**Lab0实验报告**

19373135 田旗舰

**一、实验思考题**

**思考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：

优点：节约计算机资源，运行效率高。

在熟记命令的前提下操作比GUI快。

功能强大。

稳定性好。

缺点：不方便操作，学习起来比较困难

需要记忆命令

GUI：

优点：方便操作，使用更熟悉。

图形界面美观。

缺点：运行效率比CLI低。

功能不如CLI强大

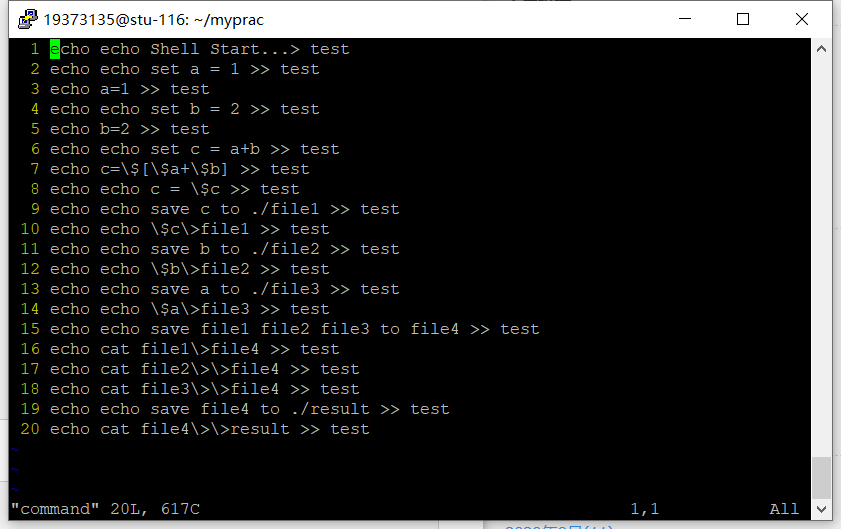
**思考题 0.2  
       使用你知道的方法（包括重定向）创建下图内容的文件（文件命名为test），将创建该文件的命令序列保存在command文件中，并将test文件作为批处理文件运行，将运行结果输出至result文件中。给出command文件和result文件的内容，并对最后的结果进行解释说明（可以从test文件的内容入手）**

**具体实现的过程中思考下列问题：**

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

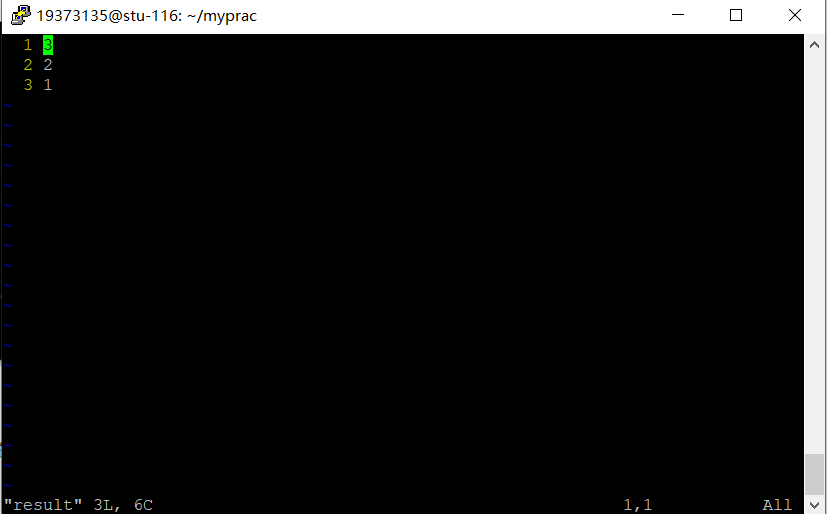
**echo \$c>file1 与 echo '\$c>file1' 效果是否有区别**

command文件内容



执行bash command，bash test

result文件中的结果



test文件的内容是将a b c3个变量分别赋值，并写入到对应文件file3 file2 file1中，再将文件内容写入到file4中。

echo ‘Shell Start’ 与 echo Shell Start 直接执行效果相同，但如果重定向到文件中则写入内容不同，分别为echo ‘Shell Start’和echo Shell Start。

echo \$c>file1将变量c写入file1中，echo '\$c>file1'直接输出 \$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？****

git add git add 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，你该怎样帮他把代码文件恢复呢？****

git checkout printf.c

****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？****

git reset HEAD printf.c

git checkout printf.c

****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？****

git rm --cached printf.c

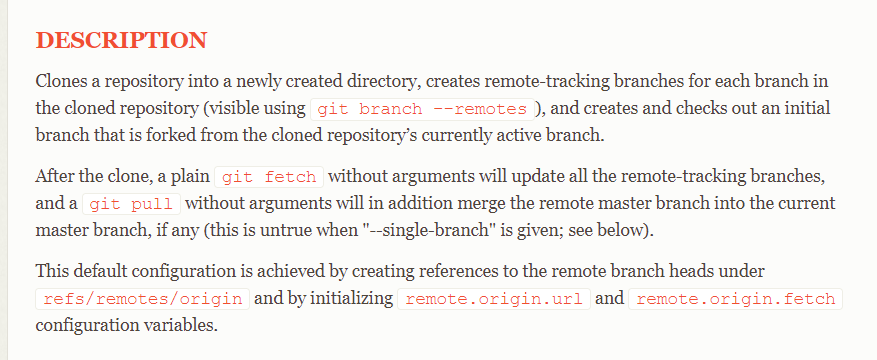
****思考题0.5****

****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。****

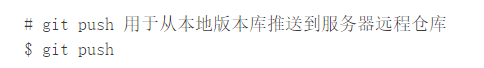
**正确**

**节选自官网**



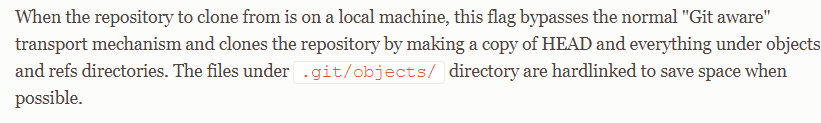
****2.克隆出的工作区中执行 git log、git status、git checkout、git commit等操作不会去访问远程版本库。****

**正确**



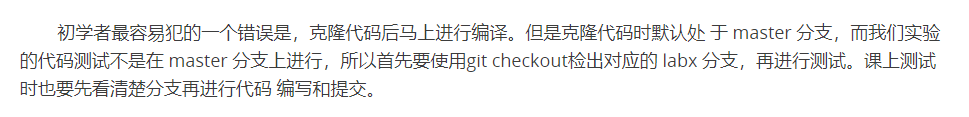
****3.克隆时只有远程版本库HEAD指向的分支被克隆。****

**错误**



****4.克隆后工作区的默认分支处于master分支。****

**正确**

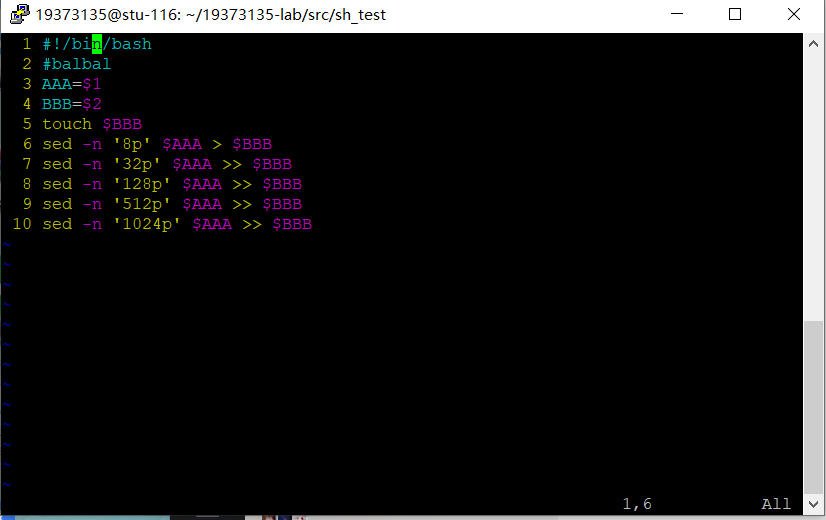


**二、实验难点图示**

**本次实验是为了认识操作系统实验环境并掌握操作系统实验所需的基本工具而进行的铺垫性实验，仅仅是熟悉Linux 操作系统（Ubuntu），了解控制终端，掌握一些常用工具并能够脱离可视化界面进行工作，总体而言难度并不高。**

**但我在上机以及课下的学习中还是遇到了一些困难，主要是由于以前从未接触过Linux操作系统，对于各类工具也不够熟练，基本上所有的内容都是边学边做的状态。**

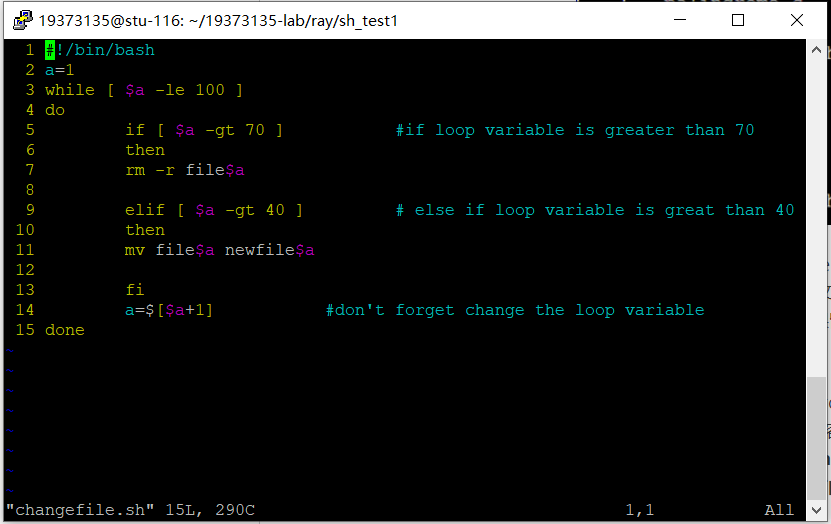
**具体来说实验的难点有：**

****1.shell编程****

对于shell的编程不熟练，对于shell的理解也不太到位。

在task3中需要用到sed和重定向的相关知识，尽管教程中有所讲解，但我在上机时仍然花费了大量的时间，最后又查阅了相关的资料才理解了什么是重定向，尽管现在看了并不是很难，但在上机时仍是一个难点。

此外，在task5中对于shell的流程控制也不够熟练，对于if else 的结构学习了一些时间。

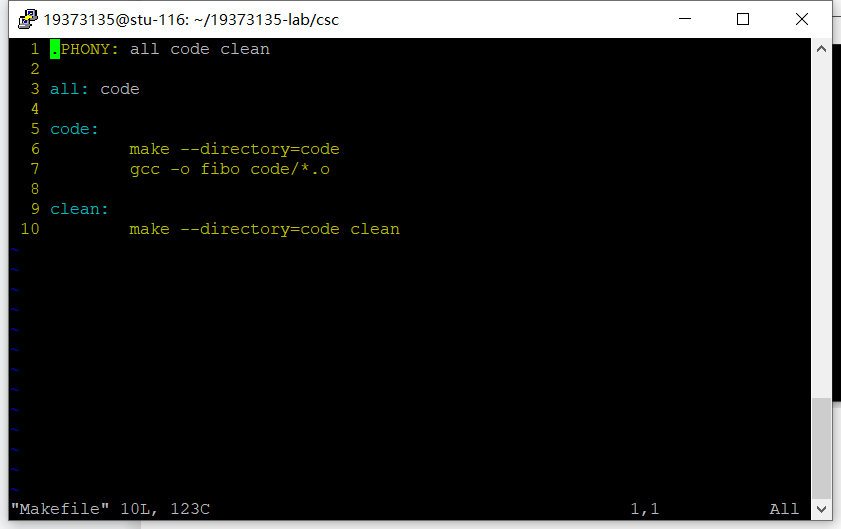


在task7中也涉及到shell编程，对于sed命令用双引号表示传入参数（或者用在单引号的命令中用单引号表示传入参数）也是一个难点。



**2.makefile编写**

在task8中对makefile的编写是我认为lab0中最难的一点



对于.PHONY的理解，利用make --directory=code调用子目录的makefile，以及对于gcc 选项的应用对我来说都是比较难的地方。

**三、体会与感想**

学习新东西的开始总是痛苦的，但熟悉以后就会好很多。

lab0总体难度适中，可能对于熟悉Linux的同学来说较为简单，而对于我这样的新手来说有一定的难度。但同时，对于我的提升也是很大的，很有收获。

我在lab0的学习时间大概为8个小时，包括上机的3小时和课下的5小时，但对我来说还不太够，我会再花费一些时间学习lab0的相关操作，以便在下周的上机中更加熟练。

**四、指导书反馈**

感觉实例少了一些，例如在重定向输入输出与sed等工具的讲解后最好能加上一些实际操作的截图让我们更直观地了解其含义。（当然我理解课程组可能是考虑到希望我们自己实际操作，但我仍觉得这样的方法更有利于新手的学习）

**五、残留难点**

lab0中感觉最难的还是makefile的编写，虽然解决了task8，但还有很多的知识没有掌握，而官方的教程感觉看不太下去，希望能了解一下大家都是怎么从0开始学习makefile的编写的。