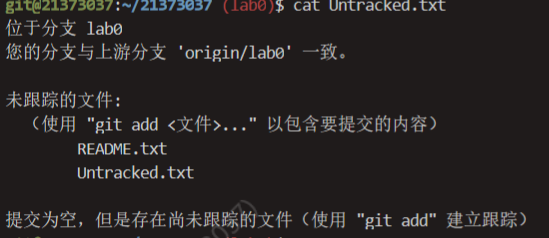
**操作系统LAB0实验报告**

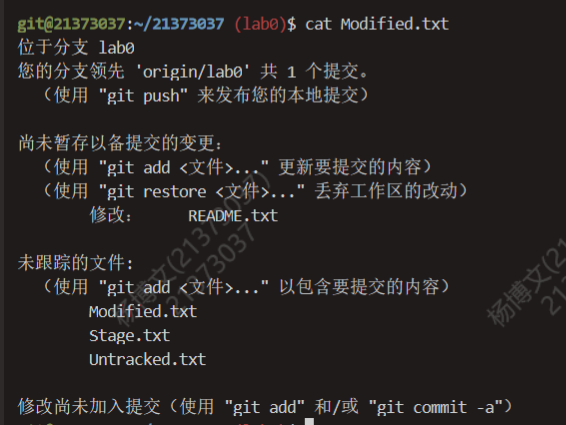
**报告人：21373037 杨博文**

1. 思考题

1.Untracked， Staged，Modified如图所示

****

****

****

README.txt文件在未add前，处于未跟踪的状态，在add后，处于暂存区中，即处在staged暂存状态，而在对该文件修改后且未再次add时，该文件处于已修改的状态。一旦初次跟踪该文件（add）后，除非删除该文件，否则将不会再处于未跟踪的状态，若对其进行修改，即使仍需再次add,那也是从已修改状态转移到暂存状态。

2. add the file: git add

stage the file: git add

commit the file: git commit -m

3. (1)git checkout --print.c

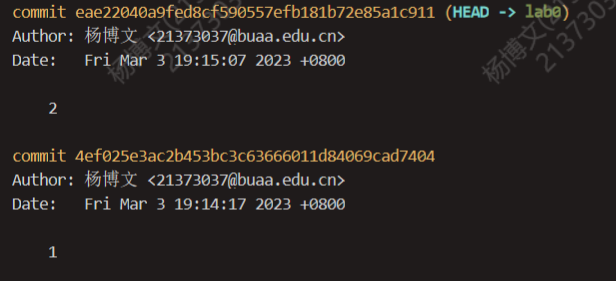
(2)git checkout HEAD print.c

(3)git rm --cached hello.txt

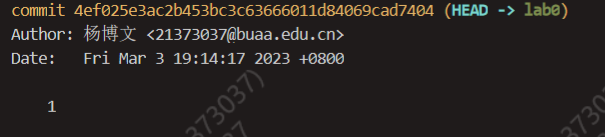
4.提交三次后版本信息如下图所示：



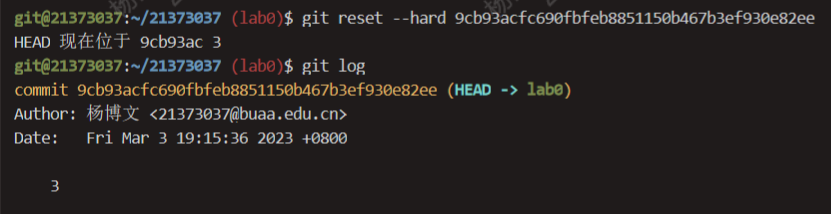
进行第一次HEAD回退后如下图所示，可见往后回退一个版本到2。



进行一次说明为1的哈希值回退后如下图所示，可见往后回退到版本1处。



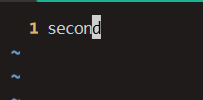
进行一次说明为3的哈希值回退后如下图所示，可见恢复到了版本3处



1. 执行第一条语句后终端回显first



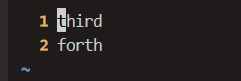
执行第二条语句后打开output.txt，其中内容为second



执行第三条语句后output.txt如下图，可见原内容被覆盖掉了。



执行第四条语句后output.txt如下图，可见新输出的文本追加在了原内容的后面。



1. echo echo Start Shell输出 echo Start Shell

echo ‘echo Start Shell输出 Start Shell

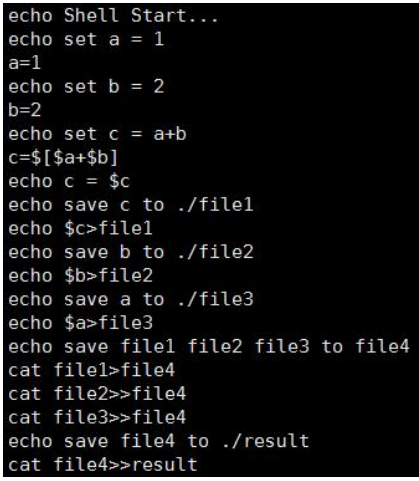
echo echo $c>file在file中输出echo

Echo ‘echo $c>file`在file中无输出

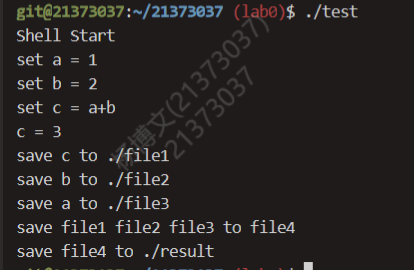
为得到目标文本，command如图



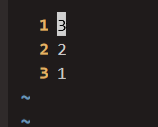
Test文件如题目要求所示：



将test赋予权限，作为批处理文件运行后，终端上显示：



Result文件中内容为：



为创建test而在command脚本的输入中，用双引号将想要输出的文字括起来即可，注意面对$时应使用\转义。而执行test文件时，c,b,a三个参数所对应的值被存入不同的file文件中，最后再将三个文件中的内容依次重定向到result文件中，最后结果即为上图，输出3，2，1。

1. 难点分析

本次lab0考察的是对于Linux命令行，Makefile，Bash脚本的基本操作编写方法，本身不算太难，其中我遇到了三个难点。

第一个难点是Makefile的嵌套调用**$(MAKE) -C <子目录>**，意思为调用子目录的makefile，实现类似于函数调用的机制。第二个难点在于两个object目标文件的链接，我们知道预处理只是加入了头文件的引用，然后并没有把实际内容放进来，这一步在链接操作中完成，可使用gcc <目标文件> <目标文件>......的方式来得到最后可执行代码。在预处理时，用到的头文件会进入系统自带的库中寻找，若该头文件是自己编写的该如何定位寻找呢，我们可以使用gcc -I <文件路径>的方式定位到手写的头文件。最后是对于各种参数的混淆与记忆不清，比如sed的各种使用方法，经常会使用错误或记不清，需要再次巩固记忆。

1. 实验体会

本次LAB0实验我总共用时三小时左右，实验本身并无很硬核的难点，但由于我没有观看教学视频，导致很多东西都是在网上现学现卖，后来才得知视频中涵盖了我大部分的困惑点，以后要注意多看教程，这样才能少走弯路。同时我深感各种指令(尤其是文本指令)的繁琐，想必记忆并熟练运用好需要一点功夫，争取在闭卷的LAB0课下考试中取得优异成绩。

我的实验报告到此结束，谢谢。