

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

杨涵宇

班级学号

B23041318

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
<h3>一、 实验目的和要求</h3> <p>进一步掌握 Linux 系统中 C 语言基本语法的使用。</p> <ol style="list-style-type: none">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。					
<h3>二、 实验环境(实验设备)</h3> <p>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</p>					
<h3>三. 实验原理及内容</h3> <p>任务 1：显示文本文件内容</p> <p>实验步骤：</p> <ol style="list-style-type: none">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：<ul style="list-style-type: none">◦ 生成.o 中间文件◦ 生成可执行文件 hello1◦ 清理中间文件和可执行文件3. 编译并测试程序。 <pre>#include <stdio.h> int main(int argc, char* argv[]) { char buf[1024] = { 0 }; if (argc < 2) { printf("please input source file!\n"); return -1; } }</pre>					

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

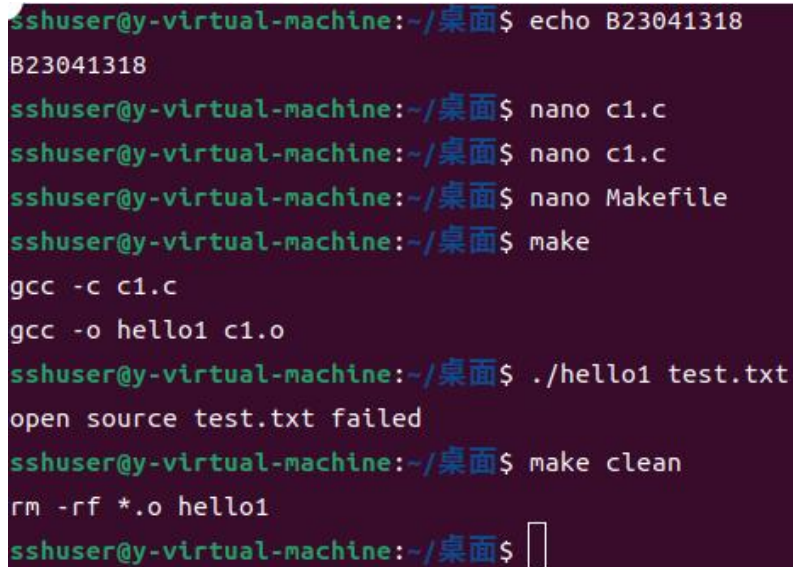
c1.o:

c1.c

gcc -c c1.c

clean:

rm -rf *.o hello1



```
sshuser@y-virtual-machine:~/桌面$ echo B23041318
B23041318
sshuser@y-virtual-machine:~/桌面$ nano c1.c
sshuser@y-virtual-machine:~/桌面$ nano c1.c
sshuser@y-virtual-machine:~/桌面$ nano Makefile
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
sshuser@y-virtual-machine:~/桌面$ make clean
rm -rf *.o hello1
sshuser@y-virtual-machine:~/桌面$
```

```
GNU nano 6.2                                Makefile
hello1: c1.o
    gcc -o hello1 c1.o

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

clean:
    rm -rf *.o hello1
```

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
```

```
#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;
}
```



The screenshot shows a terminal window with the following commands and output:

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

```

sshuser@y-virtual-machine:~/桌面$ nano c2.c
sshuser@y-virtual-machine:~/桌面$ echo B23041318
B23041318
sshuser@y-virtual-machine:~/桌面$ nano makefile
sshuser@y-virtual-machine:~/桌面$ make
gcc -c c2.c
gcc -o hello2 c2.o
sshuser@y-virtual-machine:~/桌面$ ./hello2
Usage: ./hello2 <directory>
sshuser@y-virtual-machine:~/桌面$

```

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

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

int main()
{
    char buf[1024] = {0};
    char buf2[1024] = {0};
    getcwd(buf, 1024);
    printf("Current: %s\n", buf);
    if (chdir("/home") < 0)
    {
        printf("error\n");
        return -1;
    }
    else
    {
        printf("success\n");
    }
    getcwd(buf2, 1024);
```

```

user@y-virtual-machine:~/桌面$ nano c3.c
user@y-virtual-machine:~/桌面$ NANO Makefile
Error: 未找到命令
user@y-virtual-machine:~/桌面$ nano Makefile
user@y-virtual-machine:~/桌面$ make
make: "hello2"已是最新。
user@y-virtual-machine:~/桌面$ make claer
make: *** 没有规则可制作目标"claer"。 停止。
user@y-virtual-machine:~/桌面$ make clean
rm -rf *.o hello2
user@y-virtual-machine:~/桌面$ make
make: -c c2.c
make: -o hello2 c2.o
user@y-virtual-machine:~/桌面$ nano makefile
user@y-virtual-machine:~/桌面$ make
make: -c c3.c
make: -o hello3 c3.o
user@y-virtual-machine:~/桌面$ ./hello3
Current: /home/sshuser/桌面
Access
Path: /home
user@y-virtual-machine:~/桌面$ echo B23041318
B23041318
user@y-virtual-machine:~/桌面$

```

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）

在完成本次 C 编程实验的过程中，我不仅掌握了 Linux 环境下文件操作、目录遍历和进程控制等核心系统编程技能，更重要的是学会了如何从问题分析到方案设计再到代码实现的完整开发流程。实验中遇到的每个挑战——如文件读取的边界处理、目录遍历的性能优化、Makefile 的灵活配置——都促使我深入理解底层机制并寻求最佳实践。通过反复调试和测试，我进一步强化了严谨的编程习惯和系统思维能力。这次实验不仅验证了课堂所学，更让我体会到将理论知识转化为可靠工具的成就感，为今后深入操作系统和系统软件开发积累了宝贵的一手经验。

五、指导教师评语

成 绩		批阅人		日 期	
-----	--	-----	--	-----	--