Task1

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.

```c
#include <stdio.h>

int main(int argc, char* argv[]) {
    char buf[1024] = { 0 };
    if (argc < 2) {
        printf("Please input source file!\n");
        return -1;
    }

    FILE* fp = fopen(argv[1], "r");
    if (fp == NULL) {
        printf("Open source %s failed\n", argv[1]);
        return -1;
    }

    while (fgets(buf, 1024, fp)) {
        printf("%s", buf);
    }

    fclose(fp);
    return 0;
}
```

```
hello1: c1.o
	gcc -o hello1 c1.o

c1.o: c1.c
	gcc -c c1.c

clean:
	rm -rf *.o
```

```
pengyupeng@pengyupeng-virtual-machine:~$ make
gcc -c c1.c
gcc -o hello1 c1.o
pengyupeng@pengyupeng-virtual-machine:~$ ./hello1 B22040722.txt
hello world
pengyupeng@pengyupeng-virtual-machine:~$ make clean
rm -rf *.o
```

Task2

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.

```c
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>

int main(int argc, char* argv[]) {
    DIR* dirp;
    struct dirent* direntp;

    if (argc < 2) {
        printf("请提供目录路径! \n");
        return -1;
    }

    if ((dirp = opendir(argv[1])) == NULL) {
        printf("打开目录失败\n");
        return -1;
    }

    while ((direntp = readdir(dirp)) != NULL) {
        printf("%s\n", direntp->d_name);
    }

    closedir(dirp);
    return 0;
}
```

```
hello2: c2.o
	gcc -o hello2 c2.o

c2.o: c2.c
	gcc -c c2.c

clean:
	rm -rf *.o
```

```
pengyupeng@pengyupeng-virtual-machine:~$ make
gcc -c c2.c
gcc -o hello2 c2.o
pengyupeng@pengyupeng-virtual-machine:~$ ./hello2 .
.cache
c1.c
pengyupeng2.txt
.gnupg
.demo.c.swp
.sudo_as_admin_successful
.ssh
.bash_logout
模板
.demo.c.swo
.dbus
文档
.presage
图片
c2.o
视频
.profile
demo
B22040722
```

```
pengyupeng@pengyupeng-virtual-machine:~$ make clean
rm -rf *.o
```

Task3

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.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    char buf[1024] = {0};
    char buf2[1024] = {0};

    // 获取当前工作目录
    getcwd(buf, 1024);
    printf("当前目录: %s\n", buf);

    // 尝试更改工作目录
    if (chdir("/home") < 0) {
        printf("更改目录失败\n");
    } else {
        printf("更改目录成功\n");
    }

    // 获取更改后的工作目录
    getcwd(buf2, 1024);
    printf("新目录: %s\n", buf2);

    return 0;
}
```

```
hello3: c3.o
	gcc -o hello3 c3.o

c3.o: c3.c
	gcc -c c3.c

clean:
	rm -rf *.o
```

```
pengyupeng@pengyupeng-virtual-machine:~$ make
gcc -c c3.c
gcc -o hello3 c3.o
pengyupeng@pengyupeng-virtual-machine:~$ ./hello3
当前目录: /home/pengyupeng
更改目录成功
新目录: /home
pengyupeng@pengyupeng-virtual-machine:~$ make clean
rm -rf *.o
```