows:

```
1#include <stdio.h>
2int main(int argc, char* argv[])
3{
4    char buf[1024] = { 0 };
5    FILE* fp = fopen(argv[1], "r");
6    if (argc < 2)
7    {
8        printf("please input source file!\n");
9    }
10   if (fp == NULL)
11   {
12       printf("open source %s failed\n", argv[1]);
13       return -1;
14   }
15   while (fgets(buf, 1024, fp))
16   {
17       printf("%s\n", buf);
18   }
19   return 0;
20 }
```

The status bar at the bottom shows 'C', '制表符宽度: 8', '第 3 行, 第 2 列', and '插入'.

Make sure your filename is c1.c

We can use the following makefile.



```
1 hello1: c1.o
2     gcc -o hello1 c1.o
3 c1.o: c1.c
4     gcc -c c1.c
5 clean:
6     rm -rf *.o
```

Makefile 制表符宽度: 8 第 2 行, 第 27 列 插入

Run the results:



```
Terminal - pwn@CTF: ~/桌面
桌面$ make
gcc -c c1.c
gcc -o hello1 c1.o
桌面$ ./hello1 B22041012
B22041012龙海阔
```

Task 2

Write a C program that displays all the file names in the current directory.

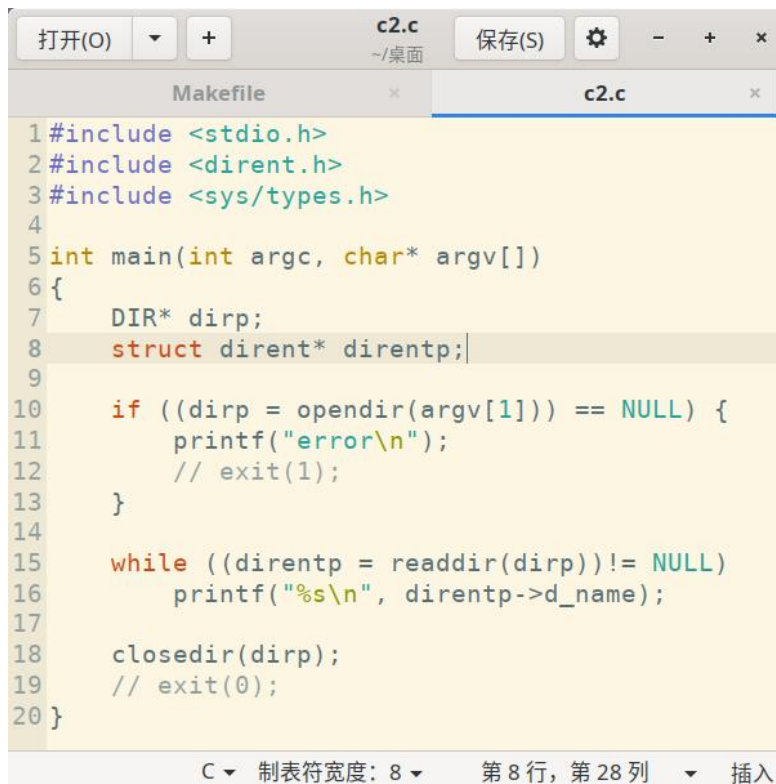
The program is compiled and

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.

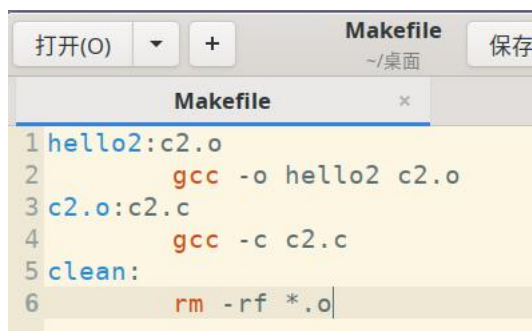


```
1#include <stdio.h>
2#include <dirent.h>
3#include <sys/types.h>
4
5int main(int argc, char* argv[])
6{
7    DIR* dirp;
8    struct dirent* direntp;
9
10    if ((dirp = opendir(argv[1])) == NULL) {
11        printf("error\n");
12        // exit(1);
13    }
14
15    while ((direntp = readdir(dirp)) != NULL)
16        printf("%s\n", direntp->d_name);
17
18    closedir(dirp);
19    // exit(0);
20}
```

C 制表符宽度: 8 第 8 行, 第 28 列 插入

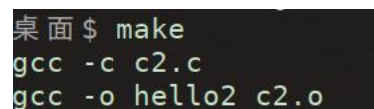
Make sure your filename is c2.c

We can use the following makefile.



```
1hello2:c2.o
2    gcc -o hello2 c2.o
3c2.o:c2.c
4    gcc -c c2.c
5clean:
6    rm -rf *.o
```

Run the results:

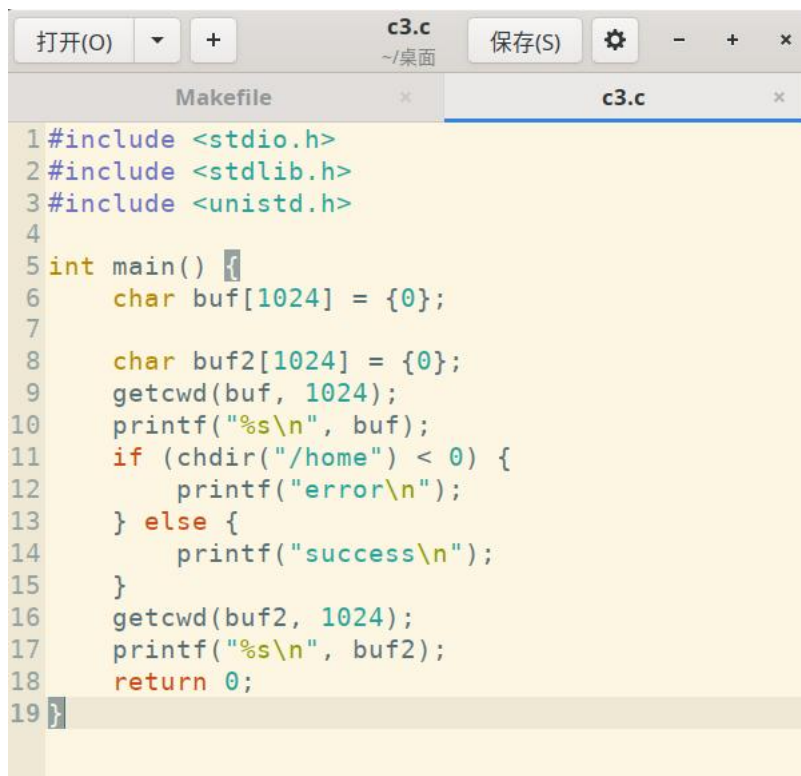


```
桌面 $ make
gcc -c c2.c
gcc -o hello2 c2.o
```

```
桌面$ ./hello2 .  
c2.o  
ret2libc (2)  
shellcode-lv1  
题目  
a  
shh  
hello2  
..  
c2.c  
srop-2024  
.  
Makefile  
05  
B22041012
```

Task 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.



```
1#include <stdio.h>
2#include <stdlib.h>
3#include <unistd.h>
4
5int main() {
6    char buf[1024] = {0};
7
8    char buf2[1024] = {0};
9    getcwd(buf, 1024);
10   printf("%s\n", buf);
11   if (chdir("/home") < 0) {
12       printf("error\n");
13   } else {
14       printf("success\n");
15   }
16   getcwd(buf2, 1024);
17   printf("%s\n", buf2);
18   return 0;
19 }
```

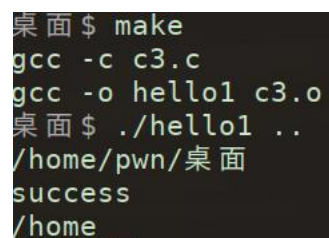
Make sure your filename is c3.c

We can use the following makefile.



```
1hello3:c3.o
2    gcc -o hello1 c3.o
3c3.o:c3.c
4    gcc -c c3.c
5clean:
6    rm -rf *.o
7
```

Run the results:



```
桌面$ make
gcc -c c3.c
gcc -o hello1 c3.o
桌面$ ./hello1 ..
/home/pwn/桌面
success
/home
```