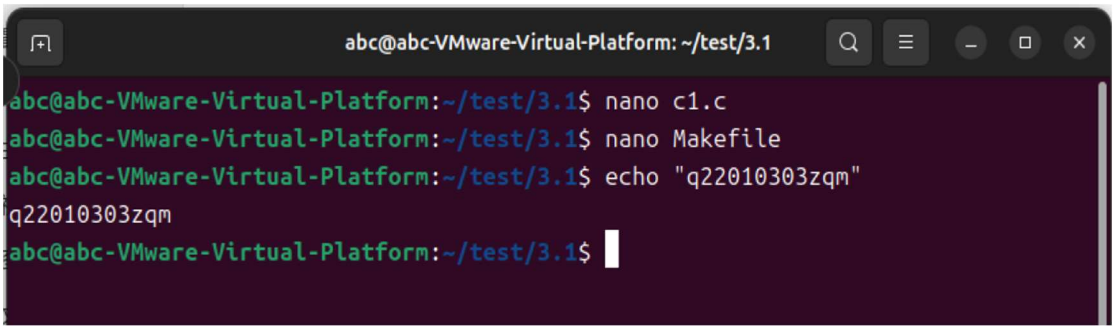
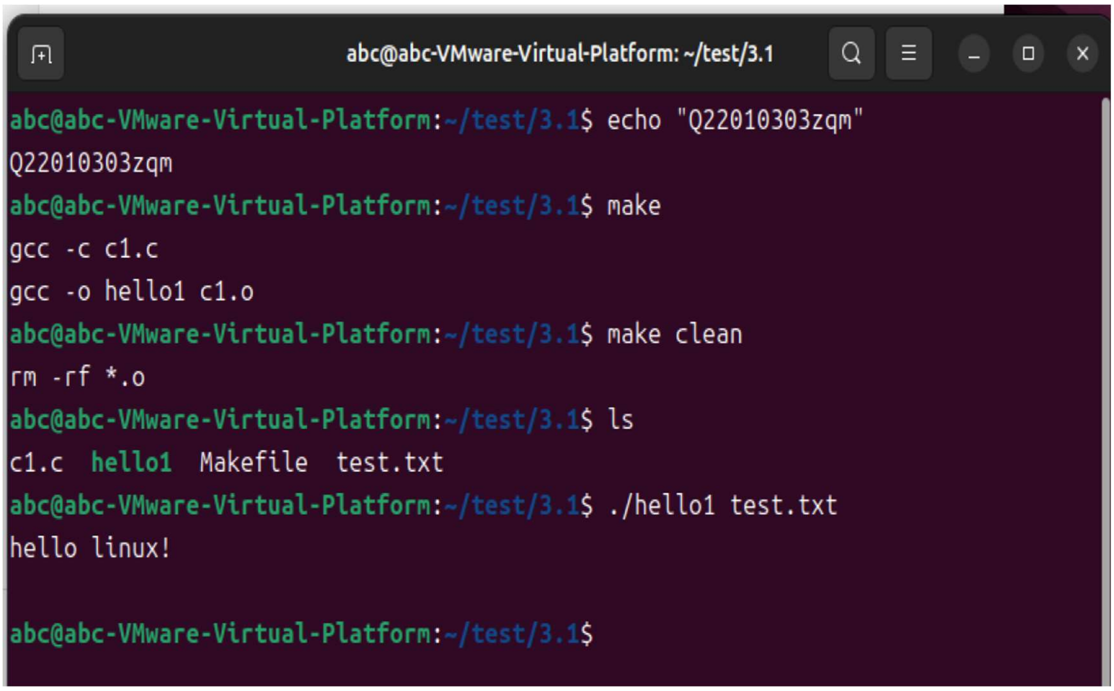
(1) Task 1

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.



```
abc@abc-VMware-Virtual-Platform:~/test/3.1$ nano c1.c
abc@abc-VMware-Virtual-Platform:~/test/3.1$ nano Makefile
abc@abc-VMware-Virtual-Platform:~/test/3.1$ echo "q22010303zqm"
q22010303zqm
abc@abc-VMware-Virtual-Platform:~/test/3.1$ 
```



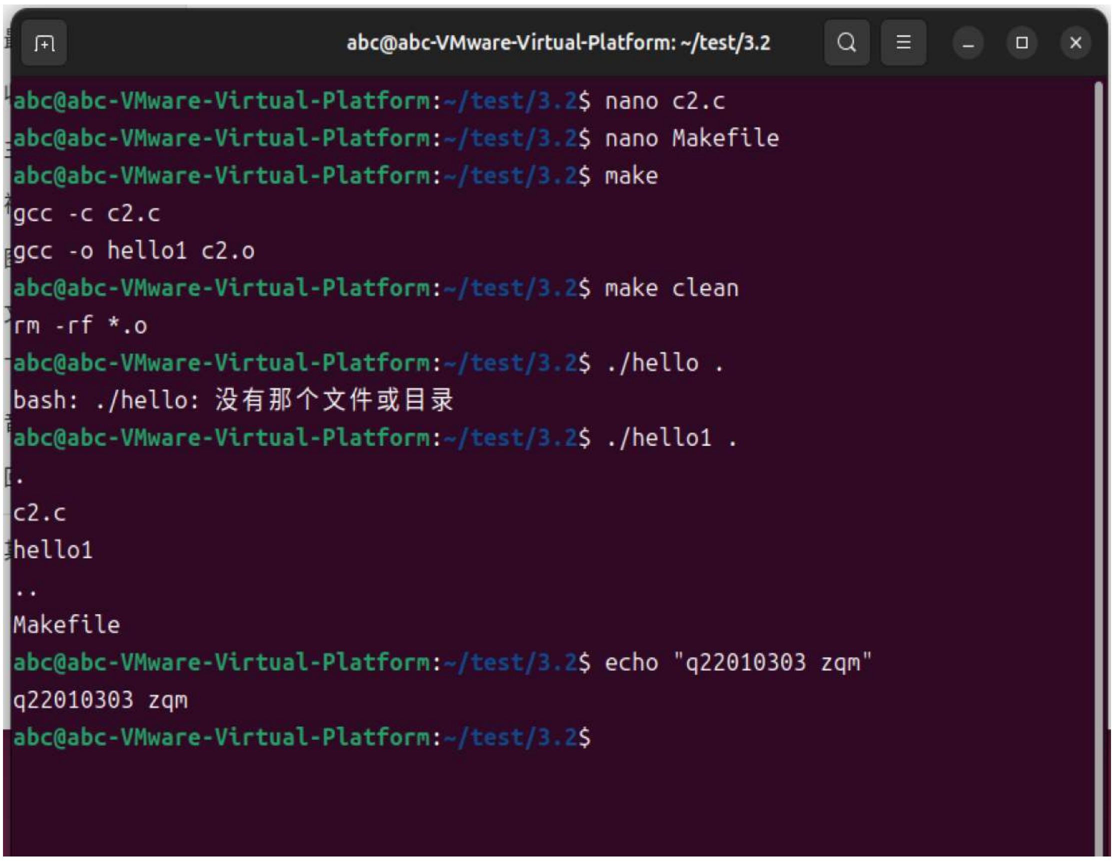
```
abc@abc-VMware-Virtual-Platform:~/test/3.1$ echo "Q22010303zqm"
Q22010303zqm
abc@abc-VMware-Virtual-Platform:~/test/3.1$ make
gcc -c c1.c
gcc -o hello1 c1.o
abc@abc-VMware-Virtual-Platform:~/test/3.1$ make clean
rm -rf *.o
abc@abc-VMware-Virtual-Platform:~/test/3.1$ ls
c1.c  hello1  Makefile  test.txt
abc@abc-VMware-Virtual-Platform:~/test/3.1$ ./hello1 test.txt
hello linux!

abc@abc-VMware-Virtual-Platform:~/test/3.1$
```

Makefile 编写时要用 tab 不用空格

(2) Task 2

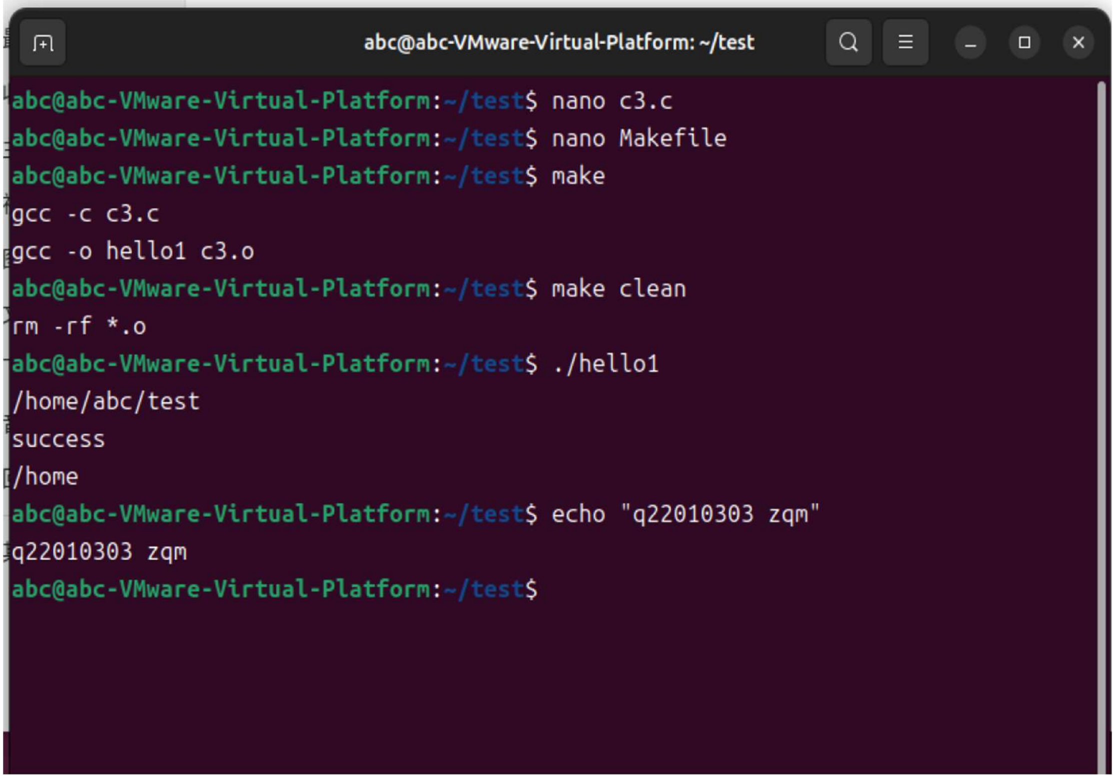
Write a C program that displays all the file names in the current directory. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile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



```
abc@abc-VMware-Virtual-Platform:~/test/3.2$ nano c2.c
abc@abc-VMware-Virtual-Platform:~/test/3.2$ nano Makefile
abc@abc-VMware-Virtual-Platform:~/test/3.2$ make
gcc -c c2.c
gcc -o hello1 c2.o
abc@abc-VMware-Virtual-Platform:~/test/3.2$ make clean
rm -rf *.o
abc@abc-VMware-Virtual-Platform:~/test/3.2$ ./hello .
bash: ./hello: 没有那个文件或目录
abc@abc-VMware-Virtual-Platform:~/test/3.2$ ./hello1 .
.
c2.c
hello1
..
Makefile
abc@abc-VMware-Virtual-Platform:~/test/3.2$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test/3.2$
```

(3) 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.



```
abc@abc-VMware-Virtual-Platform:~/test$ nano c3.c
abc@abc-VMware-Virtual-Platform:~/test$ nano Makefile
abc@abc-VMware-Virtual-Platform:~/test$ make
gcc -c c3.c
gcc -o hello1 c3.o
abc@abc-VMware-Virtual-Platform:~/test$ make clean
rm -rf *.o
abc@abc-VMware-Virtual-Platform:~/test$ ./hello1
/home/abc/test
success
/home
abc@abc-VMware-Virtual-Platform:~/test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test$
```