

(1) c1



```
1 #include <stdio.h>
2
3 int main(int argc, char* argv[]) {
4     // 检查是否传入文件名参数
5     if (argc < 2) {
6         printf("Please input source file!\n");
7         return -1;
8     }
9
10    char buf[1024] = { 0 };
11    // 打开文件
12    FILE* fp = fopen(argv[1], "r");
13    if (fp == NULL) {
14        printf("Open source file %s failed\n", argv[1]);
15        return -1;
16    }
17
18    // 读取文件内容并输出
19    while (fgets(buf, sizeof(buf), fp)) {
20        printf("%s", buf);
21    }
22
23    // 关闭文件
24    fclose(fp);
25    return 0;
26 }
```

makefile



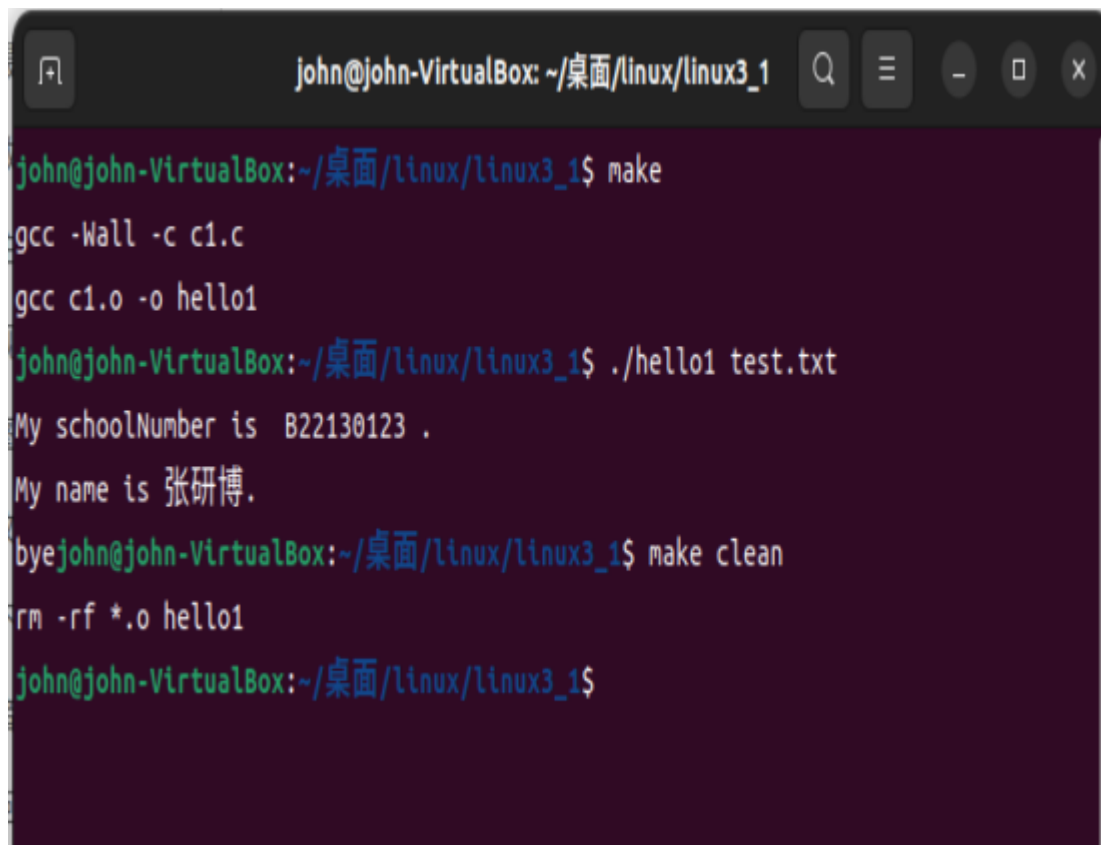
The image shows a text editor window with the title bar 'Makefile' and a path '~/.桌面/linux/linux3_1'. The editor contains a Makefile script for compiling and linking a C program. The script includes comments in Chinese and uses standard Makefile syntax with variables for compiler, flags, target, and object files. It defines a default target 'all' and a 'clean' target to remove intermediate files and the executable.

```
1 # Makefile for compiling and linking c1.c
2
3 # 编译器设置
4 CC = gcc
5 CFLAGS = -Wall
6
7 # 最终目标文件
8 TARGET = hello1
9
10 # 目标文件
11 OBJ = c1.o
12
13 # 默认目标: 编译并链接
14 all: $(TARGET)
15
16 # 链接目标文件生成可执行文件
17 $(TARGET): $(OBJ)
18     $(CC) $(OBJ) -o $(TARGET)
19
20 # 编译源代码为目标文件
21 $(OBJ): c1.c
22     $(CC) $(CFLAGS) -c c1.c
23
24 # 清理目标, 删除中间文件和可执行文件
25 clean:
26     rm -rf *.o $(TARGET)
```

test.txt



运行结果

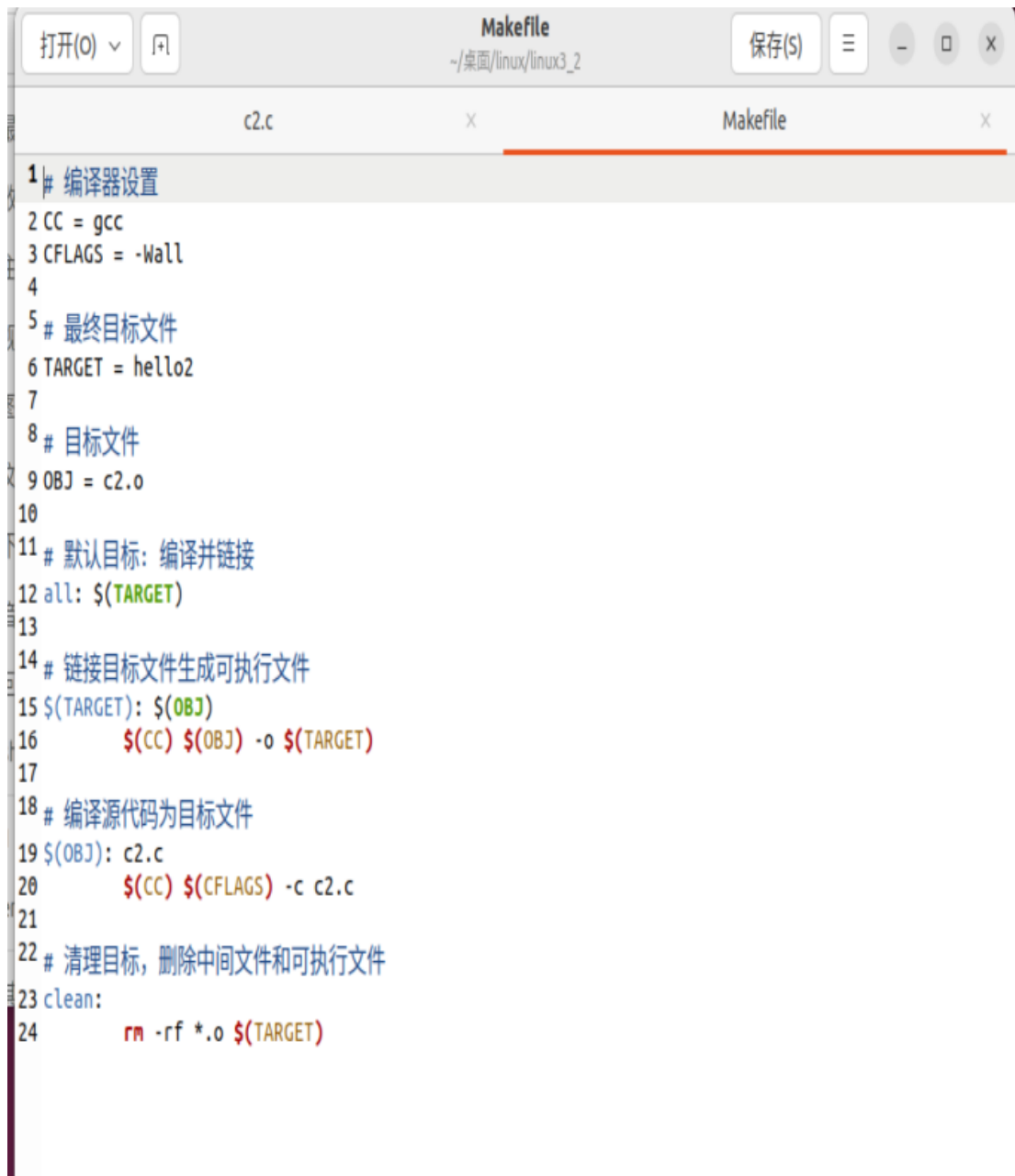


(2) c2



```
1#include <stdio.h>
2#include <stdlib.h>
3#include <dirent.h>
4#include <sys/types.h>
5
6int main(int argc, char *argv[]) {
7    if (argc < 2) {
8        printf("Please provide a directory path.\n");
9        return -1;
10    }
11
12    DIR *dirp;
13    struct dirent *direntp;
14
15    // 打开目录
16    dirp = opendir(argv[1]);
17    if (dirp == NULL) {
18        perror("Error opening directory");
19        return -1;
20    }
21
22    // 读取目录内容
23    while ((direntp = readdir(dirp)) != NULL) {
24        printf("%s\n", direntp->d_name); // 打印文件名
25    }
26
27    // 关闭目录
28    closedir(dirp);
29
30    return 0;
31 }
```

makefile



The image shows a window titled "Makefile" with a path of "~/桌面/linux/linux3_2". The window contains a Makefile for a C program named "c2.c". The Makefile defines variables for the compiler (CC = gcc), flags (CFLAGS = -Wall), the target (TARGET = hello2), and the object file (OBJ = c2.o). It also defines the default target as "all" which builds the target. The "all" target depends on the object file, which is built from "c2.c" using the compiler and flags. A "clean" target is also defined to remove the object file and the target.

```
1 # 编译器设置
2 CC = gcc
3 CFLAGS = -Wall
4
5 # 最终目标文件
6 TARGET = hello2
7
8 # 目标文件
9 OBJ = c2.o
10
11 # 默认目标: 编译并链接
12 all: $(TARGET)
13
14 # 链接目标文件生成可执行文件
15 $(TARGET): $(OBJ)
16     $(CC) $(OBJ) -o $(TARGET)
17
18 # 编译源代码为目标文件
19 $(OBJ): c2.c
20     $(CC) $(CFLAGS) -c c2.c
21
22 # 清理目标, 删除中间文件和可执行文件
23 clean:
24     rm -rf *.o $(TARGET)
```

运行结果

```
john@john-VirtualBox:~/桌面/linux/linux3_2$ make
gcc -Wall -c c2.c
gcc c2.o -o hello2
john@john-VirtualBox:~/桌面/linux/linux3_2$ ./hello2 /path/list
Error opening directory: No such file or directory
john@john-VirtualBox:~/桌面/linux/linux3_2$ ./hello2 /home/user
Error opening directory: No such file or directory
john@john-VirtualBox:~/桌面/linux/linux3_2$ ./hello2 /home/john
```

下载

.sudo_as_admin_successful

eaiInferer-Src

.java

EASY-EAI

.presage

.viminfo

.gnupg

图片

.

.ssh

.local

.pam_environment

test.txt

音乐

桌面

```
john@john-VirtualBox:~/桌面/linux/linux3_2$ make clean
rm -rf *.o hello2
```

(3) c3



```
1#include <stdio.h>
2#include <stdlib.h>
3#include <unistd.h>
4
5int main() {
6    char buf[1024] = {0};
7    char buf2[1024] = {0};
8
9    // 获取当前工作目录
10   if (getcwd(buf, sizeof(buf)) == NULL) {
11       perror("getcwd failed");
12       return -1;
13   }
14   printf("Current directory: %s\n", buf);
15
16   // 更改工作目录
17   if (chdir("/home") < 0) {
18       perror("chdir failed");
19   } else {
20       printf("Directory changed successfully to /home\n");
21   }
22
23   // 获取更改后的工作目录
24   if (getcwd(buf2, sizeof(buf2)) == NULL) {
25       perror("getcwd failed");
26       return -1;
27   }
28   printf("New current directory: %s\n", buf2);
29
30   return 0;
31 }
```

makefile



```
1 # 编译器设置
2 CC = gcc
3 CFLAGS = -Wall
4
5 # 最终目标文件
6 TARGET = hello3
7
8 # 目标文件
9 OBJ = c3.o
10
11 # 默认目标: 编译并链接
12 all: $(TARGET)
13
14 # 链接目标文件生成可执行文件
15 $(TARGET): $(OBJ)
16     $(CC) $(OBJ) -o $(TARGET)
17
18 # 编译源代码为目标文件
19 $(OBJ): c3.c
20     $(CC) $(CFLAGS) -c c3.c
21
22 # 清理目标, 删除中间文件和可执行文件
23 clean:
24     rm -rf *.o $(TARGET)
```

运行结果


```
john@john-VirtualBox: ~/桌面/linux/linux3_3
john@john-VirtualBox:~/桌面/linux/linux3_3$ make
gcc -Wall -c c3.c
gcc c3.o -o hello3
john@john-VirtualBox:~/桌面/linux/linux3_3$ ./hello3
Current directory: /home/john/桌面/linux/linux3_3
Directory changed successfully to /home
New current directory: /home
john@john-VirtualBox:~/桌面/linux/linux3_3$
```