

Linux 实验 3

(每个程序内部注释学号、运行前在界面打印学号证明)

(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
```

c1.o 内容

```
1 //B22040702张欣雨
2 #include <stdio.h>
3
4 int main(int argc, char* argv[])
5 {
6     char buf[1024] = { 0 };
7     FILE* fp = fopen(argv[1], "r");
8     if (argc < 2)
9     {
10        printf("please input source file!\n");
11    }
12    if (fp == NULL)
13    {
14        printf("open source %s failed\n", argv[1]);
15        return -1;
16    }
17    while (fgets(buf, 1024, fp))
18    {
19        printf("%s\n", buf);
20    }
21    return 0;
22 }
```

makefile 内容

```
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
```

运行结果

```
zoey@zoey:~/experiment/experiment3$ echo b22040702
b22040702
zoey@zoey:~/experiment/experiment3$ make c1.o
gcc -c c1.c
zoey@zoey:~/experiment/experiment3$ make
gcc -o hello1 c1.o
zoey@zoey:~/experiment/experiment3$ ./hello1
please input source file!
open source (null) failed
zoey@zoey:~/experiment/experiment3$ ./hello1 b22040702.txt
This is a txtfile for test.
```

(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
```

c2.o 内容

```
1 //B22040702张欣雨
2
3 #include<stdio.h>
4 #include<dirent.h>
5 #include<sys/types.h>
6 int main(int argc, char* argv[])
7 {
8     DIR* dirp;
9     struct dirent* direntp;
10    if((dirp = opendir(argv[1])) == NULL){
11        printf("error\n");
12        //exit(1);
13    }
14    while ((direntp = readdir(dirp)) != NULL)
15        printf("%s\n", direntp->d_name);
16    closedir(dirp);
17    // exit(0);
18 }
```

makefile 内容

```
1 hello2: c2.o
2         gcc -o hello2 c2
3 c2.o: c2.c
4         gcc -c c2.c
5 clean:
6         rm -rf *.o
```

运行结果

```
zoey@zoey:~/experiment/experiment3-2$ echo b22040702
b22040702
zoey@zoey:~/experiment/experiment3-2$ make c2.o
gcc -c c2.c
zoey@zoey:~/experiment/experiment3-2$ ./hello2 /home/zoey/experiment/experiment3-2
c2.c
.
Makefile
..
c2.o
hello2
zoey@zoey:~/experiment/experiment3-2$ make clean
rm -rf *.o
zoey@zoey:~/experiment/experiment3-2$ ./hello2 /home/zoey/experiment/experiment3-2
c2.c
.
Makefile
..
hello2
```

(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
```

c3.o 内容

```
1 //B22040702张欣雨
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 int main()
6 {
7     char buf[1024] = {0};
8     char buf2[1024]={0};
9     getcwd(buf, 1024);
10    printf("%s\n", buf);
11    if(chdir("/home")<0){
12        printf("error\n");
13    }
14    else
15    {
16        printf("success\n");
17    }
18    getcwd(buf2,1024);
19    printf("%s\n",buf2);
20    return 0;
```

makefile 内容

```
1 hello3:c3.o
2      gcc -o hello1 c3.o
3 c3.o:c3.c
4      gcc -c c3.c
5 clean:
6      rm -rf *.o
```

运行结果

```
zoey@zoey:~/experiment/experiment3-3$ echo b22040702
b22040702
zoey@zoey:~/experiment/experiment3-3$ vim c3.c
zoey@zoey:~/experiment/experiment3-3$ vim Makefile
zoey@zoey:~/experiment/experiment3-3$ make
gcc -c c3.c
gcc -o hello1 c3.o
zoey@zoey:~/experiment/experiment3-3$ ./hello1
/home/zoey/experiment/experiment3-3
success
/home
zoey@zoey:~/experiment/experiment3-3$
```