

Lab 0 实验报告

一、实验思考题

思考题 0.1

通过你的使用经验，简单分析 CLI Shell，GUI Shell 在你使用过程中的各自优劣（100 字以内）

CLI Shell：优势：节省资源，速度快，效率高，控制全面；劣势：上手需要一定时间的学习，界面不直观，视觉效果简陋，容错率低

GUI Shell：优势：视觉效果好，学习难度低，易于上手；劣势：占用资源多，操作效率低，开启不常用功能时消耗时间长

思考题 0.2

使用你知道的方法（包括重定向）创建下图内容的文件（文件命名为test），将创建该文件的命令序列保存在command文件中，并将test文件作为批处理文件运行，将运行结果输出至result文件中。给出command文件和result文件的内容，并对最后的结果进行解释说明（可以从test文件的内容入手）

具体实现的过程中思考下列问题：

echo echo Shell Start 与 echo 'echo Shell Start'效果是否有区别

echo echo \>file1 与 echo 'echo \>file1' 效果是否有区别

command文件内容：

```
echo 'echo Shell Start' >>test
echo 'echo set a = 1' >>test
echo 'a=1' >>test
echo 'echo set b = 2' >>test
echo 'b=2' >>test
echo 'echo set c = a + b' >>test
echo 'c=${a} + ${b}' >>test
echo 'echo c = $c' >>test
echo 'echo save c to ./file1' >>test
echo 'echo $c >file1' >>test
echo 'echo save b to ./file2' >>test
echo 'echo $b>file2' >>test
echo 'echo save a to ./file3' >>test
echo 'echo $a>file3' >>test
echo 'echo save file1 file2 file3 to file4' >>test
echo 'cat file1>file4' >>test
echo 'cat file2>>file4' >>test
echo 'cat file3>>file4' >>test
echo 'echo save file4 to ./result' >>test
echo 'cat file4>>result' >>test
```

运行结果

```
Shell Start
set a = 1
set b = 2
set c = a + b
c = 3
save c to ./file1
save b to ./file2
save a to ./file3
save file1 file2 file3 to file4
save file4 to ./result
```

result文件内容：

```
3
2
1
```

`echo echo Shell Start` 与 `echo 'echo Shell Start'` 效果是否有区别：

无区别，但是值得注意的是前一条指令中空格的多少不影响最终输出，后一条影响

`echo echo \${c}>file1` 与 `echo 'echo \${c}>file1'` 效果是否有区别：

有区别，前一条指令在控制台不输出，file1中内容为 `echo ${c}`；后一条指令在控制台中输出 `echo \${c}>file1`

思考题 0.3

仔细看看这张图，思考一下箭头中的 `add the file`、`stage the file` 和 `commit` 分别对应的是 Git 里的哪些命令呢？

- `add the file` 对应 `git add`
- `stage the file` 对应 `git add`
- `commit` 对应 `git commit`

思考题 0.4

- 深夜，小明在做操作系统实验。困意一阵阵袭来，小明睡倒在了键盘上。等到小明早上醒来的时候，他惊恐地发现，他把一个重要的代码文件 `printf.c` 删除掉了。苦恼的小明向你求助，你该怎样帮他恢复代码文件呢？

- 正在小明苦恼的时候，小红主动请缨帮小明解决问题。小红很爽快地在键盘上敲下了 `git rm printf.c`，这下事情更复杂了，现在你又该如何处理才能弥补小红的过错呢？

- 处理完代码文件，你正打算去找小明说他的文件已经恢复了，但突然发现小明的仓库里有一个叫 `Tucao.txt`，你好奇地打开一看，发现是吐槽操作系统实验的，且该文件已经被添加到暂存区了，面对这样的情况，你该如何设置才能使 `Tucao.txt` 在不从工作区删除的情况下不会被 `git commit` 指令提交到版本库？

1. `git checkout -- printf.c`
 2. 先执行 `git reset HEAD printf.c` 然后执行 `git checkout -- printf.c`
 3. `git rm --cached Tucao.txt`
-

思考题 0.5

思考下面四个描述，你觉得哪些正确，哪些错误，请给出你参考的资料或实验证据。

1. 克隆时所有分支均被克隆，但只有HEAD指向的分支被检出。
2. 克隆出的工作区中执行 `git log`、`git status`、`git checkout`、`git commit`等操作不会去访问远程版本库。
3. 克隆时只有远程版本库HEAD指向的分支被克隆。
4. 克隆后工作区的默认分支处于master分支。

1. 正确
2. 正确
3. 错误，所有分支均被克隆
4. 错误，默认处于HEAD指向的分支，不一定是master

依据

*Manual page git-clone(1)*中指出

```
Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository (visible using git branch --remotes), and creates and checks out an initial branch that is forked from the cloned repository's currently active branch.
```

且在试验机上测试，checkout非master时并未从远程版本库下载

此外，下面的文章介绍了clone后默认分支的情况以及如何修改HEAD（需要服务器访问权限）

<https://stackoverflow.com/questions/1485578/change-a-git-remote-head-to-point-to-something-besides-master>

<https://stackoverflow.com/questions/3301956/git-correct-way-to-change-active-branch-in-a-bare-repository/3302018#3302018>

So to wrap that up, you have repo A and clone it:

```
HEAD references refs/heads/master and that exists
-> you get a local branch called master, starting from origin/master
```

```
HEAD references refs/heads/anotherBranch and that exists
-> you get a local branch called anotherBranch, starting from
origin/anotherBranch
```

```
HEAD references refs/heads/master and that doesn't exist
-> "git clone" complains
```

可以使用 `git symbolic-ref HEAD refs/heads/<branch name>` 修改HEAD

二、实验难点图示

1. Makefile的使用

项目	语法
变量定义	<code>var := val</code>
变量使用	<code>\$(var)</code>
伪目标（必定执行）	<code>.PHONY target</code>
目标，依赖，构建行为	<code>target: dependence_1 dependence_2 [换行] action</code>

2. 实验中git的使用

git clone(克隆仓库)

→ 修改、书写

→ git add (添加至暂存区)

→ git commit (commit)

→ git push (提交到remote仓库)

→ git pull (拉取测试结果)

三、体会与感想

- 难度：中等偏下，主要难度在于初次学习Makefile，以及sed、awk等工具，需要学习的内容较多，以及初次使用CLI Shell环境进行实验需要适应。
- 花费时长：环境配置：4 h，Lab 0实验：2 h
- 体会与感想：命令行下操作还是相当困难的，需要进一步学习。Makefile文件用处很大且有很多技巧，包括GCC等软件都有许多很难掌握但很有用的参数，需要多下功夫。