

# OS —— Lab0实验报告

---

22373407 王飞阳

## 一、思考题

---

### 1. Git的使用1

在执行完 git add 命令之前, README.txt 文件是 Untracked 状态, 执行完 git add 命令之后, README.txt 文件是 Staged 状态

状态不一样, 第一次的 status 是未跟踪状态 (untracked), README.txt 文件并未被跟踪; 而最后的 status 是已修改状态 (modified), README.txt 文件被修改但还没有暂存 (modified but not staged)

### 2. 箭头与命令

add the file 对应 git add 命令

stage the file 对应 git add 命令

commit 对应 git commit 命令

### 3. Git 的一些场景

(1) git restore print.c

(2) git reset HEAD print.c

git restore print.c

(3) git reset HEAD hello.txt

### 4. Git 的使用 2

在提交三次之后, 使用命令 git log, 得到以下结果

```
commit d0fab341f269feff681927d33393a65326c073f5 (HEAD -> master)
Author: 王飞阳 <22373407@buaa.edu.cn>
Date: Thu Mar 7 16:16:08 2024 +0800

    3

commit fcd7155eababdf2f04d530a63c5b6d949f37057a
Author: 王飞阳 <22373407@buaa.edu.cn>
Date: Thu Mar 7 16:15:04 2024 +0800

    2

commit 305ed58d7a0049f5600076add71b2a6f13dcba7b
Author: 王飞阳 <22373407@buaa.edu.cn>
Date: Thu Mar 7 16:14:12 2024 +0800

    1
```

在使用 `git reset --hard HEAD^` 回退后，test3被撤回，回退到上一个版本

在使用 `git reset --hard hash (test1)` 后，test2被撤回，回退到第一个版本

在使用 `git reset --hard hash (test2)` 后，再次回到第二个版本

## 5. 思考-echo 的使用

在执行完题目中的4条命令后，输出分别如下：

- (1) 标准输出中输出 first
- (2) output.txt 中写入 second
- (3) output.txt 中内容被覆盖，只剩下third
- (4) output.txt 中在文末接着写入 forth

## 6. 文件的操作

command 文件

```

1 #!/bin/bash
2 echo 'echo Shell Start...' > test
3 echo 'echo set a = 1' >> test
4 echo 'a=1' >> test
5 echo 'echo set b = 2' >> test
6 echo 'b=2' >> test
7 echo 'echo set c = a+b' >> test
8 echo 'c=${a+$b}' >> test
9 echo 'echo c = $c' >> test
10 echo 'echo save c to ./file1' >> test
11 echo 'echo $c>file1' >> test
12 echo 'echo save b to ./file2' >> test
13 echo 'echo $b>file2' >> test
14 echo 'echo save a to ./file3' >> test
15 echo 'echo $a>file3' >> test
16 echo 'echo save file1 file2 file3 to file4' >> test
17 echo 'cat file1>file4' >> test
18 echo 'cat file2>>file4' >> test
19 echo 'cat file3>>file4' >> test
20 echo 'echo save file4 to ./result' >> test
21 echo 'cat file4>>result' >> test

```

result 文件

```

1 3
2 2
3 1

```

结果说明：在 test 文件中除去直接 echo 命令打印的部分，命令中首先将a赋值为1，b赋值为2，再将c赋值为a+b的和，即为3；紧接着，命令将啊c，b，a的值分别输出重定向到file1，file2，file3文件中；最后又将file1，file2，file3文件中的内容拼接到file4中，最后将file4复制到result文件中，得到最终结果3，2，1。

思考问题：

(1) echo echo Shell Start 与 echo `echo Shell Start` 效果是有区别。echo echo Shell Start 这个命令会输出字符串 "echo Shell Start" 到终端或标准输出,它会将 "echo Shell Start" 视为纯文本并直接输出; echo `echo Shell Start` 这个命令使用了反引号来执行内部的命令 "echo Shell Start", 然后将其输出作为参数传递给外部的 echo 命令。在这种情况下，内部的命令会被执行，其输出 ("Shell Start") 会作为字符串传递给外部 echo 命令。最终，外部的 echo 命令会输出 "Shell Start" 到终端或标准输出。

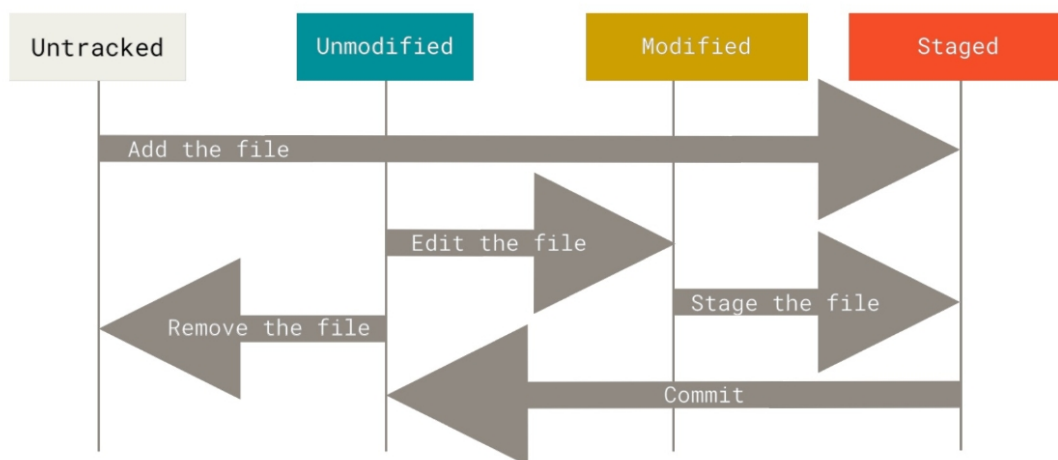
(2) echo echo c>file1 与 `echo echo c>file1` 效果是有区别。echo echo c>file1 这个命令会将字符串 "echo c" 写入文件 "file1" 中。它直接将 "echo c" 视为文本并写入文件，不进行命令替换; `echo echo c>file1` 这个命令使用了反引号来执行内部的命令 "echo c>file1", 然后将其输出作为参数传递给外部的 echo 命令。在这种情况下，内部的命令 "echo c>file1" 会将字符串 "c" 写入文件 "file1" 中。最

终，外部的 `echo` 命令不会执行任何命令替换，而是将其作为纯文本输出。

## 二、难点分析

经过假期预习以及对 Lab0 实验的完成，个人认为本实验的难点如下

- Linux 各种命令行的使用。在 Linux 操作系统中，我们需要使用一行一行命令向操作系统发出指示。但由于命令数量较多，单个命令的用法也有很多不同，导致我们在书写命令行之时，很容易遗忘每个命令具体的用法以及注意事项。[Linux常用命令大全](#)
- 各种工具的使用。在Linux操作系统中，存在很多工具供我们使用，包括 `gcc`, `make`, `ctags`, `tmux`, (`grep`、`sed`、`awk` 文本处理三剑客) 等工具，每种工具都有自己的对应使用方法，我们熟悉并掌握需要花费一定的时间。[gcc](#) [make](#) [ctags](#) [tmux](#) [grep](#) [sed](#) [awk](#)
- Git的学习认识。Git 作为一种版本控制系统，可用于实现多人合作开发；我们在提交实验时，也要使用 Git；因此，对 Git 的理解很重要。[Git的相关知识](#)。Git中文件的储存状态可以用如下图表示：



- Shell 脚本编程。Shell 脚本编程可以实现自动化操作和精准操作；作为一门新的语言，对Bash Shell 语法的了解与熟练使用是本实验的难点之一。[shell编程](#)

## 三、实验体会

本次实验内容相对容易，主要是让我们熟悉认识操作系统实验环境，掌握操作系统实验所需的基本工具并能够脱离可视化界面进行工作。在实验过程中，主要的难点在于对 Linux 命令的熟练理解和掌握，对操作系统实验环境的初步认识。实验过程中，没有遇到明显困难的问题，有困难的地方也能通过查询相关资料，询问助教等方式解决。本次实验，让我收获了许多关于 Linux 命令行的相关知识，对一些实用工具有了一定的掌握，期待下次实验的进行。