

# (1) Task 1

题目：

(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;
}
```

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
```

结果：

```
b22040523@localhost: ~/linux3
b22040523@localhost:~/linux3$ vim Make.c1
b22040523@localhost:~/linux3$ make -f Make.c1
gcc -c c1.c
gcc -o hello1 c1.o
b22040523@localhost:~/linux3$ ./hello1 c1.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)
```

```
b22040523@localhost: ~/linux3

{

printf("open source %s failed\n", argv[1]);

return -1;

}

while (fgets(buf,1024, fp))

{

printf("%s\n", buf);

}

return 0;

}

b22040523@localhost:~/linux3$ make -f Make.c1 clean
rm -rf *.o
b22040523@localhost:~/linux3$
```

## (2) Task 2

题目：

(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

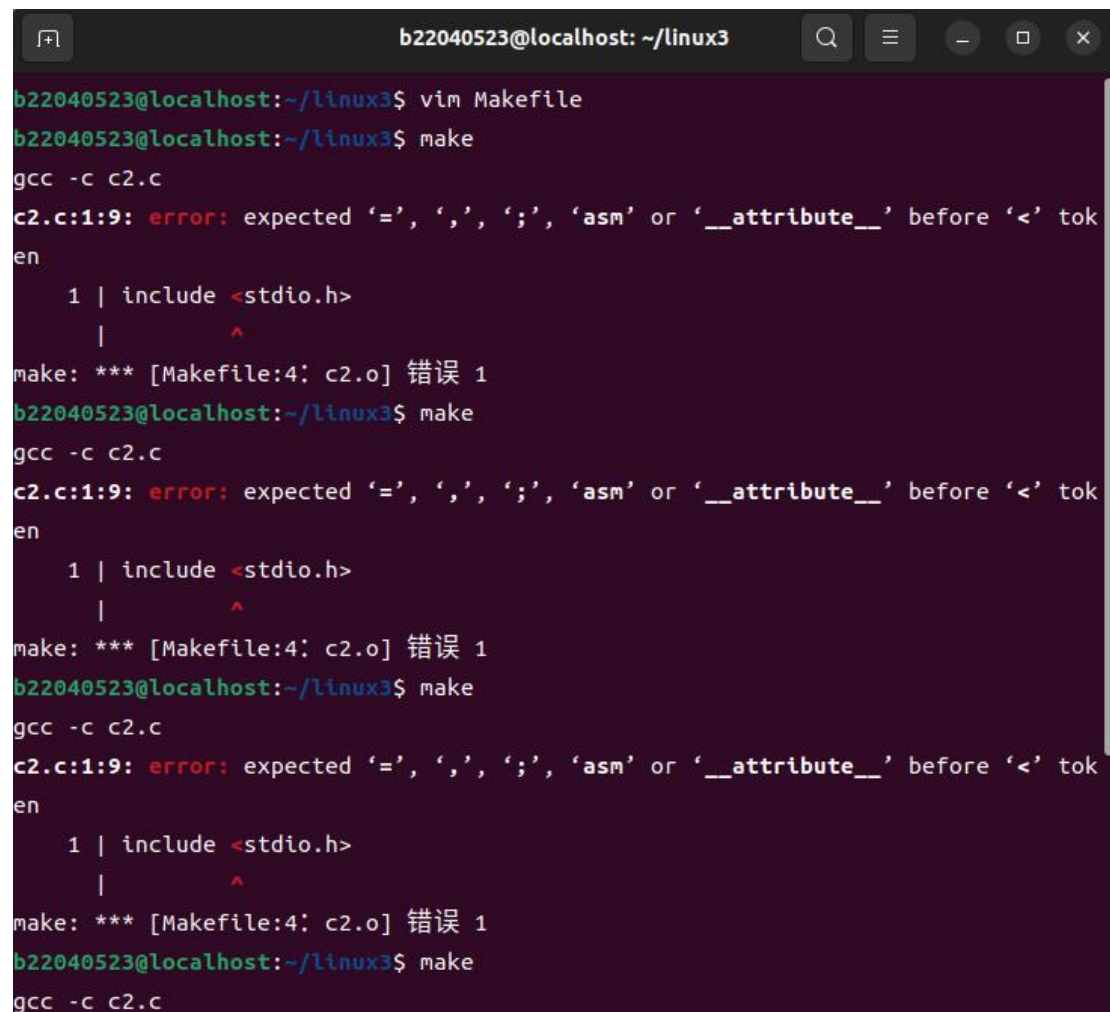
We can use the following makefile.

```

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

```

结果:



```

b22040523@localhost: ~/linux3
b22040523@localhost:~/linux3$ vim Makefile
b22040523@localhost:~/linux3$ make
gcc -c c2.c
c2.c:1:9: error: expected '=', ',', ';', 'asm' or '__attribute__' before '<' token
    1 | include <stdio.h>
      |           ^
make: *** [Makefile:4: c2.o] 错误 1
b22040523@localhost:~/linux3$ make
gcc -c c2.c
c2.c:1:9: error: expected '=', ',', ';', 'asm' or '__attribute__' before '<' token
    1 | include <stdio.h>
      |           ^
make: *** [Makefile:4: c2.o] 错误 1
b22040523@localhost:~/linux3$ make
gcc -c c2.c
c2.c:1:9: error: expected '=', ',', ';', 'asm' or '__attribute__' before '<' token
    1 | include <stdio.h>
      |           ^
make: *** [Makefile:4: c2.o] 错误 1
b22040523@localhost:~/linux3$ make
gcc -c c2.c

```

```
b22040523@localhost: ~/linux3
gcc -c c2.c
c2.c:1:9: error: expected '=', ',', ';', 'asm' or '__attribute__' before '<' token
1 | include <stdio.h>
  |         ^
make: *** [Makefile:4: c2.o] 错误 1
b22040523@localhost:~/linux3$ make
gcc -c c2.c
c2.c:1:9: error: expected '=', ',', ';', 'asm' or '__attribute__' before '<' token
1 | include <stdio.h>
  |         ^
make: *** [Makefile:4: c2.o] 错误 1
b22040523@localhost:~/linux3$ make
gcc -c c2.c
gcc -o hello1 c2.o
b22040523@localhost:~/linux3$ ./hello c2.c
bash: ./hello: 没有那个文件或目录
b22040523@localhost:~/linux3$ ./hello1 c2.c
error
段错误 (核心已转储)
b22040523@localhost:~/linux3$ make clean
rm -rf *.o
b22040523@localhost:~/linux3$
```

### (3) Task 3

题目：

(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");
    }
}
```

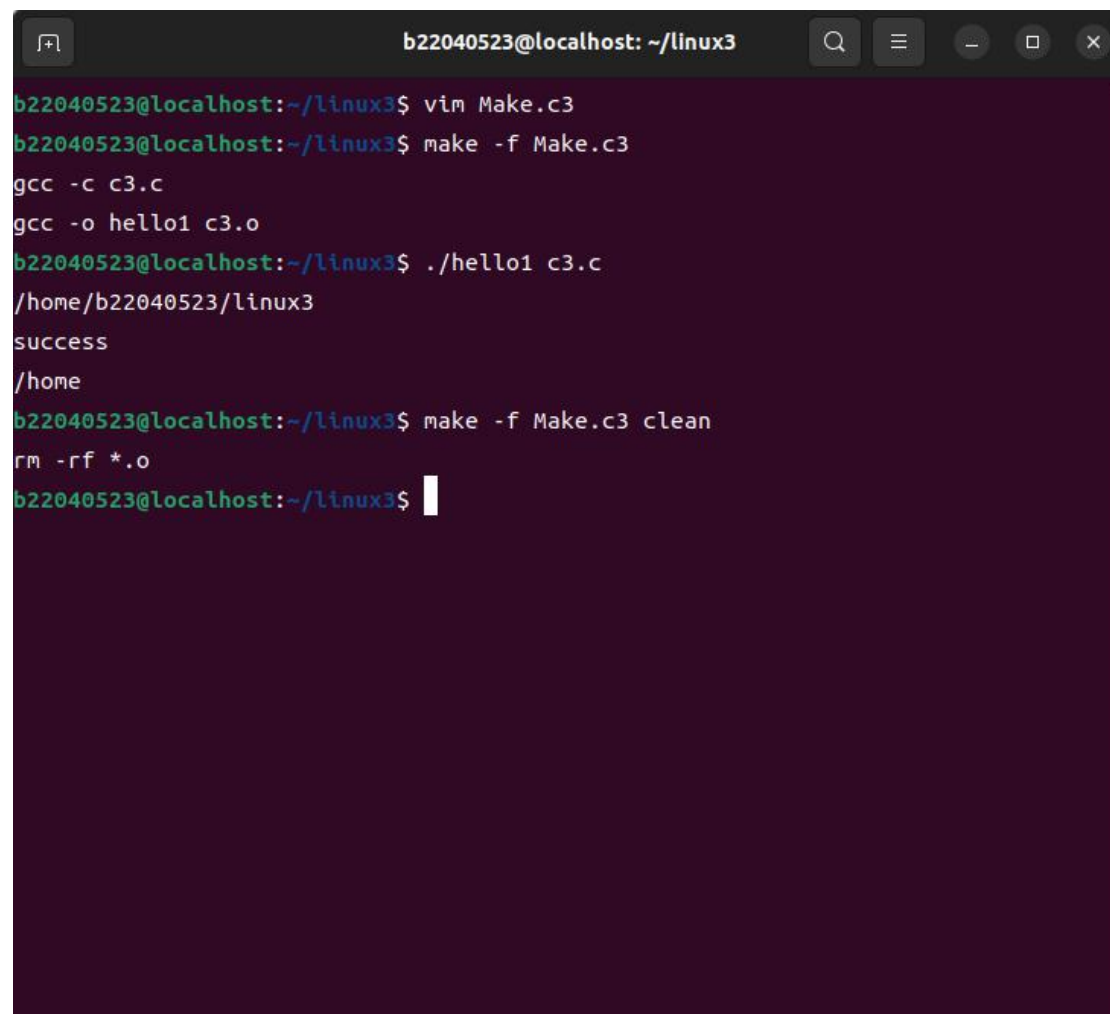
```
else
{
printf("success\n");
}
getcwd(buf2,1024);
printf("%s\n",buf2);
return 0;
}
```

Make sure your filename is c3.c

We can use the following makefile.

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

结果:

A terminal window titled 'b22040523@localhost: ~/linux3' with standard window controls. The terminal shows the following commands and output:

```
b22040523@localhost:~/linux3$ vim Make.c3
b22040523@localhost:~/linux3$ make -f Make.c3
gcc -c c3.c
gcc -o hello1 c3.o
b22040523@localhost:~/linux3$ ./hello1 c3.c
/home/b22040523/linux3
success
/home
b22040523@localhost:~/linux3$ make -f Make.c3 clean
rm -rf *.o
b22040523@localhost:~/linux3$
```