



“软件工程(C编码实践篇)”实验报告

实验一：写一个hello world小程序

SA17225549 卓钺澳

通过Linux命令行进入Code文件夹，并执行git clone

```
shiyanolou:~/ $ cd Code [19:57:58]
shiyanolou:Code/ $ git clone https://github.com/SA17225549/-549cc-.git
正克隆到 '-549cc-'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
检查连接... 完成。
shiyanolou:Code/ $ cd -549cc- [20:00:50]
```

进入实验目录，建立lab1文件夹，同时使用vi编辑hello.c文件

```
shiyanolou:-549cc-/ (master) $ mkdir lab1 [20:01:08]
shiyanolou:-549cc-/ (master) $ cd lab1 [20:01:22]
shiyanolou:lab1/ (master) $ vi hello.c [20:01:28]
shiyanolou:lab1/ (master*) $ ls [20:03:15]
hello.c
```

输入编写hello world程序

```
#include<stdio.h>
int main(){
printf("hello world!\n");
}
~
~
~
```

输入esc，：wq退出并保存

```
~
~
~
~
~
~
:wq
```

将hello.c文件编译为可执行文件hello，并执行输出hello world，提交到git

```
shiyanolou:lab1/ (master*) $ gcc -o hello hello.c [20:03:27]
shiyanolou:lab1/ (master*) $ ls [20:03:40]
hello hello.c
shiyanolou:lab1/ (master*) $ ./ hello [20:03:41]
zsh: 权限不够: ./
shiyanolou:lab1/ (master*) $ ./hello [20:03:46]
hello world!
shiyanolou:lab1/ (master*) $ git add hello.c [20:03:52]
shiyanolou:lab1/ (master*) $ git commit -m "hello world" [20:04:11]
[master 37d0889] hello world
1 file changed, 4 insertions(+)
create mode 100644 lab1/hello.c
```

```
shiyanolou:lab1/ (master*) $ git push [20:04:50]
warning: push.default 尚未设置，它的默认值在 Git 2.0 从 'matching'
变更为 'simple'。若要不再显示本信息并保持传统习惯，进行如下设置：
```

```
git config --global push.default matching
```

若要不再显示本信息并从现在开始采用新的使用习惯，设置：

```
git config --global push.default simple
```

当 push.default 设置为 'matching' 后，git 将推送和远程同名的所有本地分支。

从 Git 2.0 开始，Git 缺省采用更为保守的 'simple' 模式，只推送当前分支到远程关联的同名分支，即 'git push' 推送当前分支。

参见 'git help config' 并查找 'push.default' 以获取更多信息。

('simple' 模式由 Git 1.7.11 版本引入。如果您有时要使用老版本的 Git，为保持兼容，请用 'current' 代替 'simple')

```
Username for 'https://github.com': SA17225549
Password for 'https://SA17225549@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 367 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/SA17225549/-549cc-.git
cb3ff73..37d0889 master -> master
```

