### (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;
}</pre>
```

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
```

### 图 1 实验 3-1 的任务要求

```
Ltangfethong@ltangfethong-virtual-nachine:-/3-15 nake
gcc -c c1.c
gcc -c helloi c1.o
rm -rf *.o

Ltangfethong@ltangfethong-virtual-nachine:-/3-15 ./helloi c1.c
#include <stdto.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;
}</pre>
```

图 2 实验 3-1 的实现

#### (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);
}
```

Make sure your filename is c2.c

We can use the following makefile.

```
hello2:c2.o
    gcc -o hello1 c2.o
c2.o:c2.c
    gcc -c c2.c
clean:
    rm -rf *.o
```

图 3 实验 3-2 的任务要求

```
liangfeihong@liangfeihong-virtual-machine:~/3-2$ make
gcc -c c2.c
gcc -o hello2 c2.o
rm -rf *.o
liangfeihong@liangfeihong-virtual-machine:~/3-2$ ./hello2 ~/3-2
.
hello2
makefile
c2.c
..
liangfeihong@liangfeihong-virtual-machine:~/3-2$
```

图 4 实验 3-2 的实现

# (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 of 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;
}</pre>
```

Make sure your filename is c3.c

We can use the following makefile.

```
hello3:c3.0
gcc -o hello1 c3.0
c3.o:c3.c
gcc -c c3.c
clean:
rm -rf *.0
```

图 5 实验 3-3 的任务要求

```
liangfeihong@liangfeihong-virtual-machine:~/3-3$ touch makefile
liangfeihong@liangfeihong-virtual-machine:~/3-3$ make
gcc -c c3.c
gcc -o hello1 c3.o
rm -rf *.o
liangfeihong@liangfeihong-virtual-machine:~/3-3$ ./hello1
/home/liangfeihong/3-3
success
/home
```

图 6 实验 3-3 的实现