Experiment3 C programming experiment

Experimental purpose:

Further use the basic syntax of C programming language in Linux system, deepen the understanding of the knowledge.

(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;
}
```



```
yearless@yearless-virtual-machine:~/Linux2$ touch c1.c
yearless@yearless-virtual-machine:~/Linux2$ gcc -o c1 c1.c
yearless@yearless-virtual-machine:~/Linux2$ ./c1
please input source file!
yearless@yearless-virtual-machine:~/Linux2$ ./c1 try.txt
helloworld!
```

(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);
}
```

```
yearless@yearless-virtual-machine:~/Linux2$ touch c2.c
yearless@yearless-virtual-machine:~/Linux2$ gcc -o c2 c2.c
yearless@yearless-virtual-machine:~/Linux2$ ./c2
Please provide a directory path.
yearless@yearless-virtual-machine:~/Linux2$ ./c2 '/home/yearless/Linux2'
sy3_2
.
ex2_4.sh
syex2_31
sy3_5.cpp
sy3_1.c
..
c2
syex2_32.c
syex2_1
c1
sy3_4.cpp
ex2_5.sh
sy3_4
ex2_3.sh
syex2_2
c1.c
sy3_1
c2.c
```

(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.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;
}
```

```
yearless@yearless-virtual-machine:~/Linux2$ gcc -o c3 c3.c
yearless@yearless-virtual-machine:~/Linux2$ ./c3
/home/yearless/Linux2
success
/home
```