

1. 实现文本文件内容显示程序

在终端输入 gedit c1.c 创建脚本文件，粘贴代码后保存并退出

```
#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;
}
```

在终端输入 gedit Makefile 创建文件，粘贴代码后保存并退出

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

在终端执行 make 命令，会自动按 Makefile 规则生成 c1.o 中间文件和 hello1 可执行文件

```
b23041310matao@b23041310matao-virtual-machine:~$ gedit c1.c
b23041310matao@b23041310matao-virtual-machine:~$ gedit Makefile
b23041310matao@b23041310matao-virtual-machine:~$ make

Command 'make' not found, but can be installed with:

sudo apt install make
sudo apt install make-guile

b23041310matao@b23041310matao-virtual-machine:~$
```

这个问题是因为系统还没安装 make 工具，在终端中执行以下命令安装 make 工具即可解决：

```
sudo apt install make
```

```
b23041310matao@b23041310matao-virtual-machine:~$ sudo apt install make
[sudo] b23041310matao 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
  gir1.2-goa-1.0 gir1.2-snapd-1
使用'sudo apt autoremove'来卸载它(它们)。
建议安装:
  make-doc
下列【新】软件包将被安装:
  make
升级了 0 个软件包, 新安装了 1 个软件包, 要卸载 0 个软件包, 有 0 个软件包未被升级
。需要下载 154 kB 的归档。
解压缩后会消耗 381 kB 的额外空间。
获取:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu bionic/main amd64 make amd64 4
.1-9.1ubuntu1 [154 kB]
已下载 154 kB, 耗时 0 秒 (378 kB/s)
正在选中未选择的软件包 make。
(正在读取数据库 ... 系统当前共安装有 169982 个文件和目录。)
```

重新在终端执行 make 命令

```
b23041310matao@b23041310matao-virtual-machine:~$ make
Makefile:2: *** 遗漏分隔符 (null)。 停止。
b23041310matao@b23041310matao-virtual-machine:~$
```

这个错误是 Makefile 的语法问题: Makefile 中, 目标对应的命令行 (比如 gcc 开头的行) 必须用 Tab 键缩进, 而代码使用了空格缩进, 导致 make 识别失败。打开 Makefile 文件, 修正缩进为 Tab 键

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

重新在终端执行 make 命令

```
b23041310matao@b23041310matao-virtual-machine:~$ make
Makefile:2: *** 遗漏分隔符 (null)。 停止。
b23041310matao@b23041310matao-virtual-machine:~$ gedit Makefile
b23041310matao@b23041310matao-virtual-machine:~$ make
gcc -c c1.c
gcc -o hello1 c1.o
b23041310matao@b23041310matao-virtual-machine:~$
```

先创建一个测试文本文件 (如 test.txt) 用于验证:

```
echo -e "this is test line1\nthis is test line2" > test.txt
```

```
b23041310matao@b23041310matao-virtual-machine:~$ echo -e "this is test line1\nth
is is test line2" > test.txt
b23041310matao@b23041310matao-virtual-machine:~$
```

执行可执行文件, 传入测试文件路径:

```
./hello1 test.txt
```

```
b23041310matao@b23041310matao-virtual-machine:~$ ./hello1 test.txt
this is test line1
this is test line2
b23041310matao@b23041310matao-virtual-machine:~$
```

执行 make clean 删除.o 中间文件:

```
b23041310matao@b23041310matao-virtual-machine:~$ make clean
rm -rf *.o
b23041310matao@b23041310matao-virtual-machine:~$
```

2. 实现显示当前目录文件名程序

在终端输入 gedit c2.c 创建文件，粘贴代码后保存并退出

```
#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);
}
```

为避免和 Task1 的 Makefile 冲突，可删除原 Makefile 后新建并注意用 Tab 键缩进替换代码中空格，保存代码并退出

rm -f Makefile

gedit Makefile

```
b23041310matao@b23041310matao-virtual-machine:~$ gedit c2.c
b23041310matao@b23041310matao-virtual-machine:~$ rm -f Makefile
b23041310matao@b23041310matao-virtual-machine:~$ gedit Makefile
b23041310matao@b23041310matao-virtual-machine:~$
```

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

执行 make 命令生成 c2.o 和 hello2:

make

```
b23041310matao@b23041310matao-virtual-machine:~$ make
gcc -c c2.c
gcc -o hello2 c2.o
b23041310matao@b23041310matao-virtual-machine:~$
```

传入当前目录路径 (. 代表当前目录) 执行程序，查看目录下所有文件名：

./hello2 .

```
b23041310matao@b23041310matao-virtual-machine:~$ ./hello2 .
下载
hello2
.bashrc
c2.o
.ICEauthority
cert.p12
.bash_logout
sample
test.txt
图片
matao2.txt
decrypt2.txt
.bash_history
.
c2.c
check_prime.sh
time_check.sh
音乐
视频
page.txt
result.txt
cbc_no_base64.enc
.cache
cipher.enc
examples.desktop
桌面
文档
公共的
.aes_test.txt.swp
```

```
.aes_test.txt.swp
.config
模板
find_min.sh
sign.sha256
B23041310
rsa_public.key
page_rep
cipher.txt
Makefile
decrypt1.txt
page_replacement.c
aes_test.txt
rsa_private.key
hello1
process_image
.local
.cipher.txt.swp
num_compare.sh
.gnupg
c1.c
count_exec.sh
sample.c
B23041310.txt
ecb_no_base64.enc
ecb_base64.enc
process_image.c
.profile
cbc_base64.enc
ca.crt
..
.sudo_as_admin_successful
b23041310matao@b23041310matao-virtual-machine:~$
```

清理中间文件

make clean

```
b23041310matao@b23041310matao-virtual-machine:~$ make clean
rm -rf *.o
b23041310matao@b23041310matao-virtual-machine:~$
```

重新查看目录下所有文件名

```
b23041310matao@b23041310matao-virtual-machine:~$ ./hello2 .
下载
hello2
.bashrc
.ICEauthority
cert.piz
.bash_logout
sample
test.txt
图片
matao2.txt
decrypt2.txt
.bash_history
.
c2.c
check_prime.sh
time_check.sh
音乐
视频
page.txt
result.txt
cbc_no_base64.enc
.cache
cipher.enc
examples.desktop
桌面
文档
公共的
.aes_test.txt.swp
```

可以看到已删除中间文件.o

3. 实现修改当前进程工作目录程序

在终端输入 gedit c3.c 创建文件，粘贴代码后保存并退出

```
#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;
}
```

删除旧 Makefile，新建适配 Task3 的 Makefile，粘贴代码后保存并退出

rm -f Makefile

gedit Makefile

```
b23041310matao@b23041310matao-virtual-machine:~$ gedit c3.c
b23041310matao@b23041310matao-virtual-machine:~$ rm -f Makefile
b23041310matao@b23041310matao-virtual-machine:~$ gedit Makefile
```

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

编译程序：

make

```
b23041310matao@b23041310matao-virtual-machine:~$ make
gcc -c c3.c
gcc -o hello3 c3.o
b23041310matao@b23041310matao-virtual-machine:~$
```

直接执行可执行文件，查看工作目录变更前后的路径：

./hello3

```
b23041310matao@b23041310matao-virtual-machine:~$ ./hello3
/home/b23041310matao
success
/home
b23041310matao@b23041310matao-virtual-machine:~$
```

会先显示当前实验目录路径，再输出 success，最后显示/home 路径

清理中间文件：

make clean

```
b23041310matao@b23041310matao-virtual-machine:~$ make clean
rm -rf *.o
b23041310matao@b23041310matao-virtual-machine:~$
```