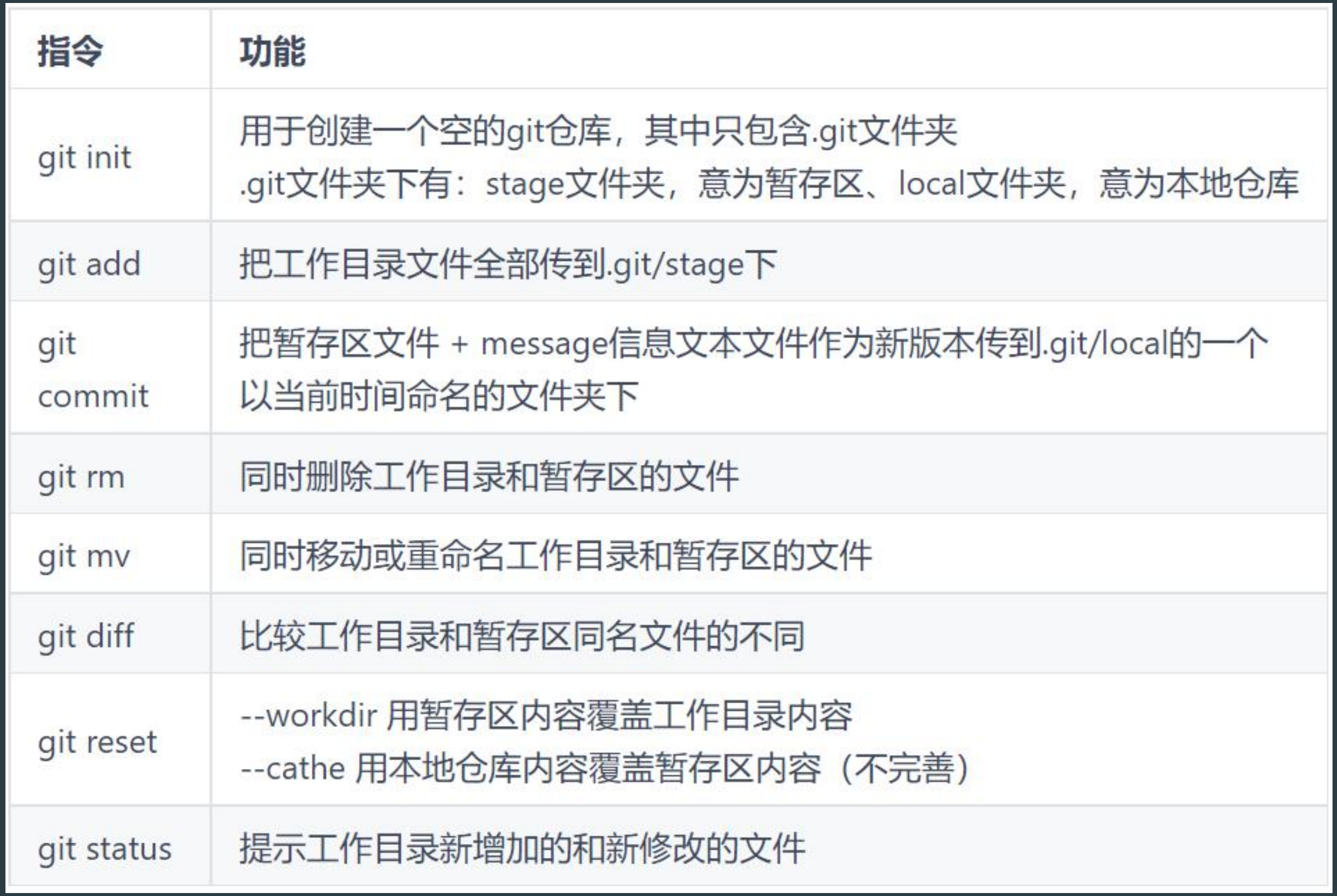
# C++夏令营结营答辩

李瑞堃

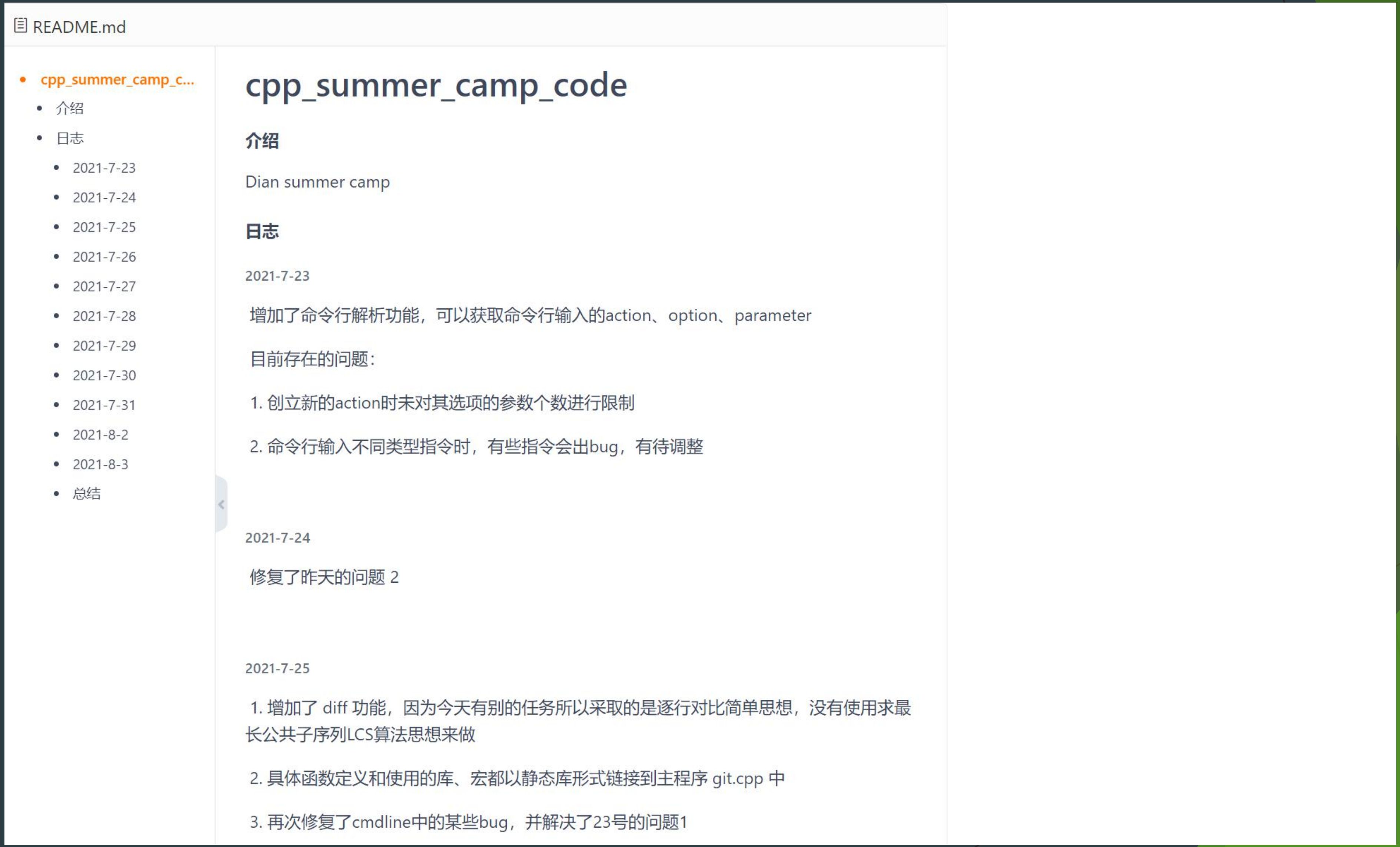
# 总体完成情况：

- 主程序：git.cpp
- 实现功能：cmdline、init、status、diff、commit、add、rm、mv、reset



| 指令 | 功能 |
|---|---|
| git init | 用于创建一个空的git仓库，其中只包含.git文件夹<br>.git文件夹下有：stage文件夹，意为暂存区、local文件夹，意为本地仓库 |
| git add | 把工作目录文件全部传到.git/stage下 |
| git commit | 把暂存区文件 + message信息文本文件作为新版本传到.git/local的一个以当前时间命名的文件夹下 |
| git rm | 同时删除工作目录和暂存区的文件 |
| git mv | 同时移动或重命名工作目录和暂存区的文件 |
| git diff | 比较工作目录和暂存区同名文件的不同 |
| git reset | --workdir 用暂存区内容覆盖工作目录内容<br>--cathe 用本地仓库内容覆盖暂存区内容（不完善） |
| git status | 提示工作目录新增加的和新修改的文件 |

# 进度记录：



README.md

## cpp_summer_camp_code

### 介绍

Dian summer camp

### 日志

2021-7-23

增加了命令行解析功能，可以获取命令行输入的action、option、parameter

目前存在的问题：

1. 创立新的action时未对其选项的参数个数进行限制

2. 命令行输入不同类型指令时，有些指令会出bug，有待调整

2021-7-24

修复了昨天的问题 2

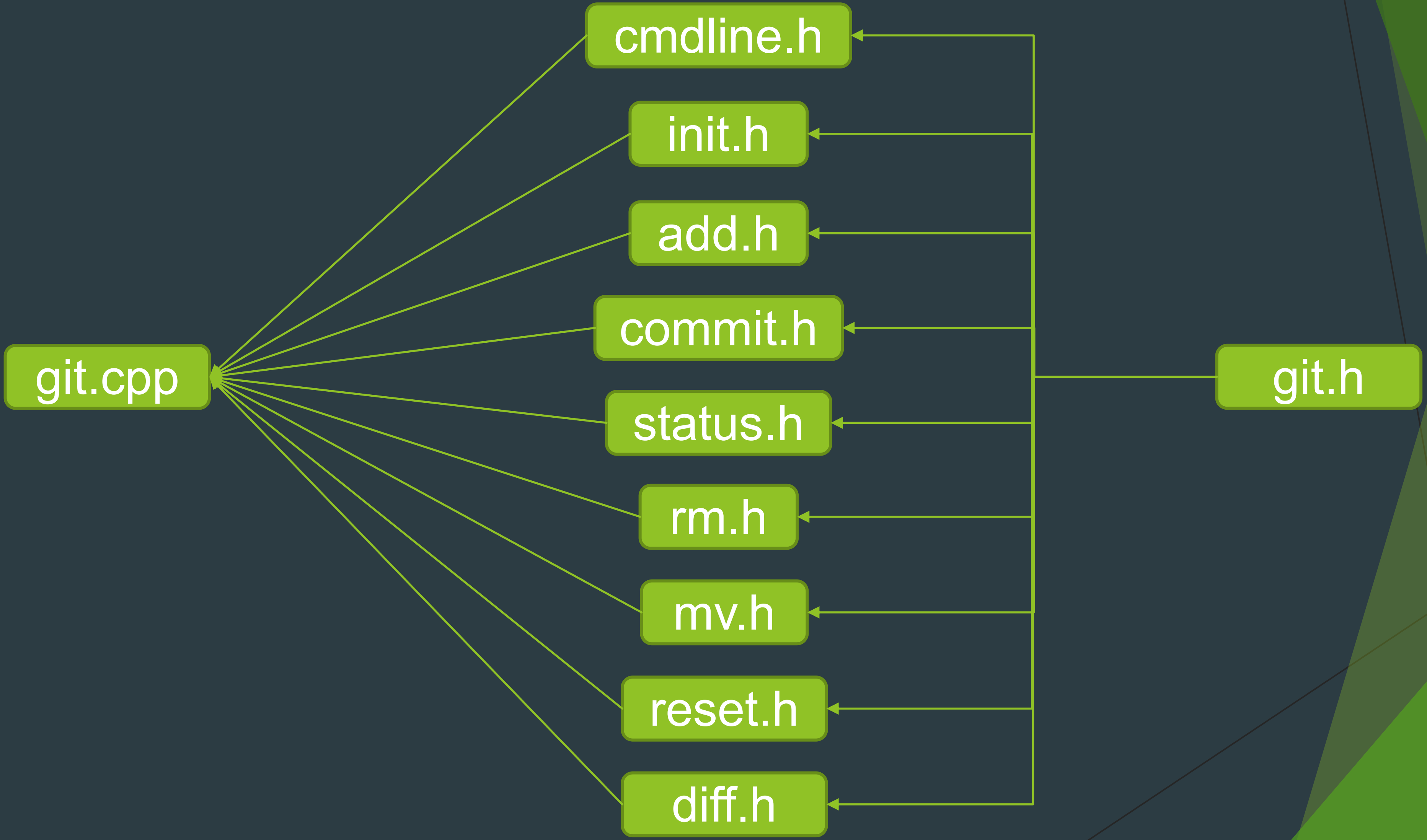
2021-7-25

1. 增加了 diff 功能，因为今天有别的任务所以采取的是逐行对比简单思想，没有使用求最长公共子序列LCS算法思想来做

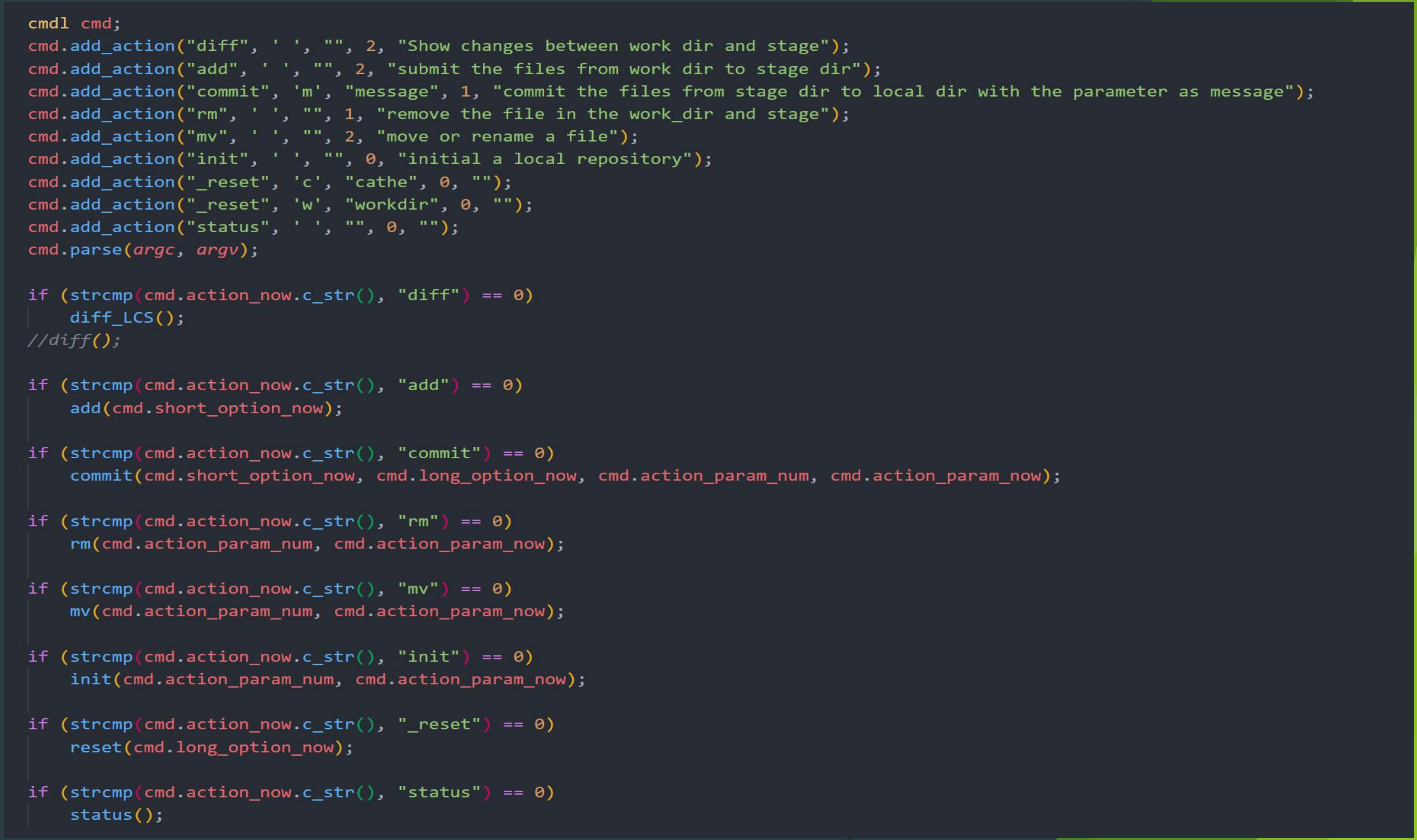
2. 具体函数定义和使用的库、宏都以静态库形式链接到主程序 git.cpp 中

3. 再次修复了cmdline中的某些bug，并解决了23号的问题1

# 代码结构：


cmdline.h
init.h
add.h
commit.h
git.cpp
status.h
git.h
rm.h
mv.h
reset.h
diff.h

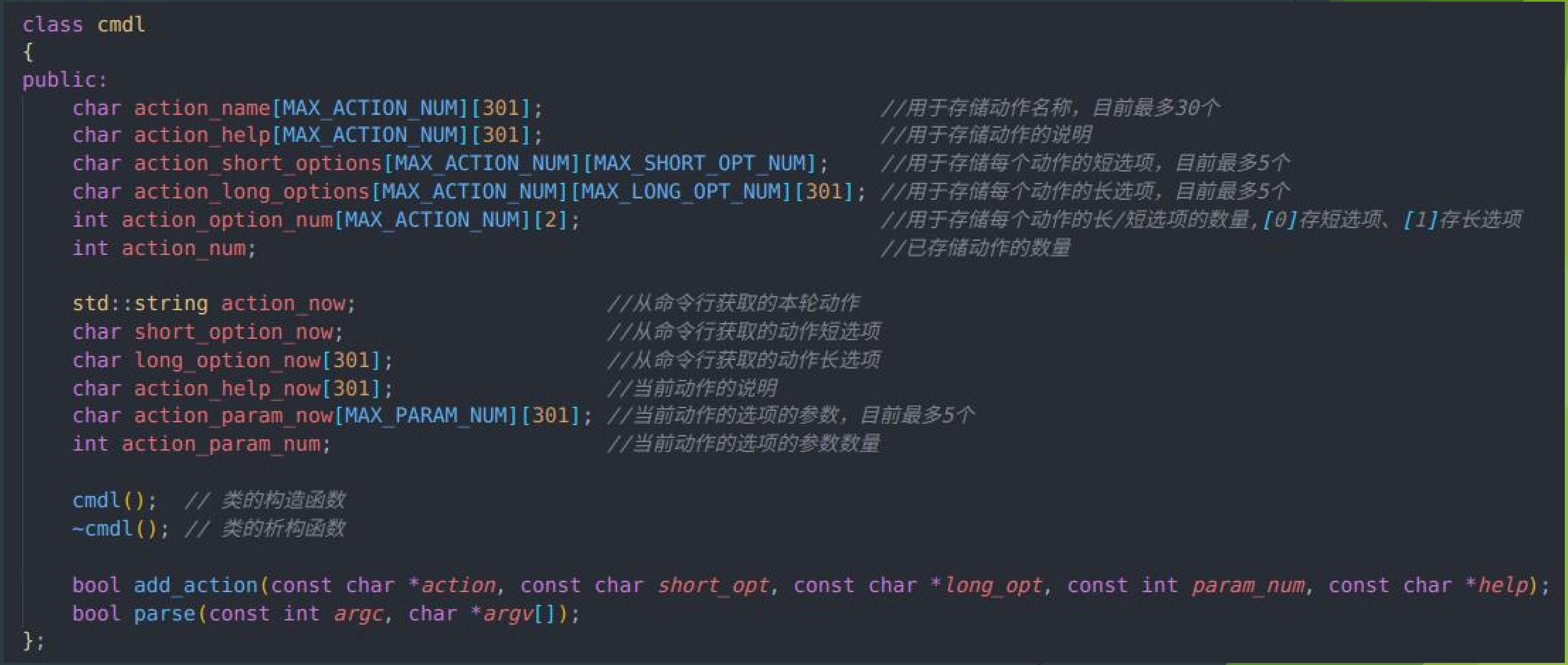
# git.cpp



```
cmdl cmd;
cmd.add_action("diff", ' ', "", 2, "Show changes between work dir and stage");
cmd.add_action("add", ' ', "", 2, "submit the files from work dir to stage dir");
cmd.add_action("commit", 'm', "message", 1, "commit the files from stage dir to local dir with the parameter as message");
cmd.add_action("rm", ' ', "", 1, "remove the file in the work_dir and stage");
cmd.add_action("mv", ' ', "", 2, "move or rename a file");
cmd.add_action("init", ' ', "", 0, "initial a local repository");
cmd.add_action("_reset", 'c', "cathe", 0, "");
cmd.add_action("_reset", 'w', "workdir", 0, "");
cmd.add_action("status", ' ', "", 0, "");
cmd.parse(argc, argv);

if (strcmp(cmd.action_now.c_str(), "diff") == 0)
    diff_LCS();
//diff();

if (strcmp(cmd.action_now.c_str(), "add") == 0)
    add(cmd.short_option_now);

if (strcmp(cmd.action_now.c_str(), "commit") == 0)
    commit(cmd.short_option_now, cmd.long_option_now, cmd.action_param_num, cmd.action_param_now);

if (strcmp(cmd.action_now.c_str(), "rm") == 0)
    rm(cmd.action_param_num, cmd.action_param_now);

if (strcmp(cmd.action_now.c_str(), "mv") == 0)
    mv(cmd.action_param_num, cmd.action_param_now);

if (strcmp(cmd.action_now.c_str(), "init") == 0)
    init(cmd.action_param_num, cmd.action_param_now);

if (strcmp(cmd.action_now.c_str(), "_reset") == 0)
    reset(cmd.long_option_now);

if (strcmp(cmd.action_now.c_str(), "status") == 0)
    status();
```

# Cmdline

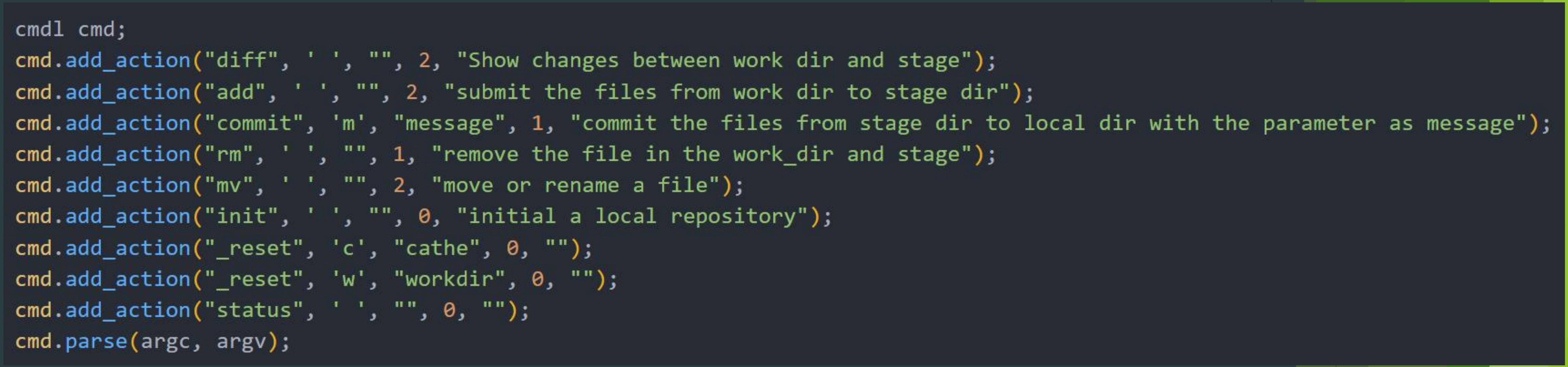
类定义及实现思路：



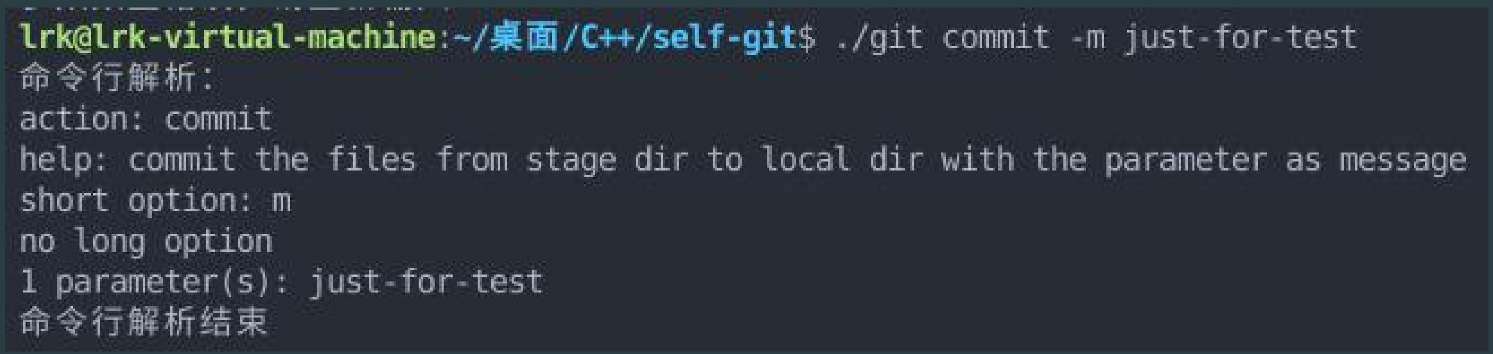
```
class cmdl
{
public:
    char action_name[MAX_ACTION_NUM][301];                           //用于存储动作名称，目前最多30个
    char action_help[MAX_ACTION_NUM][301];                           //用于存储动作的说明
    char action_short_options[MAX_ACTION_NUM][MAX_SHORT_OPT_NUM];    //用于存储每个动作的短选项，目前最多5个
    char action_long_options[MAX_ACTION_NUM][MAX_LONG_OPT_NUM][301]; //用于存储每个动作的长选项，目前最多5个
    int action_option_num[MAX_ACTION_NUM][2];                        //用于存储每个动作的长/短选项的数量,[0]存短选项、[1]存长选项
    int action_num;                                                  //已存储动作的数量

    std::string action_now;                    //从命令行获取的本轮动作
    char short_option_now;                     //从命令行获取的动作短选项
    char long_option_now[301];                 //从命令行获取的动作长选项
    char action_help_now[301];                 //当前动作的说明
    char action_param_now[MAX_PARAM_NUM][301]; //当前动作的选项的参数，目前最多5个
    int action_param_num;                      //当前动作的选项的参数数量

    cmdl();  // 类的构造函数
    ~cmdl(); // 类的析构函数

    bool add_action(const char *action, const char short_opt, const char *long_opt, const int param_num, const char *help);
    bool parse(const int argc, char *argv[]);
};
```

# Cmdline

使用及效果：



```
cmdl cmd;
cmd.add_action("diff", ' ', "", 2, "Show changes between work dir and stage");
cmd.add_action("add", ' ', "", 2, "submit the files from work dir to stage dir");
cmd.add_action("commit", 'm', "message", 1, "commit the files from stage dir to local dir with the parameter as message");
cmd.add_action("rm", ' ', "", 1, "remove the file in the work_dir and stage");
cmd.add_action("mv", ' ', "", 2, "move or rename a file");
cmd.add_action("init", ' ', "", 0, "initial a local repository");
cmd.add_action("_reset", 'c', "cathe", 0, "");
cmd.add_action("_reset", 'w', "workdir", 0, "");
cmd.add_action("status", ' ', "", 0, "");
cmd.parse(argc, argv);
```



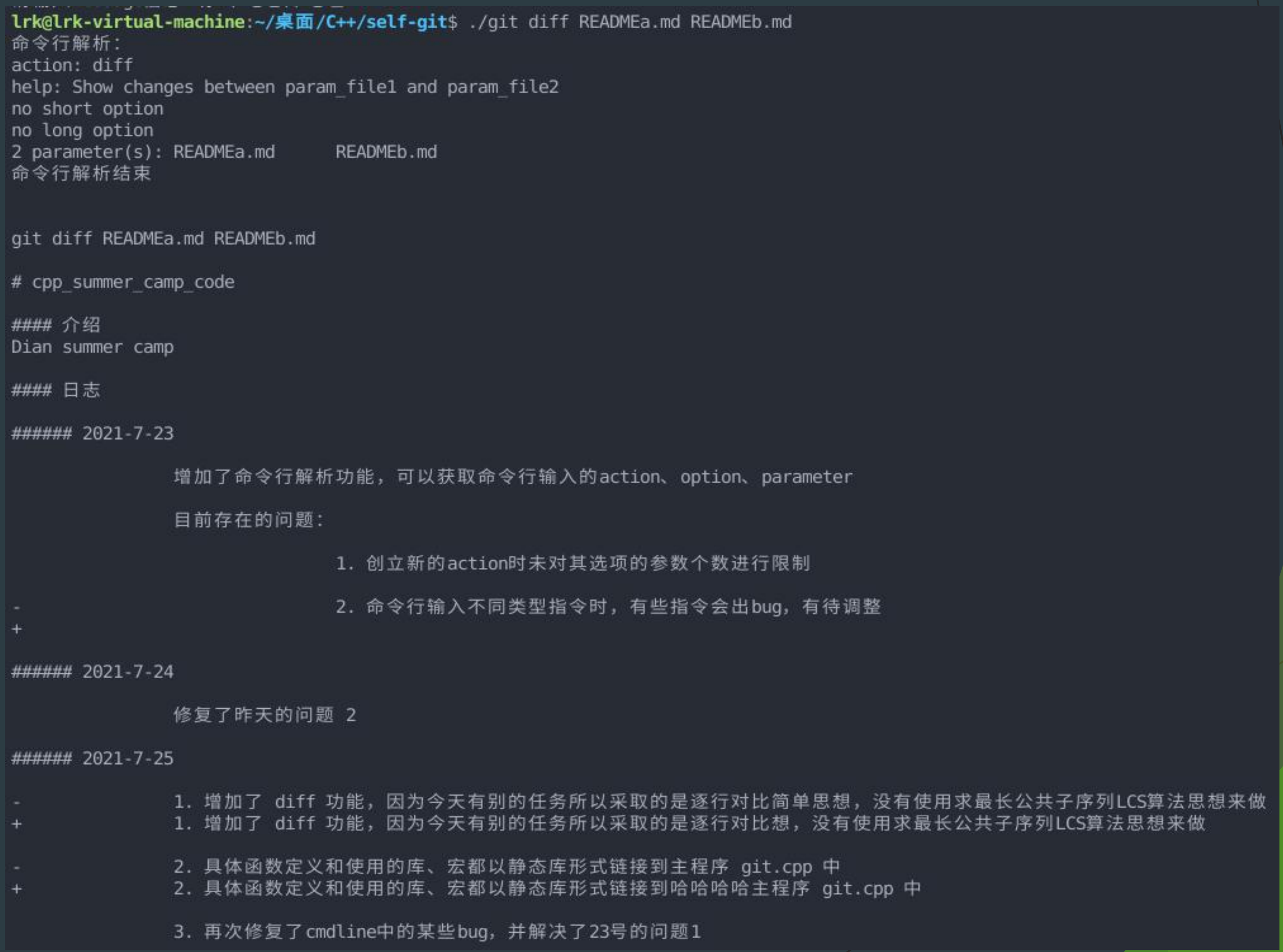
```
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git commit -m just-for-test
命令行解析：
action: commit
help: commit the files from stage dir to local dir with the parameter as message
short option: m
no long option
1 parameter(s): just-for-test
命令行解析结束
```

# Diff

预期效果：工作目录与暂存区下同名文件的对比

思路：1. 逐行比较
　　　2. LCS算法

效果演示：

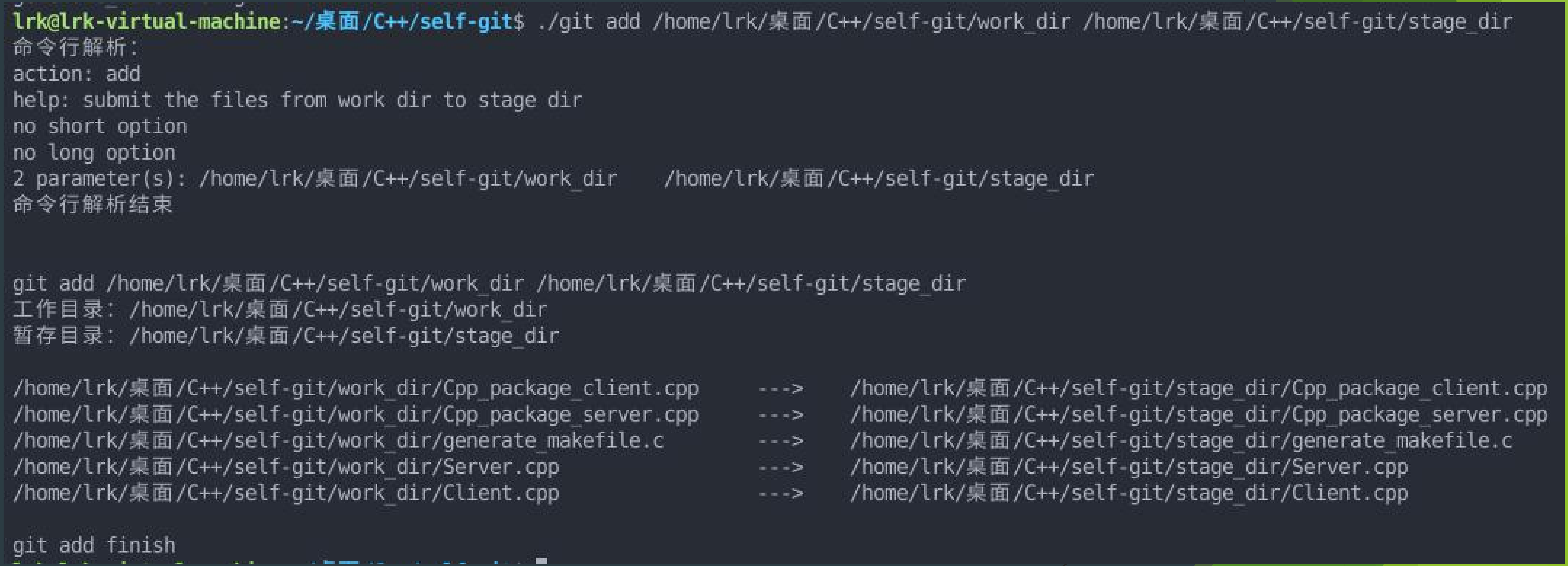


```
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git diff READMEa.md READMEb.md
命令行解析:
action: diff
help: Show changes between param_file1 and param_file2
no short option
no long option
2 parameter(s): READMEa.md      READMEb.md
命令行解析结束


git diff READMEa.md READMEb.md

# cpp_summer_camp_code

#### 介绍
Dian summer camp

#### 日志

###### 2021-7-23

                增加了命令行解析功能，可以获取命令行输入的action、option、parameter

                目前存在的问题:

                                1. 创立新的action时未对其选项的参数个数进行限制

-                               2. 命令行输入不同类型指令时，有些指令会出bug，有待调整
+

###### 2021-7-24

                修复了昨天的问题 2

###### 2021-7-25

-               1. 增加了 diff 功能，因为今天有别的任务所以采取的是逐行对比简单思想，没有使用求最长公共子序列LCS算法思想来做
+               1. 增加了 diff 功能，因为今天有别的任务所以采取的是逐行对比想，没有使用求最长公共子序列LCS算法思想来做

-               2. 具体函数定义和使用的库、宏都以静态库形式链接到主程序 git.cpp 中
+               2. 具体函数定义和使用的库、宏都以静态库形式链接到哈哈哈哈主程序 git.cpp 中

                3. 再次修复了cmdline中的某些bug，并解决了23号的问题1
```

# Add

预期效果：从工作目录向暂存区拷贝文件

效果演示：

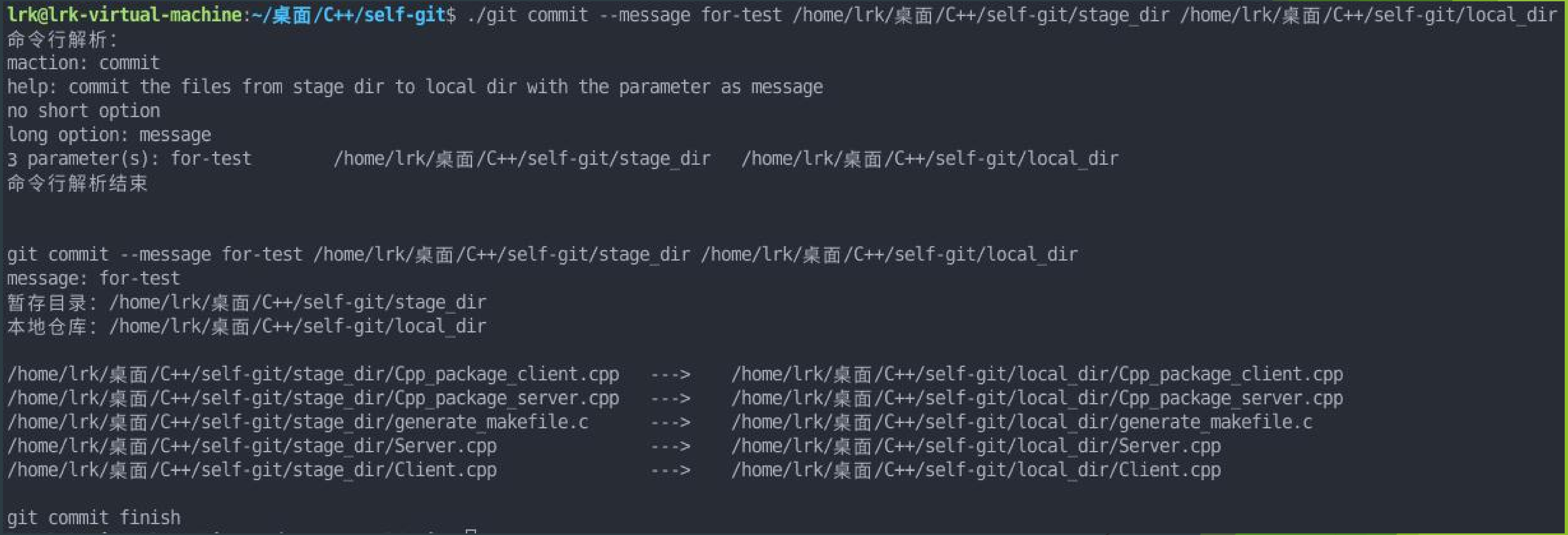


```
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git add /home/lrk/桌面/C++/self-git/work_dir /home/lrk/桌面/C++/self-git/stage_dir
命令行解析:
action: add
help: submit the files from work dir to stage dir
no short option
no long option
2 parameter(s): /home/lrk/桌面/C++/self-git/work_dir    /home/lrk/桌面/C++/self-git/stage_dir
命令行解析结束



git add /home/lrk/桌面/C++/self-git/work_dir /home/lrk/桌面/C++/self-git/stage_dir
工作目录: /home/lrk/桌面/C++/self-git/work_dir
暂存目录: /home/lrk/桌面/C++/self-git/stage_dir

/home/lrk/桌面/C++/self-git/work_dir/Cpp_package_client.cpp     --->    /home/lrk/桌面/C++/self-git/stage_dir/Cpp_package_client.cpp
/home/lrk/桌面/C++/self-git/work_dir/Cpp_package_server.cpp     --->    /home/lrk/桌面/C++/self-git/stage_dir/Cpp_package_server.cpp
/home/lrk/桌面/C++/self-git/work_dir/generate_makefile.c        --->    /home/lrk/桌面/C++/self-git/stage_dir/generate_makefile.c
/home/lrk/桌面/C++/self-git/work_dir/Server.cpp                 --->    /home/lrk/桌面/C++/self-git/stage_dir/Server.cpp
/home/lrk/桌面/C++/self-git/work_dir/Client.cpp                 --->    /home/lrk/桌面/C++/self-git/stage_dir/Client.cpp

git add finish
```

# Commit

预期效果：把暂存区文件提交到本地仓库内一个以当前时间命名的文件夹下

效果演示：

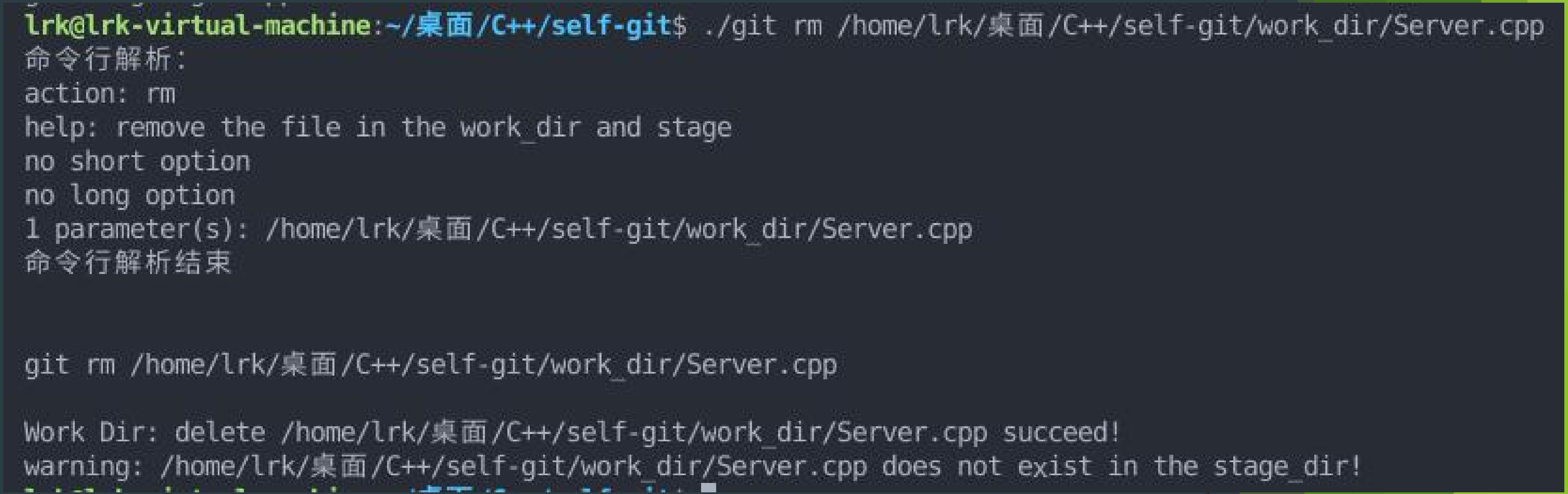


```
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git commit --message for-test /home/lrk/桌面/C++/self-git/stage_dir /home/lrk/桌面/C++/self-git/local_dir
命令行解析:
maction: commit
help: commit the files from stage dir to local dir with the parameter as message
no short option
long option: message
3 parameter(s): for-test        /home/lrk/桌面/C++/self-git/stage_dir   /home/lrk/桌面/C++/self-git/local_dir
命令行解析结束


git commit --message for-test /home/lrk/桌面/C++/self-git/stage_dir /home/lrk/桌面/C++/self-git/local_dir
message: for-test
暂存目录: /home/lrk/桌面/C++/self-git/stage_dir
本地仓库: /home/lrk/桌面/C++/self-git/local_dir

/home/lrk/桌面/C++/self-git/stage_dir/Cpp_package_client.cpp   --->    /home/lrk/桌面/C++/self-git/local_dir/Cpp_package_client.cpp
/home/lrk/桌面/C++/self-git/stage_dir/Cpp_package_server.cpp   --->    /home/lrk/桌面/C++/self-git/local_dir/Cpp_package_server.cpp
/home/lrk/桌面/C++/self-git/stage_dir/generate_makefile.c      --->    /home/lrk/桌面/C++/self-git/local_dir/generate_makefile.c
/home/lrk/桌面/C++/self-git/stage_dir/Server.cpp               --->    /home/lrk/桌面/C++/self-git/local_dir/Server.cpp
/home/lrk/桌面/C++/self-git/stage_dir/Client.cpp               --->    /home/lrk/桌面/C++/self-git/local_dir/Client.cpp

git commit finish
```

# Rm

预期效果：删除暂工作目录和指定文件并同步到暂存区

效果演示：

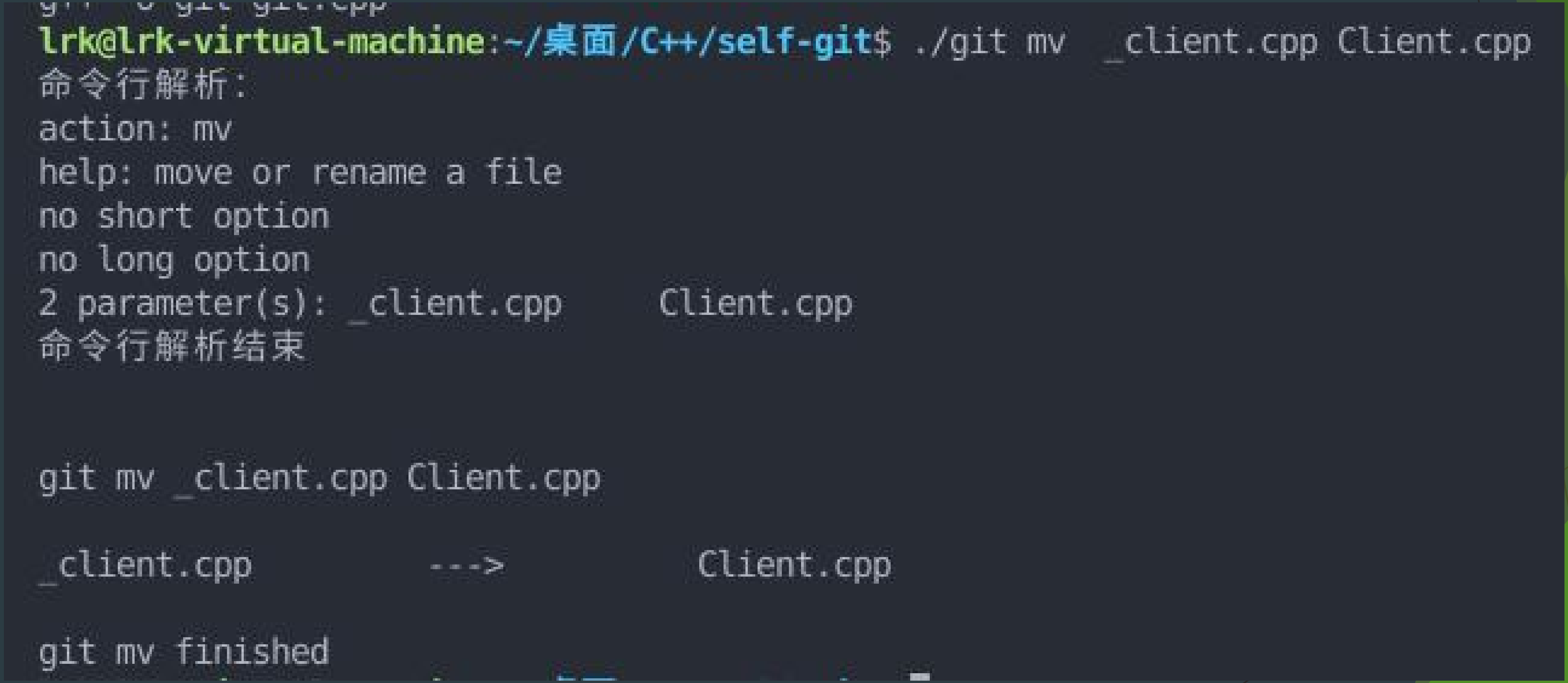


```
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git rm /home/lrk/桌面/C++/self-git/work_dir/Server.cpp
命令行解析：
action: rm
help: remove the file in the work_dir and stage
no short option
no long option
1 parameter(s): /home/lrk/桌面/C++/self-git/work_dir/Server.cpp
命令行解析结束



git rm /home/lrk/桌面/C++/self-git/work_dir/Server.cpp

Work Dir: delete /home/lrk/桌面/C++/self-git/work_dir/Server.cpp succeed!
warning: /home/lrk/桌面/C++/self-git/work_dir/Server.cpp does not exist in the stage_dir!
```

# Mv

预期效果：移动或重命名暂工作目录的指定文件并同步到暂存区

效果演示：



```
g++ -o git git.cpp
lrk@lrk-virtual-machine:~/桌面/C++/self-git$ ./git mv  _client.cpp Client.cpp
命令行解析:
action: mv
help: move or rename a file
no short option
no long option
2 parameter(s): _client.cpp     Client.cpp
命令行解析结束


git mv _client.cpp Client.cpp

_client.cpp          --->          Client.cpp

git mv finished
```

## Init

预期效果：初始化指定文件夹为工作目录，并生成.git文件夹。其中包含暂存区stage、本地仓库local

## Status

预期效果：提示工作目录下新增的文件和修改的文件

## Reset

预期效果：--workdir 用暂存区内容覆盖工作目录内容
--cathe 用本地仓库内容覆盖暂存区内容