

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	Linux 下 C 编程
实验时间	2024 年 12 月 13 日
指导单位	计算机学院网络空间安全系
指导教师	王磊

学生姓名	吴梦	班级学号	B22041204
学院(系)	计软网安学院	专 业	信息安全

实 验 报 告

实验名称	Linux 下 C 编程			指导教师	王磊
实验类型	上机	实验学时	2	实验时间	2024.12.13
<p>一、 实验目的和要求</p> <p>1. 在 Linux 系统中进一步使用 C 编程语言的基本语法，加深对知识的理解</p>					
<p>二、实验环境(实验设备)</p> <p>虚拟机下的 Ubuntu 系统</p>					
<p>三、实验内容</p> <p>(1) 任务 1</p> <p>1. 要求：编写一个使用标准 I/O 库显示文本文件内容的 C 程序。该程序由 <code>make</code> 工具编译和链接，该工具需要首先生成 <code>.o</code> 文件，然后生成可执行文件，并具有删除 <code>makefile</code> 文件中的中间文件（<code>.o</code>）的功能。</p>					

```

#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;
}

```

请确保您的文件名为 `c1.c`。
我们可以使用以下 `makefile`

```

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

```

2.我的实验： 如图所示，实现了显示文本文件内容的功能，我还尝试了其他可选项 `-w`、`-s` 等的功能。

```
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 }
21 }
22
```

```
ubuntu@ubuntu-virtual-machine: ~/桌面/B22041204-C编程/C1
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ vim hello1.txt
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ cat hello1.txt
hello!The filename is helloc.txt
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make
gcc -c c1.c
gcc -o hello1 c1.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make clean
rm -rf *.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ ./hello1 hello1.txt
hello!The filename is helloc.txt
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$
```

```
ubuntu@ubuntu-virtual-machine: ~/桌面/B22041204-C编程/C1
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make
gcc -c c1.c
gcc -o hello1 c1.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ ./hello1 hello1.txt
hello!The filename is hello1.txt
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make clean
rm -rf *.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ rm hello1
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make -s
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ ./hello1 hello1.txt
hello!The filename is hello1.txt
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make clean
rm -rf *.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ rm hello1
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$ make -w
make: 进入目录"/home/ubuntu/桌面/B22041204-C编程/C1"
gcc -c c1.c
gcc -o hello1 c1.o
make: 离开目录"/home/ubuntu/桌面/B22041204-C编程/C1"
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C编程/C1$
```

创建 目标hello1

执行程序，显示文件内容

在执行命令时不显示

在处理规则前后显示工作目录

(2)任务 2

1. 要求：编写一个 C 程序，显示当前目录中的所有文件名。该程序由 make 工具编译和链接，它需要首先生成.o 文件，然后生成可执行文件，并删除 makefile 文件中的中间文件（.o）。

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);
}
```

Make sure your filename is c2.c

We can use the following makefile.

```
hello2:c2.o
    gcc -o hello1 c2.o
c2.o:c2.c
    gcc -c c2.c
clean:
    rm -rf *.o
```

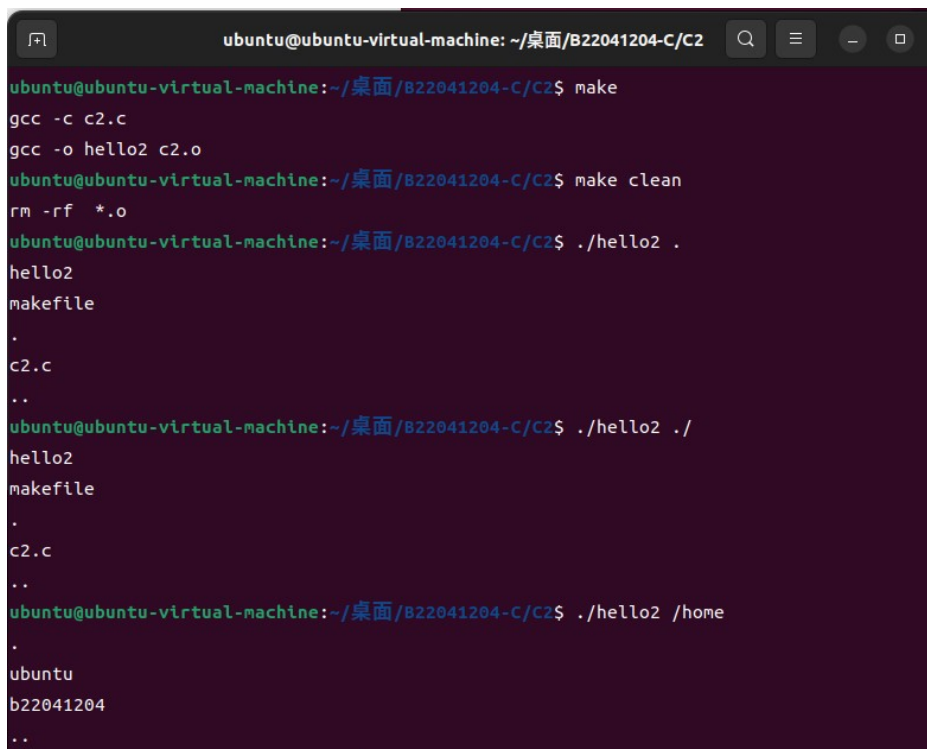
2. 我的实验:



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



```
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
```



```
ubuntu@ubuntu-virtual-machine: ~/桌面/B22041204-C/C2
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ make
gcc -c c2.c
gcc -o hello2 c2.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ make clean
rm -rf *.o
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ ./hello2 .
hello2
makefile
.
c2.c
..
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ ./hello2 ./
hello2
makefile
.
c2.c
..
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ ./hello2 /home
.
ubuntu
b22041204
..
```

```
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$ ./hello2 /home/ubuntu
.config
.bashrc
.lesshst
.
.local
.ssh
图片
音乐
.sudo_as_admin_successful
.bash_history
others
视频
snap
.bash_logout
..
.viminfo
backup.tar.gz
下载
.cache
桌面
模板
.profile
文档
公共的
.gnupg
ubuntu@ubuntu-virtual-machine:~/桌面/B22041204-C/C2$
```

(3)任务 3

1.要求：编写一个 C 程序，更改当前进程的工作目录。该程序由 make 工具编译和链接，该工具需要首先生成.o 文件，然后生成可执行文件，并具有删除中间文件（.o）的功能


```
#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;
}
```

您的文件名是 c3.c

我们可以使用以下 makefile

```
hello3:c3.o
    gcc -o hello1 c3.o
c3.o:c3.c
    gcc -c c3.c
clean:
    rm -rf *.o
```

2. 我的实验：结果如图所示，分析如下：

①获取当前工作目录：

getcwd(buf, 1024)：调用 getcwd 函数获取当前工作目录，并将其存储在 buf 数组中。

printf("%s\n", buf)：打印当前工作目录。

②更改工作目录：

if (chdir("/home") < 0)：尝试将当前工作目录更改为/home。如果更改失败（返回值小于 0），则打印 "error"。

else：如果更改成功，则打印 "success"。

③再次获取当前工作目录：

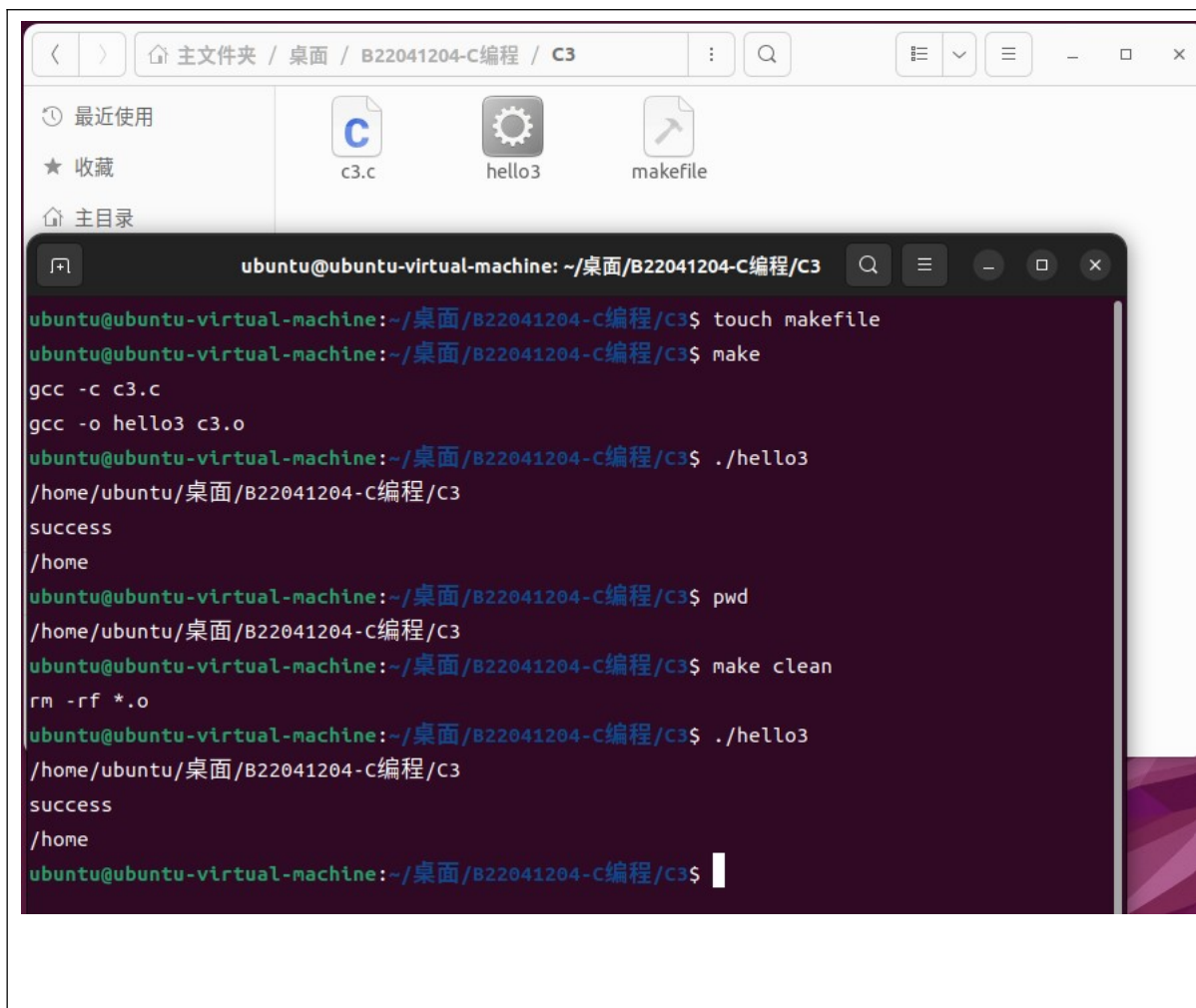
getcwd(buf2, 1024): 调用 getcwd 函数获取新的当前工作目录, 并将其存储在 buf2 数组中。
printf("%s\n", buf2): 打印新的当前工作目录



```
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);
10    printf("%s\n",buf);
11    if(chdir("/home")<0)
12    {
13        printf("error\n");
14    }
15    else
16    {
17        printf("success\n");
18    }
19 }
20 getcwd(buf2,1024);
21 printf("%s\n",buf2);
22 return 0;
23
24 }
```



```
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
```



四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

五、指导教师评语

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