

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	Experiment3: Linux 下 C 编程			
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	李源	班级学号	B22041124
学院(系)	计算机学院	专 业	信息安全

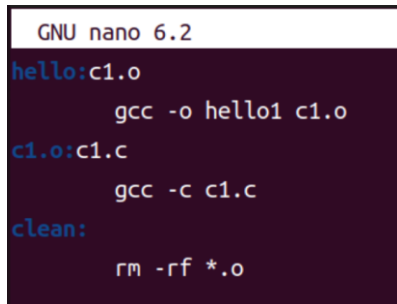
实 验 报 告

实验名称	Linux 基本命令操作			指导教师	王磊
实验类型	验证	实验学时	2	实验时间	2024. 1. 13
<p>一、 实验目的和要求:</p> <ol style="list-style-type: none">1. 掌握 Linux 下的 C 编程2. 掌握项目管理工具 GNU Make3. 进一步理解文件路径的概念					
<p>二、 实验环境(实验设备)</p> <p>硬件: 微型计算机</p> <p>软件: Ubuntu(或其他 Linux 版本)</p>					
<p>三、 实验原理及内容</p> <p>1.编写一个 C 程序, 使用标准输入/输出库来显示文本文件的内容。该程序通过 make 工具进行编译和链接, 首先需要生成 .o 文件, 然后生成可执行文件, 并在 makefile 中包含删除中间文件 (.o) 的功能。</p> <p>(1) C 程序:</p> <pre>#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; }</pre> <p>确保文件名是 c1.c</p> <p>我们可以使用下面的 makefile</p>					

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

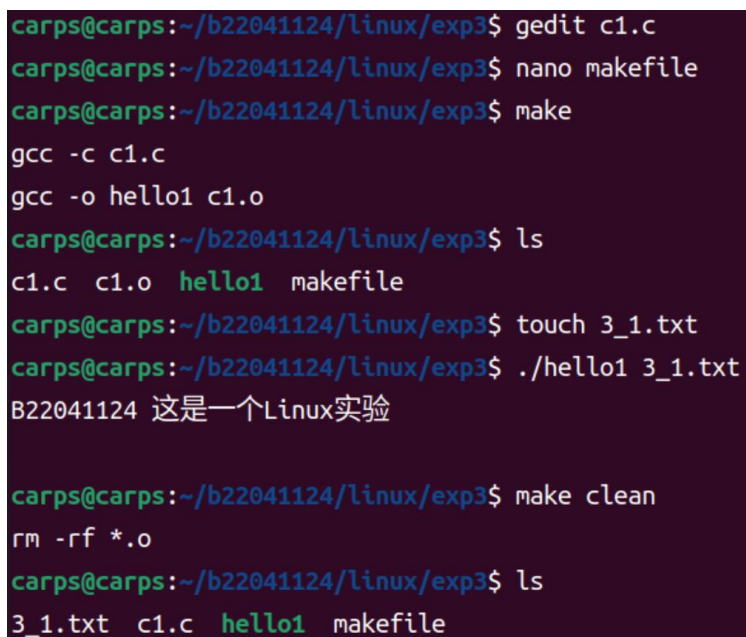
(2) 运行结果:

makefile 的内容



```
GNU nano 6.2
hello:c1.o
    gcc -o hello1 c1.o
c1.o:c1.c
    gcc -c c1.c
clean:
    rm -rf *.o
```

传入 3_1.txt 这个参数成功显示文本内容



```
carps@carps:~/b22041124/linux/exp3$ gedit c1.c
carps@carps:~/b22041124/linux/exp3$ nano makefile
carps@carps:~/b22041124/linux/exp3$ make
gcc -c c1.c
gcc -o hello1 c1.o
carps@carps:~/b22041124/linux/exp3$ ls
c1.c  c1.o  hello1  makefile
carps@carps:~/b22041124/linux/exp3$ touch 3_1.txt
carps@carps:~/b22041124/linux/exp3$ ./hello1 3_1.txt
B22041124 这是一个Linux实验

carps@carps:~/b22041124/linux/exp3$ make clean
rm -rf *.o
carps@carps:~/b22041124/linux/exp3$ ls
3_1.txt  c1.c  hello1  makefile
```

(3) 原因解释:

首先定义一个字符数组用于存储读取的行，并检查参数数量，如果没有提供，提示未提供文件名，并返回错误代码，然后检查文件是否成功打开，未成功则会打印错误信息并返回错误代码。接着逐行读取文件内容并打印，最后关闭文件

2. 编写一个 C 程序，显示当前目录中的所有文件名。该程序使用 **make** 工具进行编译和链接，要求首先生成 **.o** 文件，然后生成可执行文件，并在 **makefile** 中包含删除中间文件（**.o** 文件）的功能。

(1) C 程序

```
#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);
}
```

确保文件名为 c2.c 使用如下的 makefile:

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

(2) 运行结果:

```
carps@carps:~/b22041124/linux/exp3$ gedit c2.c
carps@carps:~/b22041124/linux/exp3$ nano makefile
carps@carps:~/b22041124/linux/exp3$ make
make: "hello2"已是最新。
carps@carps:~/b22041124/linux/exp3$ gedit c2.c
carps@carps:~/b22041124/linux/exp3$ nano makefile
carps@carps:~/b22041124/linux/exp3$ make
cc -c -o c2.o c2.c
gcc -o hello2 c2.o
carps@carps:~/b22041124/linux/exp3$ ls
3_1.txt 3_2 c1.c c2.c c2.o hello1 hello2 makefile
carps@carps:~/b22041124/linux/exp3$ ./hello2 3_2
123.sh
123.txt
.
..
carps@carps:~/b22041124/linux/exp3$ cd 3_2
carps@carps:~/b22041124/linux/exp3/3_2$ ls
123.sh 123.txt
```

(3) 原因解释:

程序接受一个命令行参数, 即要列出的目录的路径。使用 `opendir` 函数打开指定的目录 (必须是所在路径下的目录), 检查是否成功打开目录。如果打开失败, 程序会打印错误信息。使用 `readdir` 函数逐个读取目录项, 并打印每个项的名称。最后使用 `closedir` 关闭目录流。

3.编写一个 C 程序，改变当前进程的工作目录。该程序使用 **make** 工具编译和链接，首先需要生成 .o 文件，然后生成可执行文件，并在 **makefile** 中包含删除中间文件 (.o) 的功能。


(1) 输入命令：

```
#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;
}
```

确保文件名为 c3.c 使用如下的 makefile:

```
hello3:c3.o
    gcc -o hello3 c3.o
c3.o:c3.c
    gcc -c c3.c
clean:
    rm -rf *.
```

(2) 运行结果：



```
carps@carps:~/b22041124/linux/exp3/3_2$ gedit c3.c
carps@carps:~/b22041124/linux/exp3/3_2$ nano makefile
carps@carps:~/b22041124/linux/exp3/3_2$ make
gcc -c c3.c
gcc -o hello3 c3.o
carps@carps:~/b22041124/linux/exp3/3_2$ ./hello3
/home/carps/b22041124/linux/exp3/3_2
success
/home
```

(3) 原因解释：

使用 `getcwd` 函数获取并打印当前工作目录，然后使用 `chdir` 函数将工作目录更改为 `/home`，并检查 `chdir` 函数的返回值，以确定是否成功更改目录。如果更改成功，使用 `getcwd` 打印新的工作目录。

实 验 报 告

四、实验小结（包括问题和解决方法、心得体会、意见与建议、实验出错信息及解决方案等）

中文五号宋体，英文五号 Times new roman 字体，1.25 倍行距

说明：这部分内容主要包括：在编程、调试或测试过程中遇到的问题及解决方法、本次实验的心得体会、进一步改进的设想等。

（一）实验中遇到的主要问题及解决方法

问题一：编写 makefile 对于保存方式不熟悉，导致 makefile 没有成功保存，正确的是 Ctrl+X 退出，输入 Y（确认保存），然后直接回车。

问题二：前两个程序需要传递参数，尤其是第二个程序，需要在当前文件夹新建一个目录，其他路径下的目录尝试了一下好像不可以

（二）实验心得

本次实验主要是 Linux 的 C 语言编程和 make 管理工具的使用，对于 Linux 的 C 程序的编写和 Windows 基本一致，主要是对于编译和链接，以及怎么用 makefile 实现，一开始的确有些手足无措，但是多尝试几次就好很多了。

（三）意见与建议（没有可省略）

五、支撑毕业要求指标点

☒2-1-M 能够应用数学、自然科学、计算机科学和网络空间安全的基本原理对复杂安全问题的关键过程环节进行识别和判断。

☐3-2-H 能够在信息安全系统中合理的组织数据、有效存储和处理数据，正确的设计算法以及对算法进行分析和评价。

☐4-1-M 针对信息安全领域复杂工程问题，能基于专业理论并采用科学方法，设计可行的实验方案，具有根据解决方案进行工程实施的能力。

六、指导教师评语

评 分 细 则	评分项	优秀	良好	中等	合格	不合格
	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	课题功能实现情况					
	算法设计合理性					
	算法效能评价					
	回答问题准确度					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价 (总成绩)		批阅人		日期	