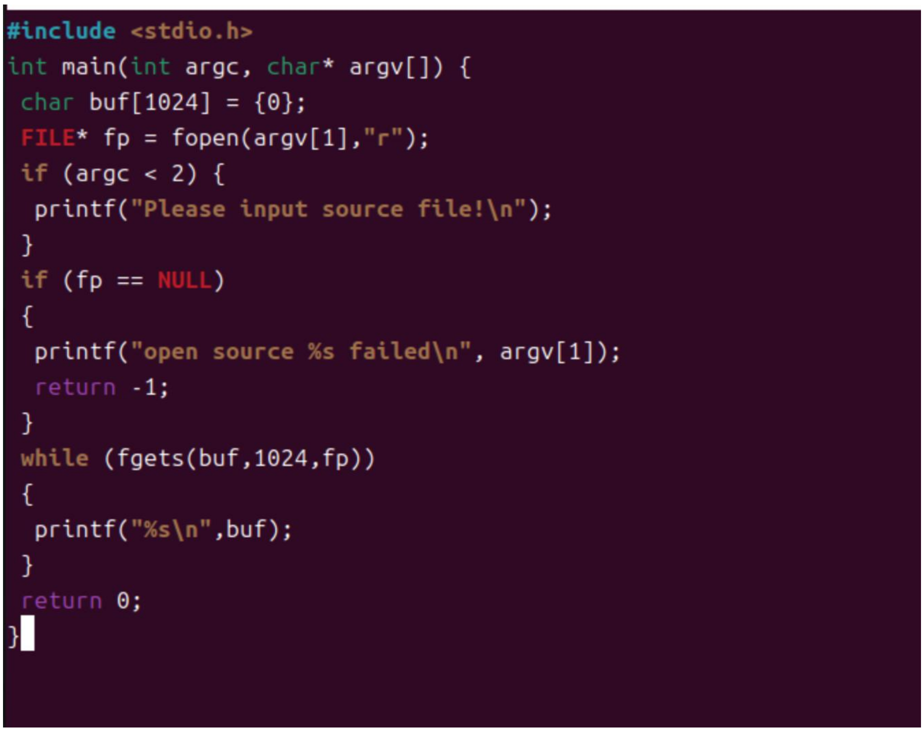
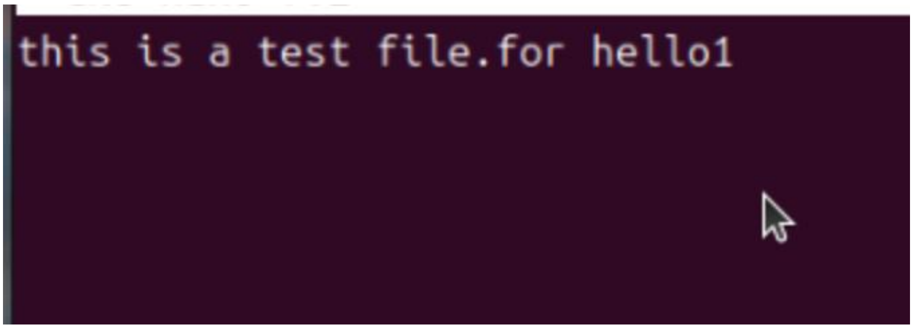
(1) Task 1

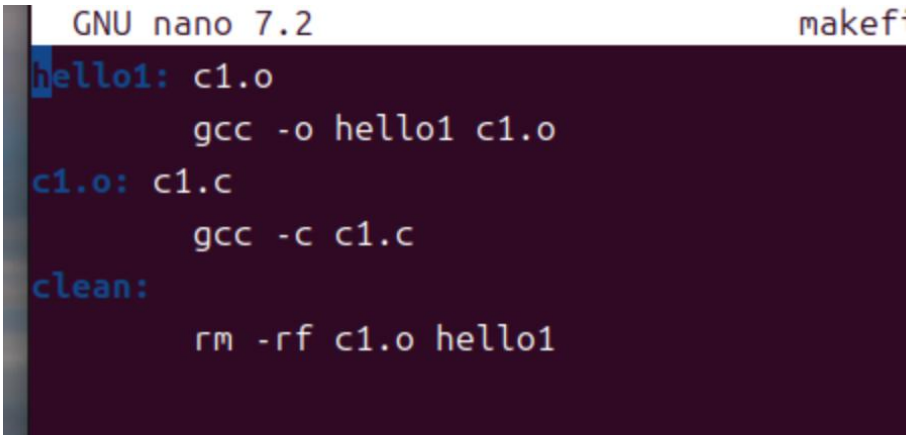
(1) Write a C program that uses standard I/O libraries to display the contents of text files. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o)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



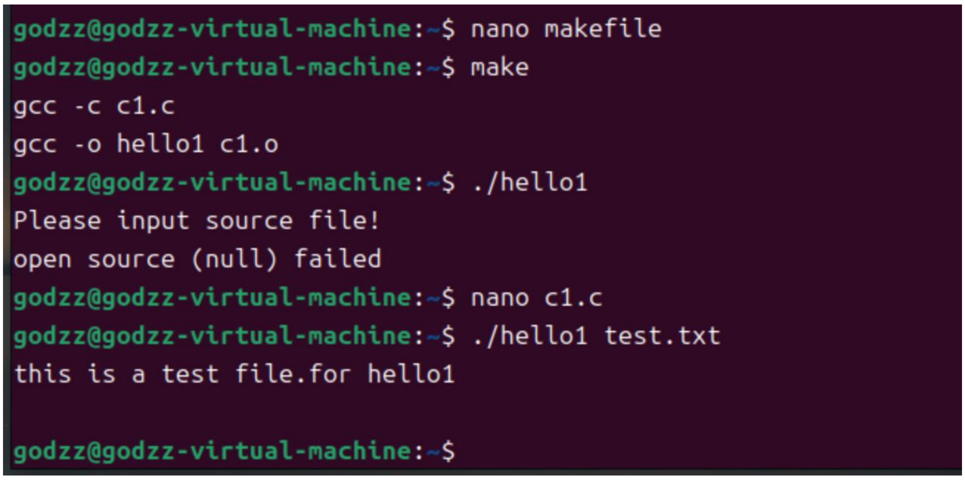
```c
#include <stdio.h>
int main(int argc, char* argv[]) {
 char buf[1024] = {0};
 FILE* fp = fopen(argv[1],"r");
 if (argc < 2) {
  printf("Please input source file!\n");
 }
 if (fp == NULL)
 {
  printf("open source %s failed\n", argv[1]);
  return -1;
 }
 while (fgets(buf,1024,fp))
 {
  printf("%s\n",buf);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



```
this is a test file.for hello1
```



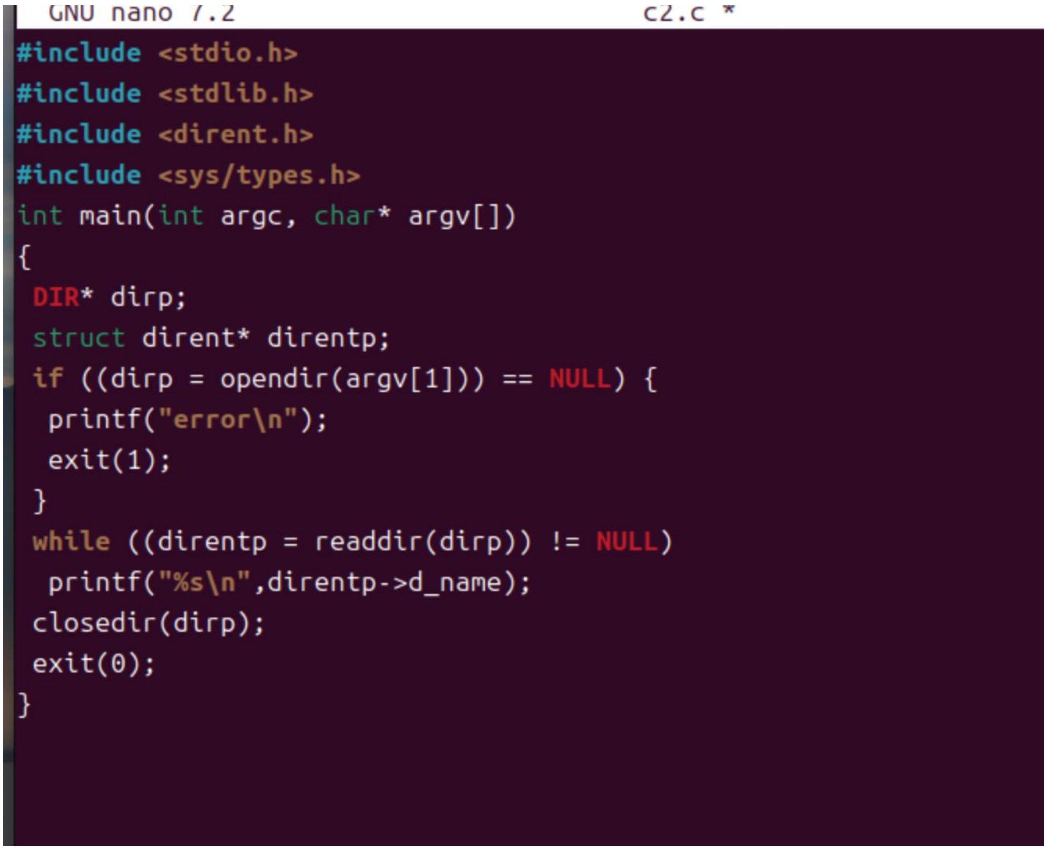
```makefile
hello1: c1.o
	gcc -o hello1 c1.o
c1.o: c1.c
	gcc -c c1.c
clean:
	rm -rf c1.o hello1
```



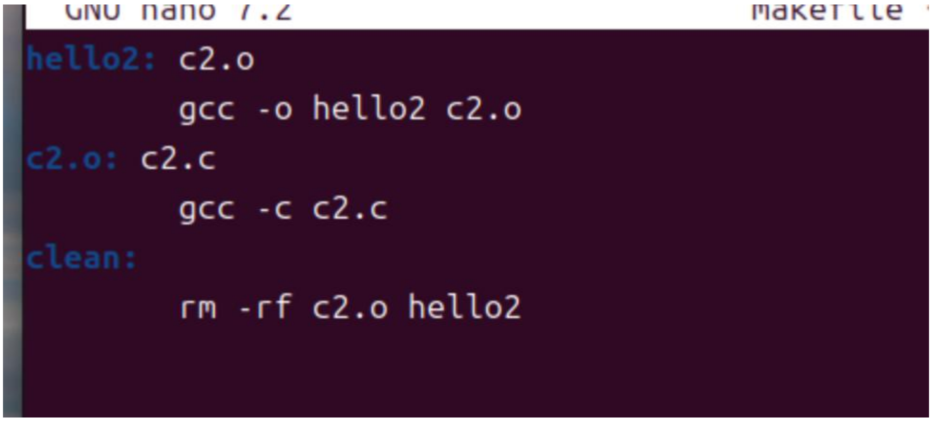
```
godzz@godzz-virtual-machine:~$ nano makefile
godzz@godzz-virtual-machine:~$ make
gcc -c c1.c
gcc -o hello1 c1.o
godzz@godzz-virtual-machine:~$ ./hello1
Please input source file!
open source (null) failed
godzz@godzz-virtual-machine:~$ nano c1.c
godzz@godzz-virtual-machine:~$ ./hello1 test.txt
this is a test file.for hello1

godzz@godzz-virtual-machine:~$
```

(2) Task 2

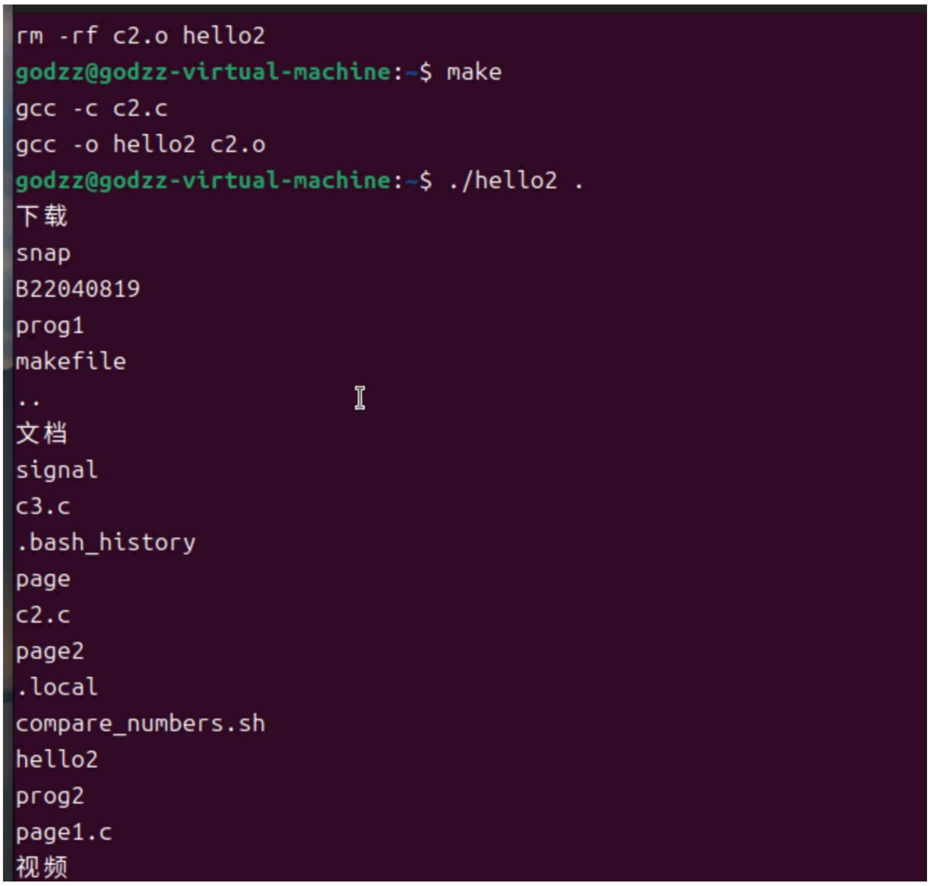
(2) Write a C program that displays all the file names in the current directory. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile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



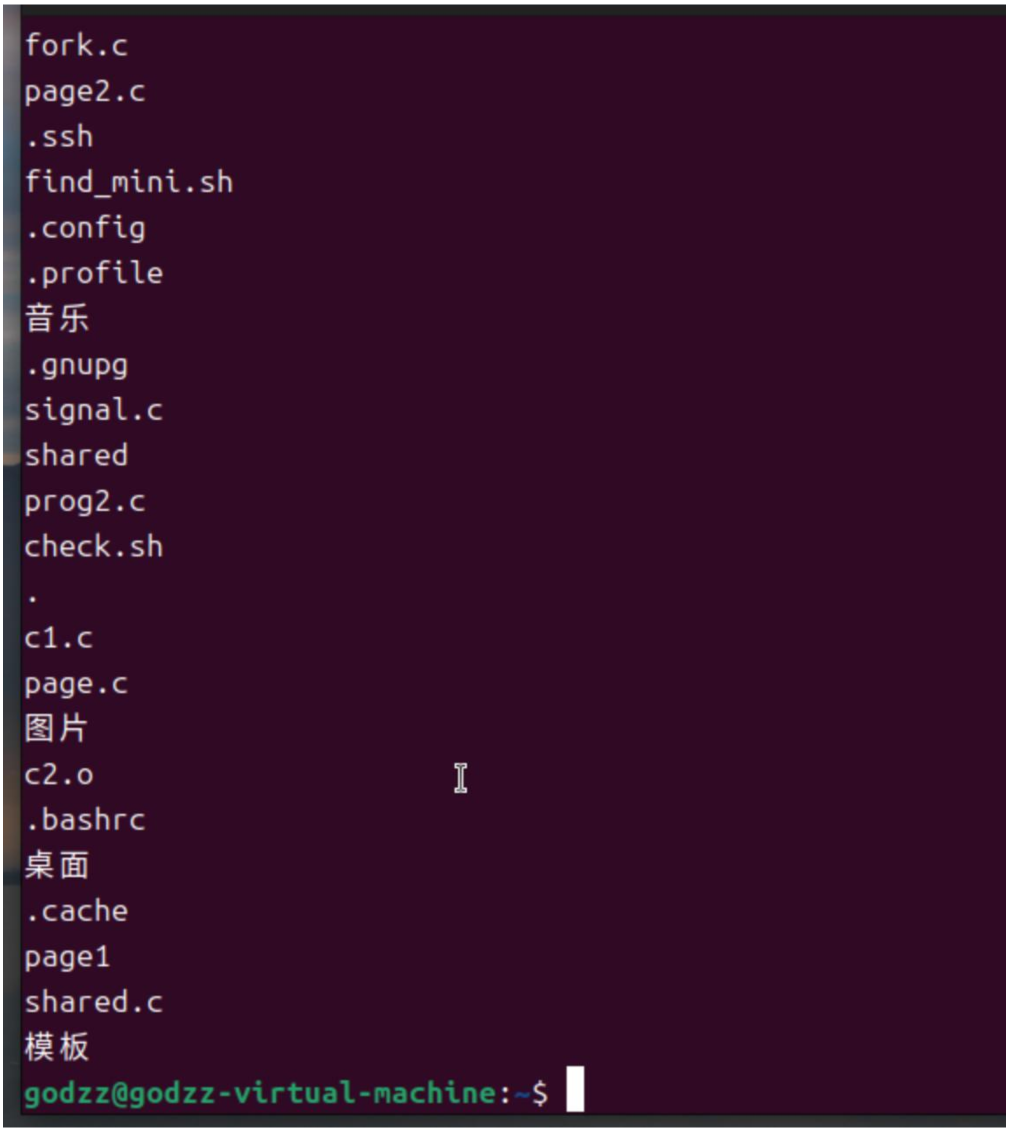
```
  GNU nano 7.2                              c2.c *
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/types.h>
int main(int argc, char* argv[])
{
 DIR* dirp;
 struct dirent* direntp;
 if ((dirp = opendir(argv[1])) == NULL) {
  printf("error\n");
  exit(1);
 }
 while ((direntp = readdir(dirp)) != NULL)
  printf("%s\n",direntp->d_name);
 closedir(dirp);
 exit(0);
}
```



```
hello2: c2.o
	gcc -o hello2 c2.o
c2.o: c2.c
	gcc -c c2.c
clean:
	rm -rf c2.o hello2
```



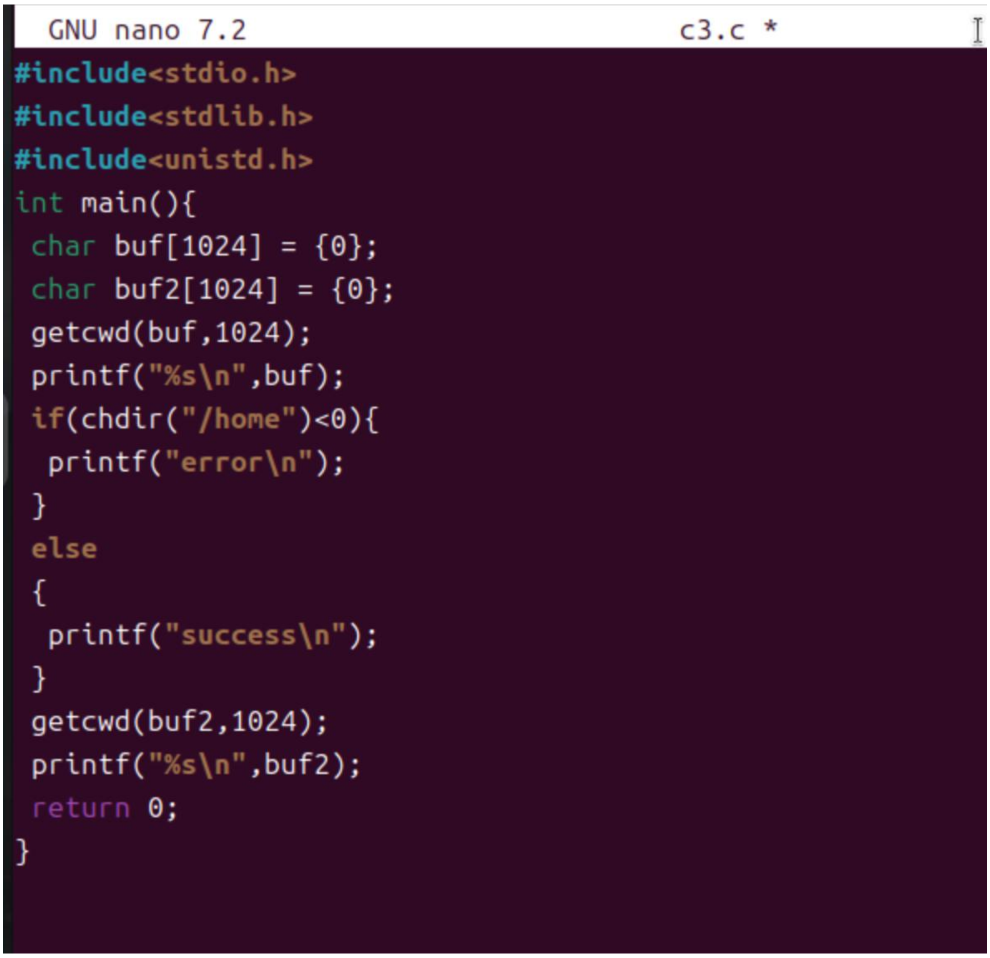
```
rm -rf c2.o hello2
godzz@godzz-virtual-machine:~$ make
gcc -c c2.c
gcc -o hello2 c2.o
godzz@godzz-virtual-machine:~$ ./hello2 .
下载
snap
B22040819
prog1
makefile
..
文档
signal
c3.c
.bash_history
page
c2.c
page2
.local
compare_numbers.sh
hello2
prog2
page1.c
视频
```



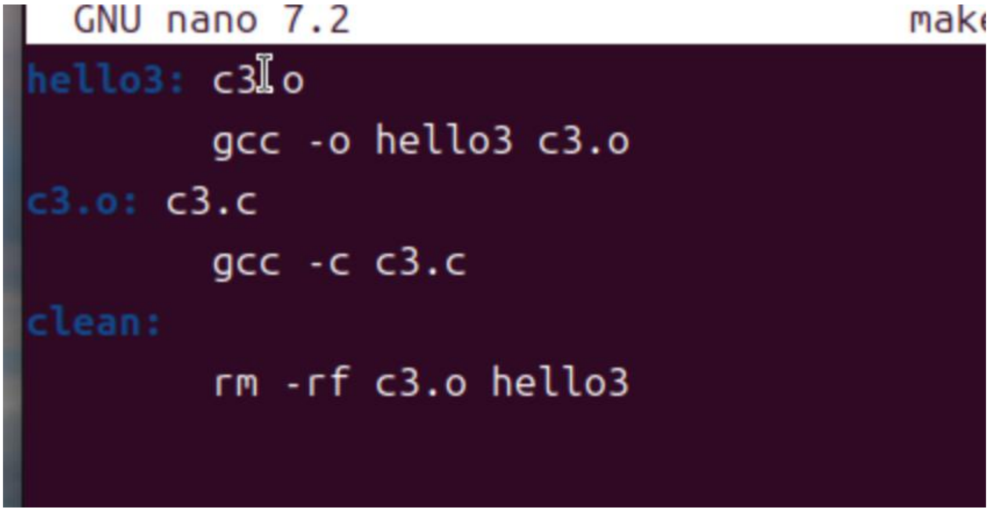
```
fork.c
page2.c
.ssh
find_mini.sh
.config
.profile
音乐
.gnupg
signal.c
shared
prog2.c
check.sh
.
c1.c
page.c
图片
c2.o
.bashrc
桌面
.cache
page1
shared.c
模板
godzz@godzz-virtual-machine:~$ 
```

(3) Task 3

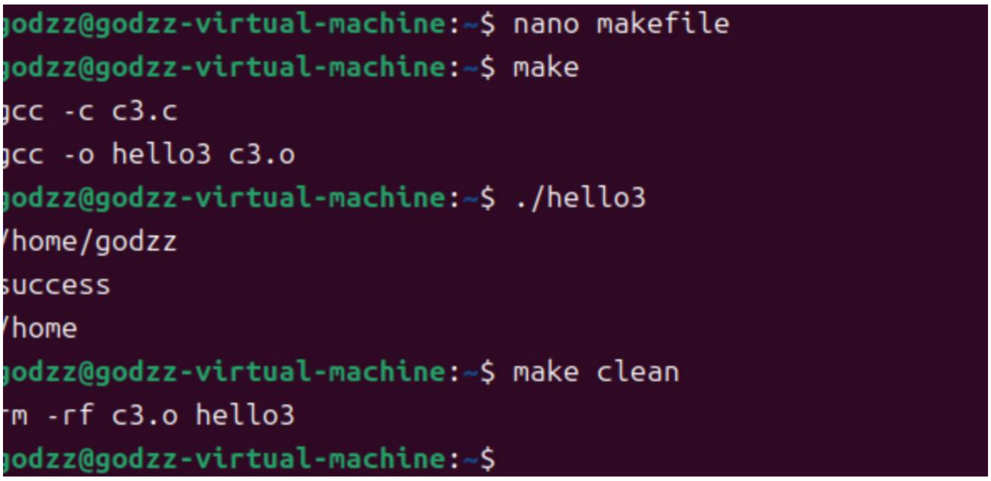
(3) Write a C program that changes the working directory of the current process. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile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



```c
#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;
}
```



```makefile
hello3: c3.o
	gcc -o hello3 c3.o
c3.o: c3.c
	gcc -c c3.c
clean:
	rm -rf c3.o hello3
```



```
godzz@godzz-virtual-machine:~$ nano makefile
godzz@godzz-virtual-machine:~$ make
gcc -c c3.c
gcc -o hello3 c3.o
godzz@godzz-virtual-machine:~$ ./hello3
/home/godzz
success
/home
godzz@godzz-virtual-machine:~$ make clean
rm -rf c3.o hello3
godzz@godzz-virtual-machine:~$
```