南京邮电大学

实 验 报 告

（ 2024 / 2025 学年 第 一 学期）

课程名称	Linux编程
实验名称	实验三
实验时间	2024 年 12 月 13 日
指导单位	计算机学院、软件学院、网络空间安全学院
指导教师	王磊

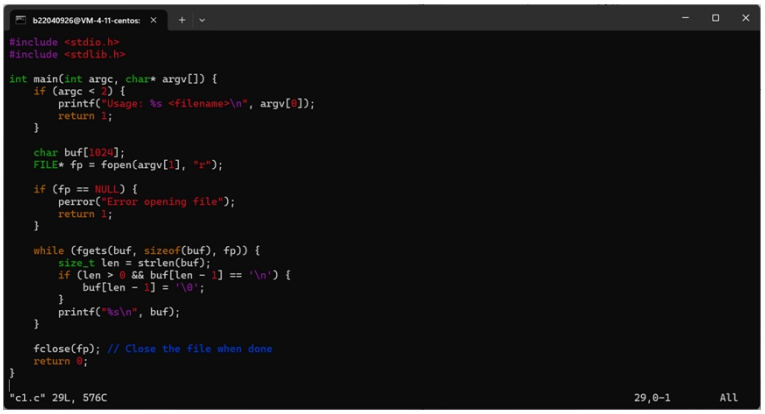
学生姓名	毛赠博	班级学号	B22041007
学院(系)	计算机学院	专　　业	信息安全

实 验 报 告

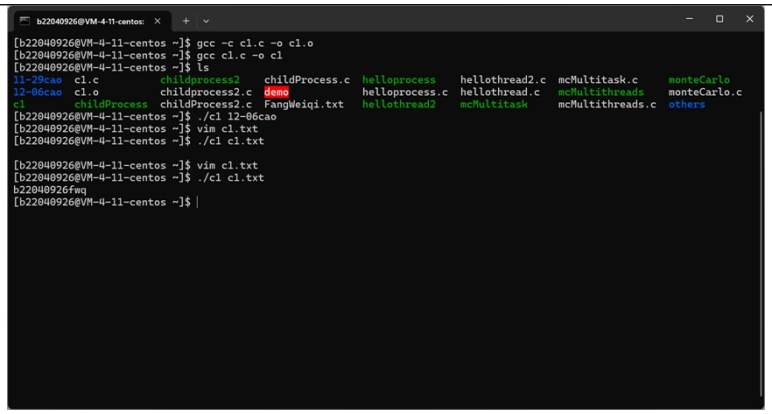
实验目的：进一步使用 Linux 系统中 C 编程语言的基本语法，加深对知识的理解。

任务 1 （1） 编写一个使用标准 I/O 库来显示文本文件内容的 C 程序。程序由 make 工具编译和链接，需要先生成 .o 文件，再生成可执行文件，并有删除 makefile 文件中中间文件（.o）的功能。

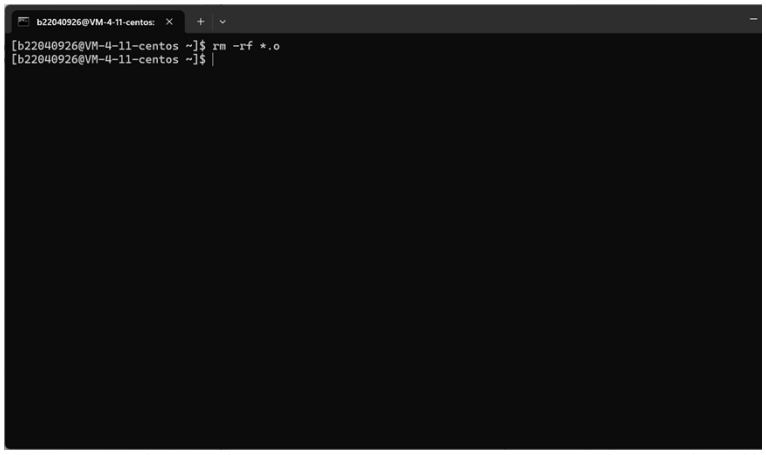
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char* argv[]) {
    if (argc < 2) {
        printf("Usage: %s <filename>\n", argv[0]);
        return 1;
    }

    char buf[1024];
    FILE* fp = fopen(argv[1], "r");

    if (fp == NULL) {
        perror("Error opening file");
        return 1;
    }

    while (fgets(buf, sizeof(buf), fp)) {
        size_t len = strlen(buf);
        if (len > 0 && buf[len - 1] == '\n') {
            buf[len - 1] = '\0';
        }
        printf("%s\n", buf);
    }

    fclose(fp); // Close the file when done
    return 0;
}
```



运行代码，成功读取到指定文件：



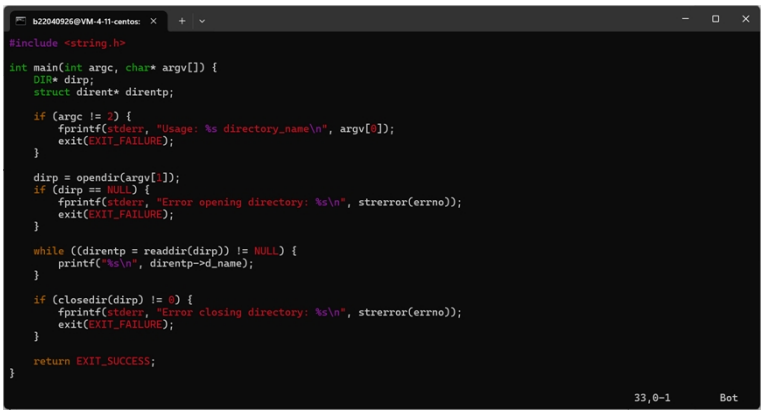
我们可以使用以下 makefile



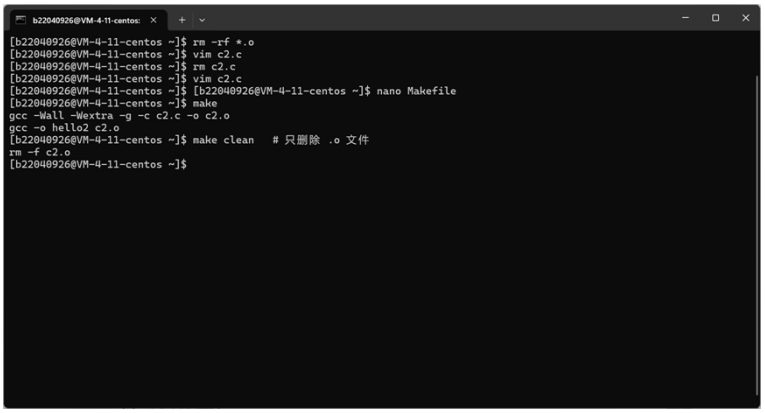
（2） 任务 2 （2） 编写一个 C 程序，显示当前目录中的所有文件名。程序由 make 工具编译和链接，需要先生成 .o 文件，再生成可执行文件，并有删除 makefile 文件中中间文件（.o）的功能。

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <errno.h>
#include <string.h>

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

   if (argc != 2) {
      fprintf(stderr, "Usage: %s directory_name\n", argv[0]);
      exit(EXIT_FAILURE);
   }

   dirp = opendir(argv[1]);
   if (dirp == NULL) {
      fprintf(stderr, "Error opening directory: %s\n", strerror(errno));
      exit(EXIT_FAILURE);
   }

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

   if (closedir(dirp) != 0) {
```

```
        fprintf(stderr, "Error closing directory: %s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }

    return EXIT_SUCCESS;
}
```



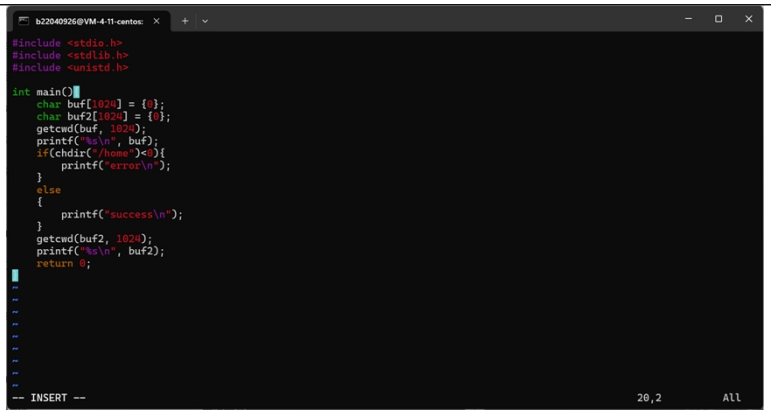
运行结果：

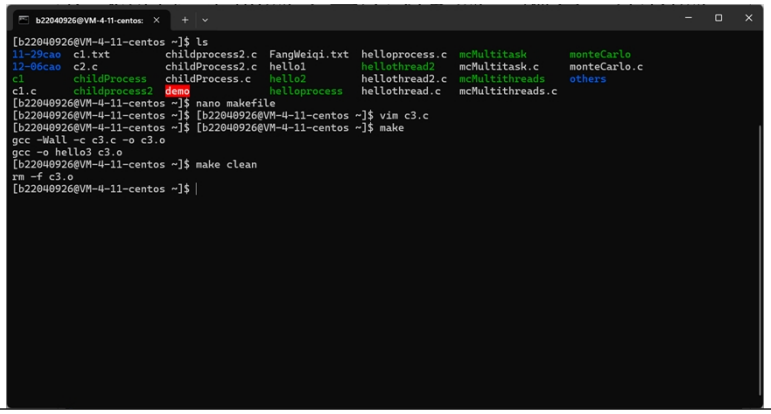


（3） 编写一个 C 程序，更改当前进程的工作目录。程序由 make 工具编译和链接，需要先生成 .o 文件，再生成可执行文件，并有删除 makefile 文件中中间文件（.o）的功能。

```
# 定义可执行文件名
TARGET = hello3
SRC = c3.c
OBJ = $(SRC:.c=.o)
CC = gcc
CFLAGS = -Wall
$(TARGET): $(OBJ)
	$(CC) -o $@ $^
$(OBJ): $(SRC)
	$(CC) $(CFLAGS) -c $< -o $@
clean:
	rm -f $(OBJ)

.PHONY: clean
```

运行代码：


b22040926@VM-4-11-centos:
#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;
}
-- INSERT --
20,2
All


b22040926@VM-4-11-centos:
[b22040926@VM-4-11-centos ~]$ ls
11-29cao c1.txt childprocess2.c FangWeiqi.txt helloprocess.c mcMultitask monteCarlo
12-06cao c2.c childProcess2.c hello1 hellothread2 mcMultitask.c monteCarlo.c
c1 childProcess childProcess.c hello2 hellothread2.c mcMultithreads others
c1.c childprocess2 demo helloprocess hellothread.c mcMultithreads.c
[b22040926@VM-4-11-centos ~]$ nano makefile
[b22040926@VM-4-11-centos ~]$ [b22040926@VM-4-11-centos ~]$ vim c3.c
[b22040926@VM-4-11-centos ~]$ [b22040926@VM-4-11-centos ~]$ make
gcc -Wall -c c3.c -o c3.o
gcc -o hello3 c3.o
[b22040926@VM-4-11-centos ~]$ make clean
rm -f c3.o
[b22040926@VM-4-11-centos ~]$