

# Lab0实验报告

## 一、实验思考题

### Thinking 0.1

不一样。第一次add前, README.txt属于untracked files, 现在是changes not staged for commit。

### Thinking 0.2

add the file / stage the file: `git add ...`

commit: `git commit ...`

### Thinking 0.3

(1) `git restore printf.c` 或 `git checkout -- printf.c`

(2) `git reset HEAD printf.c` + `git restore printf.c`

**[建议1]**

**经查找, 指导书前面未讲git rm file指令, 只介绍了git rm --cached, 这题要看懂就得去额外检索。建议加入对 rm、git rm、git rm -f、git rm --cached的区别和介绍**

(3) `git rm --cached printf.c`

### Thinking 0.4

Hashcode/commit\_id会记录版本库的历史, 根据这个值可以回到任意版本的状态 (包括未来, 即误回退的版本); 若丢失Hashcode, 可以通过 `git reflog` 查看命令历史以获取相应版本Hashcode。

### Thinking 0.5

1.对; 2.对; 3.错; 4.对

和印象中个人经验相符

### Thinking 0.6

1、输出"first"; 2、产生output.txt文件内容为second; 3、output.txt文件内容变为third; 4、output.txt的third下方追加了forth

### Thinking 0.7

```
#!/bin/bash
# command文件
echo echo Shell Start... > test
echo echo set a = 1 >> test
echo a=1 >> test

cat >> ./test << EOF
echo set b = 2
b=2
echo set c = a+b
c=${a+$b}
echo c = $c
```

EOF

```
var=(c b a)
order=(1 2 3)
for ((i=0;i<3;i++))
do
    echo echo save ${var[$i]} to ./file${order[$i]} >> test
    echo echo \${var[$i]}\>file${order[$i]} >> test
done

cat >> ./test << EOF
echo save file1 file2 file3 to file4
cat file1>file4
cat file2>>file4
cat file3>>file4
echo save file4 to ./result
cat file4>>result
EOF
```

result文件如下

```
3
2
1
```

**解释：**根据result文件，我们先定义变量a=1、b=2、c=a+b=3；然后将c、b、a分别存入file1、file2、file3中；之后依次将file1写入file4（cat + 重定向）、file2、file3（cat+重定向追加）；最后将file4加入result：所以result如上所示。

**思考：**第一个没区别；第二个有区别，因为单引号字符串内均原样输出，不会解释变量

#### [建议2]

**感觉shell脚本的讲解略显凌乱，看完写不了什么。**

**希望像菜鸟教程简明地一并讲变量、数组、运算符、流程控制、常用指令等，利于快速上手；但菜鸟也不太齐全，比如for循环的两种形式**

## 二、实验难点

本次实验难点来源于各种繁杂知识

**Exercise 0.1 - 3：** sed p命令使用，'4p;8p;32p'用分号拼接可以打印多行

**Exercise 0.3：** grep使用，awk使用，管道连接

**Exercise 0.4：** 双层Makefile使用：调用与配合（虽然用外层即可）

## 三、体会与感想

本次实验本身不难，但由于东西多而杂，反复搜寻学习Makefile、Shell、git、sed/grep/awk/...等相关基础内容还是花了较长的时间。

## 四、指导书建议

希望教程能更简明，集成度更高，能对重要指令/文件/方法能有更充分的介绍

1-2见前

3. P43 第二行说“重定向在前面已经有过介绍”，但经检索似乎在指导书前面没有介绍