

一、实验目的

在 Linux 系统中进一步运用 C 语言编程的基础语法，加深对相关知识的理解与掌握。

二、实验要求

任务 1: 编写一个使用标准 I/O 库显示文本文件内容的 C 程序。要求通过 make 工具进行编译和链接，需先生成.o 目标文件，再生成可执行文件；同时 Makefile 中需包含删除中间文件（.o 文件）的功能，程序文件命名必须为 c1.c。

任务 2: 编写一个显示当前目录下所有文件名的 C 程序。要求通过 make 工具进行编译和链接，需先生成.o 目标文件，再生成可执行文件；同时 Makefile 中需包含删除中间文件（.o 文件）的功能，程序文件命名必须为 c2.c。

任务 3: 编写一个修改当前进程工作目录的 C 程序。要求通过 make 工具进行编译和链接，需先生成.o 目标文件，再生成可执行文件；同时 Makefile 中需包含删除中间文件（.o 文件）的功能，程序文件命名必须为 c3.c。

三、实验内容

（一）基于标准 I/O 库的文本文件内容显示程序

1、目标：编写 c1.c 读取显示指定文本文件内容；编写 Makefile，通过 make 命令完成编译流程，支持中间文件清理

2、编译步骤：执行 make 命令生成 c1.o 和 hello1，输出文件内容，输出正常即功能正常

3、操作截图：

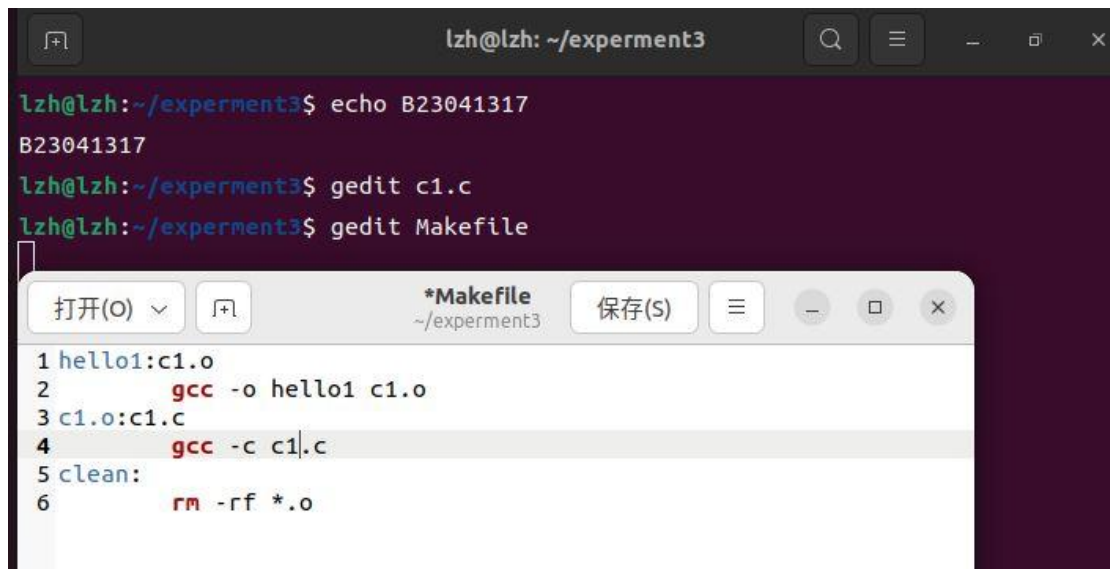
c1.c:



```
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c1.c
```

```
1 #include <stdio.h>
2 int main(int argc, char* argv[])
3 {
4     char buf[1024] = { 0 };
5     FILE* fp = fopen(argv[1], "r");
6     if (argc < 2)
7     {
8         printf("please input source file!\n");
9     }
10    if (fp == NULL)
11    {
12        printf("open source %s failed\n", argv[1]);
13        return -1;
14    }
15    while (fgets(buf, 1024, fp))
16    {
17        printf("%s\n", buf);
18    }
19    return 0;
20 }
```

Makefile:



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

```
lzh@lzh: ~/experment3
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c1.c
lzh@lzh:~/experment3$ gedit Makefile
```

A gedit window titled `*Makefile` is open, showing the following content:

```
1 hello1:c1.o
2     gcc -o hello1 c1.o
3 c1.o:c1.c
4     gcc -c c1.c
5 clean:
6     rm -rf *.o
```

结果:



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

```
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c1.c
lzh@lzh:~/experment3$ gedit Makefile
lzh@lzh:~/experment3$ make
make: "hello1"已是最新。
lzh@lzh:~/experment3$ ./hello1
please input source file!
open source (null) failed
lzh@lzh:~/experment3$ ./hello1 Makefile
hello1:c1.o

    gcc -o hello1 c1.o

c1.o:c1.c

    gcc -c c1.c

clean:

    rm -rf *.o

lzh@lzh:~/experment3$
```

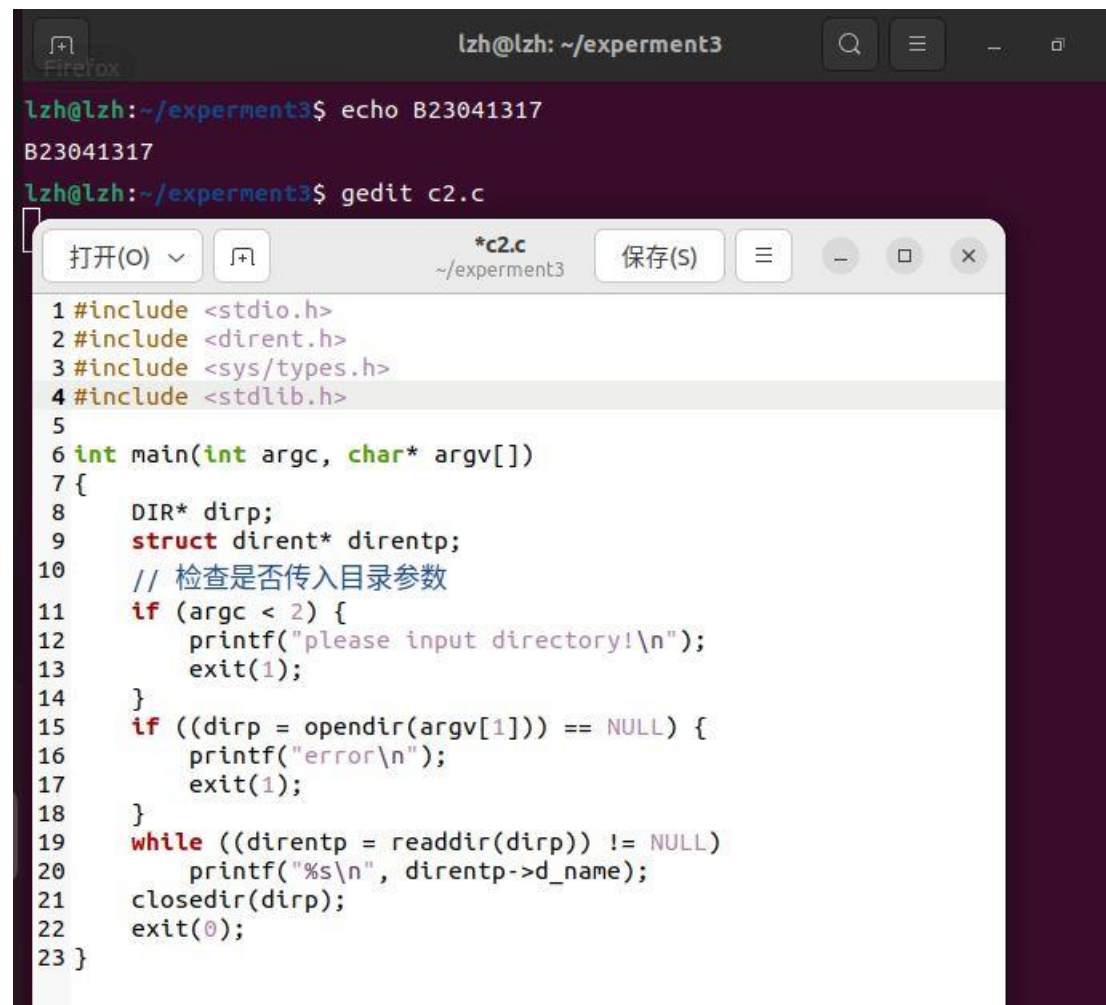
(二) 当前目录下所有文件名的便利显示程序

- 1、目标: 编写 `c2.c` 遍历显示指定目录下所有文件名; 编写 `Makefile`, 通过 `make` 命令完成编译流程, 支持中间文件清理
- 2、编译步骤: 执行 `make` 命令生成 `c2.o` 和 `hello2`, 遍历显示当前目录下所有文件, 输出正

常即功能正常

3、操作截图：

c2.c:



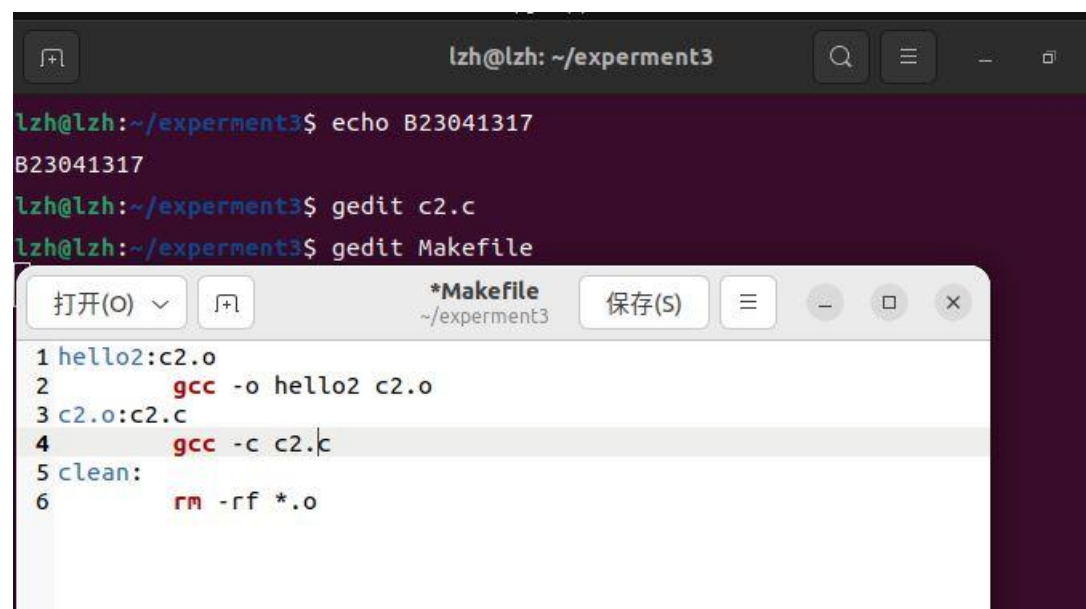
The screenshot shows a terminal window with the following commands and output:

```
lzh@lzh: ~/experment3
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c2.c
```

Below the terminal, a code editor window titled `*c2.c` is open, showing the following C code:

```
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <sys/types.h>
4 #include <stdlib.h>
5
6 int main(int argc, char* argv[])
7 {
8     DIR* dirp;
9     struct dirent* direntp;
10    // 检查是否传入目录参数
11    if (argc < 2) {
12        printf("please input directory!\n");
13        exit(1);
14    }
15    if ((dirp = opendir(argv[1])) == NULL) {
16        printf("error\n");
17        exit(1);
18    }
19    while ((direntp = readdir(dirp)) != NULL)
20        printf("%s\n", direntp->d_name);
21    closedir(dirp);
22    exit(0);
23 }
```

Makefile:



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

```
lzh@lzh: ~/experment3
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c2.c
lzh@lzh:~/experment3$ gedit Makefile
```

Below the terminal, a code editor window titled `*Makefile` is open, showing the following Makefile content:

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

结果:

```
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c2.c
lzh@lzh:~/experment3$ gedit Makefile
lzh@lzh:~/experment3$ make
gcc -c c2.c
gcc -o hello2 c2.o
lzh@lzh:~/experment3$ ./hello2 ./
..
c1.o
c3.c
Makefile2
c2.c
c11.c
c1.c
Makefile1
hello1
Makefile3
hello3
.
c2.o
Makefile
hello2
lzh@lzh:~/experment3$ ./hello2 /home
..
lizhenh
. 显示应用程序
lzh
```

(三) 当前进程工作目录修改程序


1、目标: 编写 c3.c 实现当前进程工作目录修改功能; 编写 Makefile, 通过 make 命令完成编译流程, 支持中间文件清理

2、编译步骤: 执行 make 命令生成 c3.o 和 hello3, 成功将工作目录修改

3、操作截图:

c3.c:

```
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c3.c
```




```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 int main()
5 {
6     char buf[1024] = {0};
7     char buf2[1024] = {0};
8     getcwd(buf, 1024);
9     printf("%s\n", buf);
10    if(chdir("/home") < 0){
11        printf("error\n");
12    }
13    else
14    {
15        printf("success\n");
16    }
17    getcwd(buf2, 1024);
18    printf("%s\n", buf2);
19    return 0;
20 }
```

Makefile:

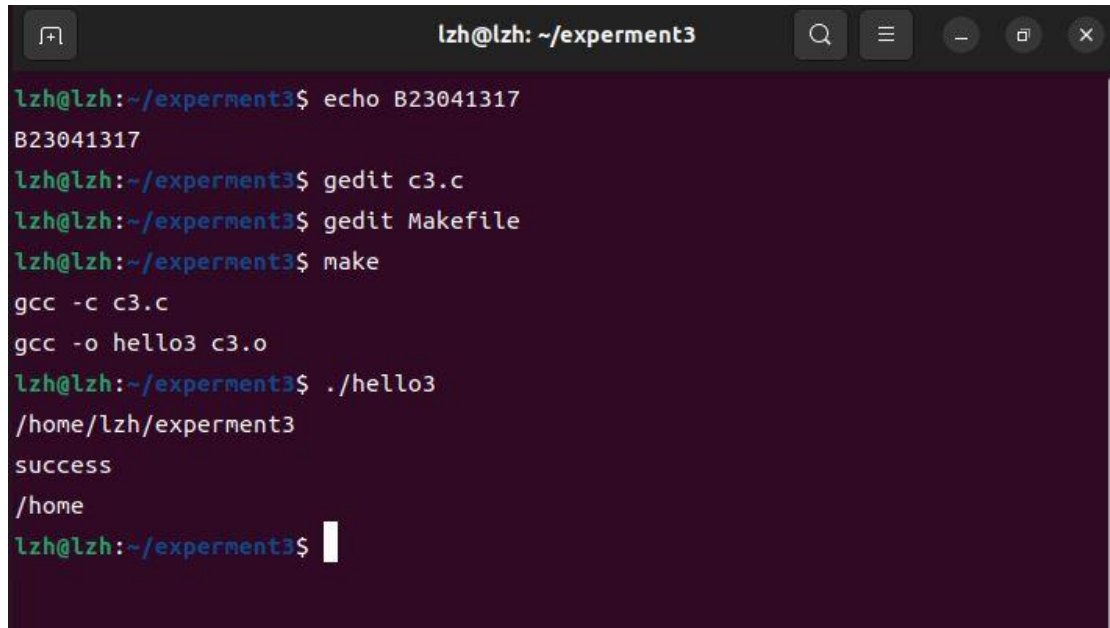
```
lzh@lzh: ~/experment3
```

```
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c3.c
lzh@lzh:~/experment3$ gedit Makefile
```



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

结果:



```
lzh@lzh: ~/experment3
lzh@lzh:~/experment3$ echo B23041317
B23041317
lzh@lzh:~/experment3$ gedit c3.c
lzh@lzh:~/experment3$ gedit Makefile
lzh@lzh:~/experment3$ make
gcc -c c3.c
gcc -o hello3 c3.o
lzh@lzh:~/experment3$ ./hello3
/home/lzh/experment3
success
/home
lzh@lzh:~/experment3$
```

四、实验问题即解决

在实验中，编写 makefile 之时，出现报错提示，了解后才发现缩进处需要用 tab 键缩进，修改后顺利运行！