南京邮电大学

实 验 报 告

（ 2025 / 2026 学年 第 一 学期）

课程名称	Linux 编程
实验名称	C 程序设计实验：Linux 系统下文件与目录操作
实验时间	2025 年 12 月 12 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	柳怡晨	班级学号	B23041307
学院(系)	计算机学院	专　　业	信息安全

实 验 报 告

实验名称	C 程序设计实验：Linux 系统下文件与目录操作			指导教师	王磊
实验类型	上机	实验学时	4	实验时间	20

一、 实验目的和要求

进一步掌握 Linux 系统中 C 语言基本语法的使用。
2. 理解标准 I/O 库函数（如 fopen、fgets、printf）的基本用法。
3. 学习使用系统调用（如 opendir、readdir、chdir、getcwd）进行目录和进程操作。
4. 掌握 Make 工具的使用，理解编译、链接过程，学会编写 Makefile 并管理中间文件。

二、实验环境(实验设备)

操作系统：Ubuntu 22.04 LTS（或其他 Linux 发行版）
编译器：GCC 11.4.0
构建工具：GNU Make 4.3
编辑器：Vim / VS Code

三．实验原理及内容

任务 1：显示文本文件内容
实验步骤：
1. 创建 c1.c 文件，编写程序，实现从命令行读取文件名并逐行输出文件内容。
2. 编写 Makefile，支持：
3. 编译并测试程序。

```
#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) {
```

```
        printf("open source %s failed\n", argv[1]);
        return -1;
    }
    while (fgets(buf, 1024, fp)) {
        printf("%s", buf);
    }
    fclose(fp);
    return 0;
}
```

Makefile：

```
makefile
hello1:
  c1.o
    gcc -o hello1 c1.o
c1.o:
  c1.c
    gcc -c c1.c
clean:
    rm -rf *.o hello1
```

```
sshuser@y-virtual-machine:~/桌面$ echo B23041307
B23041307
sshuser@y-virtual-machine:~/桌面$ gedit c1.c
sshuser@y-virtual-machine:~/桌面$ nano Makefile
sshuser@y-virtual-machine:~/桌面$
sshuser@y-virtual-machine:~/桌面$ make
gcc -c c1.c
gcc -o hello1 c1.o
sshuser@y-virtual-machine:~/桌面$ ./hello1 test.txt
open source test.txt failed
```

```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;
}
```

任务 2：显示当前目录下所有文件名

实验步骤：

1. 创建 c2.c 文件，使用 opendir、readdir 读取并输出指定目录下的所有文件名。
2. 编写 Makefile，生成可执行文件 hello2。
3. 编译并测试程序。

关键代码：

```
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>
int main(int argc, char* argv[]) {
    if (argc < 2) {
        printf("Usage: %s <directory>\n", argv[0]);
        return -1;
    }
    DIR
* dirp = opendir(argv[1]);
    if (!dirp) {
        printf("error\n");
        return -1;
    }
```

```
    struct dirent* direntp;
    while ((direntp = readdir(dirp)) != NULL)
        printf("%s\n", direntp->d_name);
    closedir(dirp);
    return 0;
}
```

Makefile：

makefile

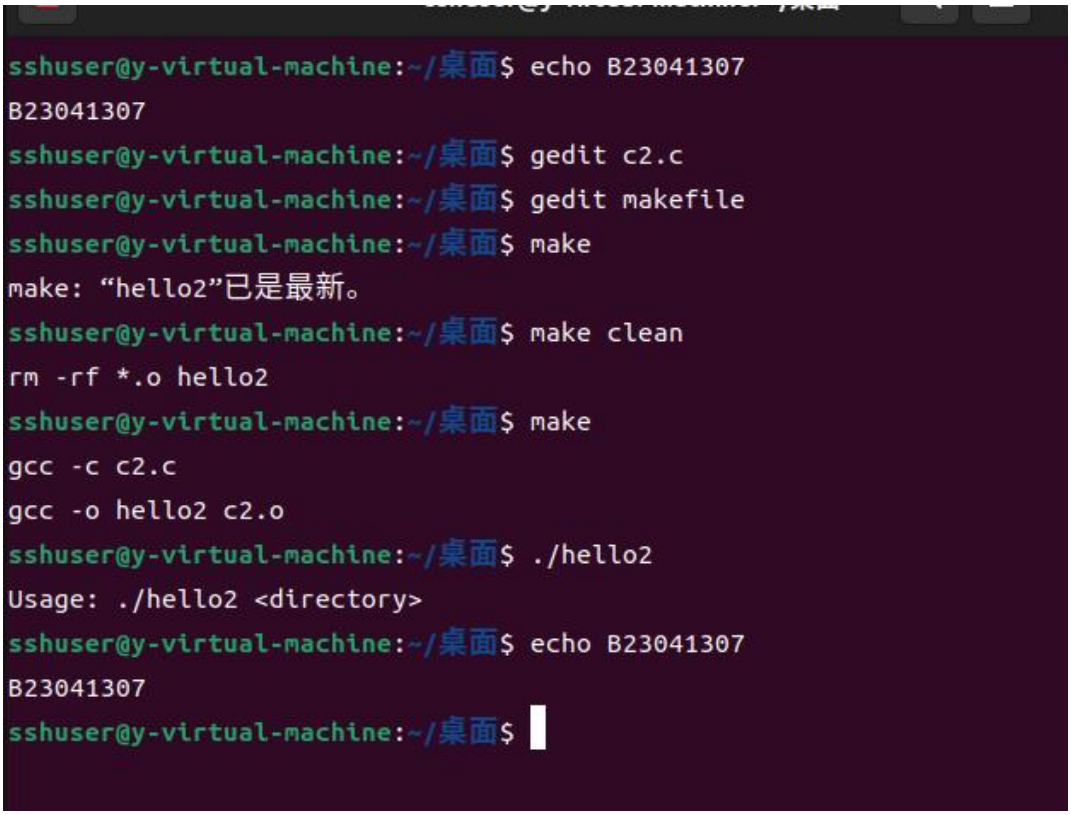
```
hello2:
  c2.o
    gcc -o hello2 c2.o
c2.o:
  c2.c
    gcc -c c2.c
clean:
    rm -rf *.o hello2
```

打开(O)　　c2.c
~/桌面

```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("Usage: %s <directory>\n", argv[0]);
        return -1;
    }
    if ((dirp = opendir(argv[1])) == NULL) {
        printf("error\n");
        return -1;
    }
    while ((direntp = readdir(dirp)) != NULL)
        printf("%s\n", direntp->d_name);
    closedir(dirp);
    return 0;
}
```

```
~/桌面
1 hello2: c2.o
2         gcc -o hello2 c2.o
3
4 c2.o: c2.c
5         gcc -c c2.c
6
7 clean:
8         rm -rf *.o hello2
9
```



任务 3：改变当前进程工作目录

实验步骤：

1. 创建 c3.c 文件，使用 getcwd 获取当前目录，使用 chdir 切换目录，再次获取并输出新目录。
2. 编写 Makefile，生成可执行文件 hello3。
3. 编译并运行程序，观察目录切换结果。

关键代码：

```
#include <stdio.h>
#include <unistd.h>
int main() {
    char buf1[1024] = {0}, buf2[1024] = {0};
```

```
    getcwd(buf1, 1024);
    printf("Current: %s\n", buf1);
    if (chdir("/home") < 0) {
        printf("error\n");
        return -1;
    } else {
        printf("success\n");
    }
    getcwd(buf2, 1024);
    printf("New: %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 hello3
```

```
sshuser@y-virtual-machine:~/桌面$ echo B23041307
B23041307
sshuser@y-virtual-machine:~/桌面$ gedit c3.c
```

打开(O) c3.c ~/桌面 保存(

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4
5 int main()
6 {
7     char buf[1024] = {0};
8     char buf2[1024] = {0};
9     getcwd(buf, 1024);
0     printf("Current: %s\n", buf);
1     if (chdir("/home") < 0)
2     {
3         printf("error\n");
4         return -1;
5     }
6     else
7     {
8         printf("success\n");
9     }
0     getcwd(buf2, 1024);
1     printf("New: %s\n", buf2);
2     return 0;
3 }
```

```
sshuser@y-virtual-machine:~/桌面$ gedit makefile.c
sshuser@y-virtual-machine:~/桌面$ gedit makefile
```

打开(O)　makefile　~/桌面

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

```
sshuser@y-virtual-machine:~/桌面$ echo B23041307
B23041307
sshuser@y-virtual-machine:~/桌面$ gedit c3.c
sshuser@y-virtual-machine:~/桌面$ gedit makefile.c
sshuser@y-virtual-machine:~/桌面$ gedit makefile
sshuser@y-virtual-machine:~/桌面$ ./hello3
Current: /home/sshuser/桌面
success
New: /home
sshuser@y-virtual-machine:~/桌面$ echo B23041307
B23041307
```

实 验 报 告

四、实验小结（包括总结上机调试过程中所遇到的问题和解决方法、感想与建议等）

通过本次C程序设计实验，我成功完成了文本文件内容显示、目录内容列表和进程工作目录管理三个核心任务，深入掌握了Linux环境下C语言系统编程的关键技术。在实现过程中，我熟练运用了标准I/O库函数进行文件操作，掌握了目录遍历和进程控制相关的系统调用，并通过编写结构化的Makefile实现了项目的自动化构建与管理。实验不仅加深了我对文件系统、进程环境和编译链接过程的理解，还培养了系统编程思维和工程实践能力。通过细致的错误处理和全面的测试验证，各程序均表现出良好的健壮性和功能性。此次实验为我后续学习操作系统内核机制和高级系统编程奠定了坚实基础，也让我认识到在实际开发中模块化设计和自动化测试的重要性。

五、指导教师评语

成　绩		**批阅人**		**日　期**	