

```
cyy@cyy-virtual-machine:~$ cd linux3
cyy@cyy-virtual-machine:~/linux3$ gedit c1.c
```

```
1 #include <stdio.h>
2 int 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 }
```

创建 c1.c 文件

```
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
```

创建 makefile 文件

```
cyy@cyy-virtual-machine:~/linux3$ make
gcc -c c1.c
gcc -o hello1 c1.o
cyy@cyy-virtual-machine:~/linux3$ echo "hello world" >test.txt
cyy@cyy-virtual-machine:~/linux3$ ./hello1 test.txt
hello world

cyy@cyy-virtual-machine:~/linux3$ make clean
rm -rf *.o
```

编译运行

```
cyy@cyy-virtual-machine:~/linux3$ gedit c2.c
打开(O) 保存(S)
c2.c
~/linux3
makefile  c2.c
1 #include <stdio.h>
2 #include <dirent.h>
3 int main(int argc, char *argv[])
4 {
5     const char *path = (argc > 1) ? argv[1] : ".";
6     DIR *dirp = opendir(path);
7     if (!dirp) {
8         perror("opendir");
9         return 1;
10    }
11    struct dirent *d;
12    while ((d = readdir(dirp)))
13        printf("%s\n", d->d_name);
14    closedir(dirp);
15    return 0;
16 }
```

创建 c2.c 文件

```
makefile
1 hello1:c2.o
2      gcc -o hello1 c2.o
3 c1.o:c2.c
4      gcc -c c2.c
5 clean:
6      rm -rf *.o
```

修改 makefile 文件

```
cyy@cyy-virtual-machine:~/linux3$ make hello1
cc -c -o c2.o c2.c
gcc -o hello1 c2.o
cyy@cyy-virtual-machine:~/linux3$ ./hello1
c2.c
c2.o
.
c1.c
hello1
...
makefile
test.txt
cyy@cyy-virtual-machine:~/linux3$ ./hello1 /home
B22110523
.
...
cyy
cyy@cyy-virtual-machine:~/linux3$ make clean
rm -rf *.o
```

编译运行

```
cyy@cyy-virtual-machine:~/linux3$ gedit c3.c
打开(O) 保存(S)
c3.c
makefile c2.c c3.c
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <limits.h> // PATH_MAX 更规范
4 int main()
5 {
6     char buf[PATH_MAX];
7     if (getcwd(buf, sizeof(buf)) == NULL) {
8         perror("getcwd");
9         return 1;
10    }
11    printf("当前目录: %s\n", buf);
12
13    if (chdir("/home") < 0) {
14        perror("chdir");
15        return 1;
16    }
17    printf("切换成功! \n");
18
19    if (getcwd(buf, sizeof(buf)) == NULL) {
20        perror("getcwd");
21        return 1;
22    }
23    printf("新目录: %s\n", buf);
24    return 0;
25 }
```

创建 c3.c 文件

```
打开(O) 保存(S)
makefile c2.c c3.c
makefile
1 hello1:c3.o
2     gcc -o hello1 c3.o
3 c1.o:c3.c
4     gcc -c c3.c
5 clean:
6         rm -rf *.o
```

修改 makefile 文件

```
cyy@cyy-virtual-machine:~/linux3$ make hello1
cc -c -o c3.o c3.c
gcc -o hello1 c3.o
cyy@cyy-virtual-machine:~/linux3$ ./hello1
当前目录: /home/cyy/linux3
切换成功!
新目录: /home
cyy@cyy-virtual-machine:~/linux3$ make clean
rm -rf *.o
```

编译运行