

VE482 Lab Report

Lab 3 - Fall 2020

Name: Wu Qinhang

ID: 518370910041

Email: william_wu@sjtu.edu.cn

Table of Contents

- Git in practice
- useful linux commands
- sed/awk
- regular expression

Git in practice

Introduction Sequence

Ramping Up

A mixed bag

Remote

Source code

rsync

diff / patch

Scripting

air quality info

ip address

Git in practice

Git branch game accessed from [this site](#).

Introduction Sequence

```
1 # 1.1 Git Commit
2 git commit
3 git commit
```

```
1 # 1.2 Git Branches: branch early, and branch often
2 git checkout -b bugFix
```

```
1 # 1.3 Merging in Git
2 git checkout -b bugFix
3 git commit
4 git checkout master
5 git commit
6 # Merge branch bugFix into master
7 git merge bugFix
```

```
1 # 1.4 Rebase Introduction: making a nice linear sequence of commits
2 git checkout -b bugFix
3 git commit
4 git checkout master
5 git commit
6 git checkout bugFix
7 git rebase master
```

Ramping Up

```
1 # 2.1 Moving around in Git
2 # detach HEAD from the branch
3 git checkout C4
```

```
1 # 2.2 Relative Refs
2 git checkout bugFix^
```

```
1 # 2.3 ~ operator
2 git checkout HEAD^
3 # move (by force) the branch to ...
4 git branch -f bugFix C0
5 git branch -f master C6
```

```
1 # 2.4 Reversing Changes in Git
2 # reverse the most recent commit on local and pushed
3 git reset HEAD~1
4 git checkout pushed
5 git revert HEAD
```

A mixed bag

```
1 # 4.1 Locally stacked commits
2 git checkout master
3 git cherry-pick bugFix
```

```
1 # 4.4 Git Tag
2 git tag v0 c1
3 git tag v1 C2
4 git checkout C2
```

Remote

```
1 # 6.1 Git Remotes
2 git clone
```

```
1 # 6.2 Git Remote Branches
2 git commit
3 git checkout o/master
4 git commit
```

```
1 # 6.3 Git Fetch
2 git fetch
```

```
1 # 6.4 Git Pull: fetch and merge
2 git pull
```

```
1 # 6.6 Git Push
2 git commit
3 git commit
4 git push
```

The screenshot shows the Git GUI application interface. At the top, there's a window title bar with standard macOS window controls and a 'Select a level' button. Below this are two tabs: 'Main' and 'Remote'. The main content area is divided into several sections, each with a title and a description. The 'Introduction Sequence' section includes a list of levels (1-4) with green checkmarks indicating completion. The 'Ramping Up' section includes a list of levels (1-4) with green checkmarks. The 'Moving Work Around' section includes a list of levels (1-2) with green checkmarks. The 'A Mixed Bag' section includes a list of levels (1-5) with green checkmarks. The 'Advanced Topics' section includes a list of levels (1-3) with green checkmarks. The 'Push & Pull -- Git Remotes!' section includes a list of levels (1-8) with green checkmarks. The 'To Origin And Beyond -- Advanced Git Remotes!' section includes a list of levels (1-8) with green checkmarks.

Select a level

Main Remote

Introduction Sequence

A nicely paced introduction to the majority of git commands

2: Branching in Git

1 2 3 4

Ramping Up

The next serving of 100% git awesomes-ness. Hope you're hungry

3: Relative Refs #2 (~)

1 2 3 4

Moving Work Around

"Git" comfortable with modifying the source tree :P

1: Cherry-pick Intro

1 2

A Mixed Bag

A mixed bag of Git techniques, tricks, and tips

5: Git Describe

1 2 3 4 5

Advanced Topics

For the truly brave!

1: Rebasing over 9000 times

1 2 3

Push & Pull -- Git Remotes!

Time to share your 1's and 0's kids; coding just got social

1: Clone Intro

1 2 3 4 5 6 7 8

To Origin And Beyond -- Advanced Git Remotes!

And you thought being a benevolent dictator would be fun...

1: Push Master!

1 2 3 4 5 6 7 8

Source code

rsync

```
1 # rsync in Minix 3
2 pkgin install rsync
3 man rsync
4 # Create an exact copy of the directory /usr/src into the directory
  /usr/src_orig
5 cp -r /usr/src /usr/src_orig
6 # Create an exact copy of the Minix 3 directory /usr/src_orig into your
  Linux system, using rsync and ssh
7 # note: -e "ssh -p $(PortNumber)" is used to choose an alternative remote
  shell program
8 rsync -avz -e "ssh -p 2222" root@192.168.227.129:/usr/src_orig ~/qhhttp
```

diff/patch

```
1 man diff
2 man patch
3 # create a patch corresponding to my changes
4 diff -Naur /usr/src_orig /usr/src > patch.txt
5 # apply a patch
6 patch -p3 < patch.txt
7 # revert a patch
8 patch -R < patch.txt
```

Scripting

air quality info 1 2 3

```
1 curl "https://aqicn.org/?
  city=Shanghai&widgetscript&size=large&id=52b39d71decf07.20261781" 2>
  /dev/null | sed -rn 's/.*title=\\.*">([[[:digit:]]{1,3})*.hdrt.*10px.*>
  ([-]?[[[:digit:]]{1,5})<.*AQ: \\1 Temp: \\2 °C/p'
```

data set accessed from [this site](#).

ip address

```
1 ifconfig | awk -e '{for(i=1;i<=NF;i++){ if($i=="inet"){print $(i+1)} } }'
```

1. "Regex101 - online regex editor and debugger," @*regex101*, 2020. [Online]. Available: <https://regex101.com/>. [Accessed: 29-Sep-2020]. [↵](#)

2. Pablo, "How to output only captured groups with sed?," *Stack Overflow*, 06-May-2010. [Online]. Available: <https://stackoverflow.com/questions/2777579/how-to-output-only-captured-groups-with-sed/43997253>. [Accessed: 29-Sep-2020]. ↵
3. Operator Precedence of Regular Expression, POSIX. [Online]. Available: https://www.boost.org/doc/libs/1_56_0/libs/regex/doc/html/boost_regex/syntax/basic_extended.html#boost_regex.syntax.basic_extended.operator_precedence. ↵