Unix and Linux

Github

Git and Github

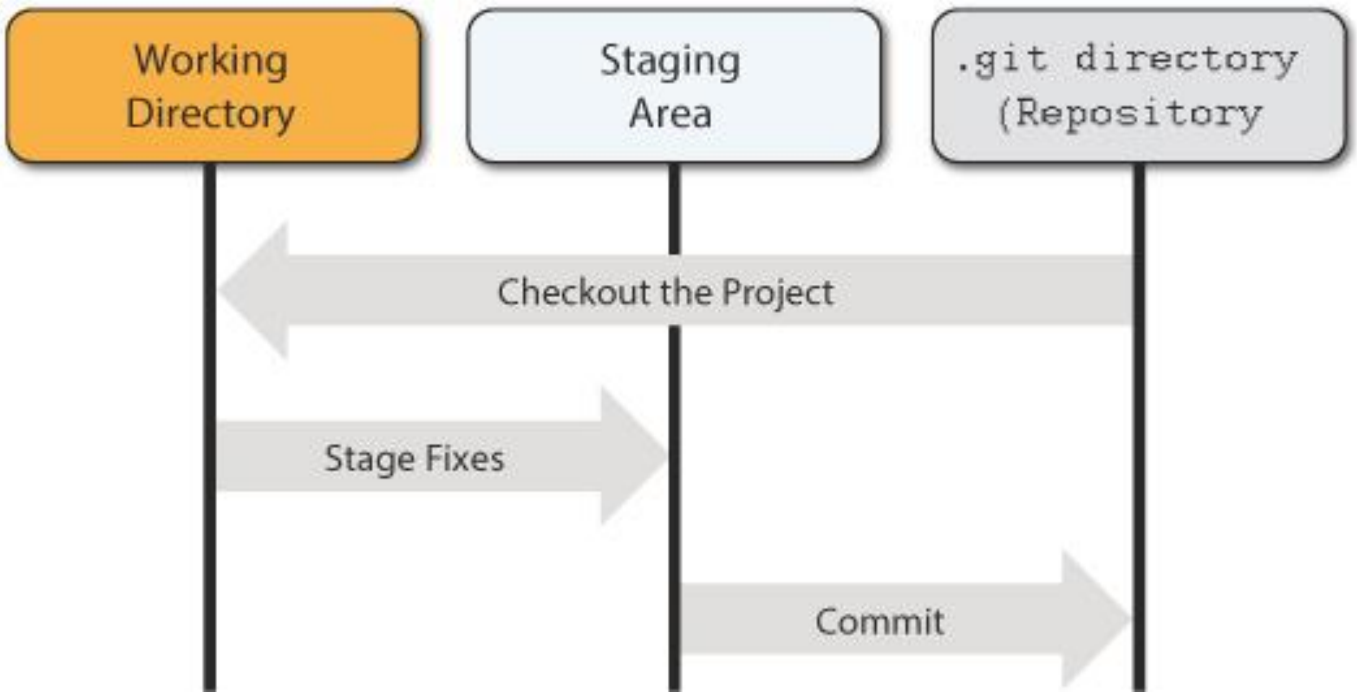
- **Git** is the distributed version control system.
- **Git** is responsible for keeping track of changes to content (usually source code files), and it provides mechanisms for sharing that content with others.
- **GitHub** is a company that provides **Git** repository hosting
- **Git** has been the most widely used version control system since 2014

Git Basics

To get a copy of a project from a Git repository, you use a process called cloning. Cloning doesn’t just create a copy of all the files from the repository; it actually performs three primary functions:

- Creates a local repository of the project under the project_name/.git directory in your home directory. The files of the project in this location are considered to be “checked out” from the central repository.
- Creates a directory where you can directly see the files. This is called the working area. Changes made in the working area are not immediately version controlled.
- Creates a staging area. The staging area is designed to store changes to files before you commit them to the local repository.

Git Stages


Working Directory
Staging Area
.git directory (Repository
Checkout the Project
Stage Fixes
Commit

Git an a Mac

If you have a Mac, you will probably not be able to install the git client from source code, but you probably already have a git client. To check this, type:

```
git –version
```

If you get a response other than “Command not found” then you have a git client installed already.

If you do not have a git client installed, try

```
brew install git
```

Skip the instructions on installing git from source code skip to page 7 and then to page 9.

Install the git client from source code

Create a `~/git295` directory where you will install the git client

Change to your `~/download` directory

Use the “`wget`” program to download the git code tarball from:

`https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz`

Uncompress the archive file

Extract the individual files from the archive

Change to the new git-2.9.5 directory

Create the Makefile, specifying `/home/username/git295` as the installation directory (or `/users/username/git295` if you are on a Mac)

Run `make` to build the software

This will take a while so, while `make` is running, create a Github account by following the instructions on the next page

Github account

If you already have a Github account that you would like to use for your classes at Chapman, post a link to your Github profile in Blackboard.

If not, then

- go to https://github.com/
- Create a (free) account
- Post a link to your Github profile in Blackboard
- Edit your Github profile to display your real name

Install the git client and make it the default

Once make has finished, use make install to install the client to the `~/git295` directory

Edit your .bash_profile and add the line

```
PATH=/home/username/git295/bin:$PATH
```

Or, for Mac users

```
PATH=/users/username/git295/bin:$PATH
```

Before the `export PATH` line

Then type `source .bash_profile` to execute the commands in your `.bash_profile`

Configuring Git

In your home directory, type

```
git config --global user.name “Michael Fahy“
git config --global user.email fahy@chapman.edu
```

(Of course you will replace my name and email address with your own)
This will create a `.gitconfig` file in your home directory.

The username.sh script

You will create a bash script named “`username.sh`” that will prompt a user to type a potential username and will check to see if the string typed by the user obeys the rules for a username which (for the purposes of this assignment) are:

- The only characters that can be used are
  - lower case letters,
  - digits, and
  - the underscore character
- It must start with a lower case letter
- It must contain at least three but no more than 12 characters

This script is very similar to the zip.sh script that you created on Day 5 and you will use the zip.sh code as a starting point.

Cloning the username project

Login to your github account and go to

`https://github.com/chapman-cpsc-298/username`

(This one time I really mean type “username” !)

Fork a copy of the project to your personal github account by clicking the

button in the upper right hand corner,

Clone your fork by:

- Clicking the

button
- Copying the URL of the repo
- In your home directory on your terminal, type `git clone` and paste in the URL of the repo

This will create a directory named `username` in your home directory

Creating the `username.sh` script

Change to your username directory

Copy the `zip.sh` file to a file named `username.sh`

Edit the `username.sh` file by updating the filename and adding the user prompts that display the rules.

Add the `username.sh` file to the staging area by typing:

```
git add username.sh
```

Commit your change to the local repository by typing

```
git commit –m “Adds username.sh”
```

Push your change to github by typing

```
git push –u origin master
```

Regular Expressions

We covered regular Expressions on Day 7

Recall:

{x} means find x number of occurrences

Something new:

{x,y} means find at least x but not more than y occurrences

Finish the `username.sh` script

Finish the `username.sh` script so that it will check to see if the string typed by the user obeys the rules for a username.

When it is working properly:

- Add it to your staging area
- Commit to your local repository
- Push it to your github account
- Post a link in Blackboard to your github repository

Quiz 6 and review quiz

If you have created and successfully tested the `username.sh` script, pushed it to github, and posted the link in Blackboard, take Quiz #6 until you get a perfect score.

After you get a perfect score on Quiz #6, take the Review Quiz as many times as you can.