UM–SJTU Joint Institute

Introduction to Operating Systems

(VE482)

Laboratory Report

Lab 3

Name: Ji Xingyou ID: 515370910197
Date: 28 September 2017

Contents

1 Simple git

- **Search what is *git***

  Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

- **Install a *git* client**

  Done.

- **Search the use of the following *git* commands:**

- help	- branch	- merge	- tag	- commit
- init	- push	- add	- log	- clone
- checkout	- pull	- diff	- fetch	- reset

- help: print the synopsis and a list of the most commonly used commands.

- branch: list, create, or delete branches.

- merge: join two or more development histories together.

- tag: create, list, delete or verify a tag object signed with GPG.

- commit: record changes to the repository.

- init: create an empty Git repository or reinitialize an existing one.

- push: update remote refs along with associated objects. **- add:** add file contents to the index.

- log: show commit logs.

- clone: clone a repository into a new directory.

- checkout: switch branches or restore working tree files.

- pull: fetch from and integrate with another repository or a local branch.

- diff: show changes between commits, commit and working tree, etc.

- **fetch:** download objects and refs from another repository.

- **reset:** reset current HEAD to the specified state.

- **Setup your git repository on the VE482 server.** First, edit the config file.

```
Host ve482
        HostName 202.120.43.199
        Port 2482
        IdentityFile ~/.ssh/id_rsa
```

Then, in the terminal type

```
cd Desktop/VE482/Project1
git init
git add
git commit –m '482'
git remote add ve482 git@ve482:515370910197/p1
git push ve482
```

2 Git game

Done.

3 Working with source code

3.1 The *rsync* command

- **In Minix 3 install the rsync software**

```
pkgin install rsync
```

- **Install rsync on you Linux system**

```
sudo apt install rsync
```

- **Read rsync manpage**

```
man rsync
```

- **Create an exact copy of the directory /usr/src into the directory /usr/src_orig**

```
mkdir src_orig

cp -r src/. src_orig
```

- **If you have altered Minix 3 source code during homework 2 remove your changes from /usr/src orig**

- **Create an exact copy of the Minix 3 directory /usr/src_orig into your Linux system, using rsync and ssh (note that the ssh server must be activated under Linux)**

```
rsync -av root@2222::www /usr/src_orig
```

3.2 The *diff* and *patch* command

- **Read the manpages of diff and patch**

  diff: compare files line by line

```
        diff [OPTION]... FILES
```

  patch: apply a diff file to an original

```
        patch -pnum <[patchfile]
```

- **Using the diff command, create a patch corresponding to your changes in homework 2**

```
        diff -Naur usr/src_orig usr/src > diff_file
```

- **Retrieve your patch on your Linux system**

```
patch -p0 < diff_file
```

- **Apply your patch to the copy of /usr/src orig on your Linux system**

```
cd Desktop/VE482/LAB/Lab3
patch -p0 < diff_file
```

- **Revert the patch**

```
patch -RE -p0 < diff_file
```

3.3 Remarks