An introduction to git & GitHub

The standard tools for version control

Alex de Siqueira, Data Science Outreach Lead
Ciera Martinez, Biodiversity and Environmental Sciences Lead
Adam Anderson, Research Training Program Manager
Maryam Vareth, I4H Co-Director, Health and Life Sciences Lead
Liliana Cardile, Industry Relations and Events Specialist
Marsha Fenner, Communications/Program Manager
David Mongeau, Executive Director
Alexandre (Alex) de Siqueira

- Graduated in Mathematics (2007)
- Master’s, PhD in Materials Science (2009–2015)
- Native language: Portuguese
Part 1. Why was git invented?
Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(
Linux in 2020: 27.8 million lines of code in the kernel, 1.3 million in systemd

Systemd? It's the proper technical solution, says kernel maintainer

Tim Anderson    Mon 6 Jan 2020 // 12:30 UTC

The Linux kernel has around 27.8 million lines of code in its Git repository, up from 26.1 million a year ago, while systemd now has nearly 1.3 million lines of code, according to GitHub stats analysed by Michael Larabel at Phoronix.

Icons by the GNOME Project, CC-BY-SA
Band Aid icon, public domain

BitKeeper is the original distributed source management system. Now available as Open Source under the Apache 2.0 License.

BitKeeper is a fast, enterprise-ready, distributed SCM that scales up to very large projects and down to tiny ones.

At www.bitkeeper.org
Why should I use git?
● Two scenarios
From Tango! icons
"FINAL.doc"

FINAL.doc!

FINAL_rev.2.doc

FINAL_rev.6.COMMENTS.doc

FINAL_rev.8.comments5.CORRECTIONS.doc

FINAL_rev.18.comments7.corrections9.MORE.30.doc

FINAL_rev.22.comments49.corrections.10.##WHYDIDICOMETOGRADSCOLLEAD??.doc
Alice and Bob both modify the file. Alice does this:

```java
class Foo {
    public void printAWittyMessage() {
        System.out.println("Alice is the coolest");
    }
}
```

and Bob does this:

```java
class Foo {
    public void printAWittyMessage() {
        System.out.println("Alice is teh suk");
    }
}
```

Alice checks her version in first. When Bob attempts to check his in, Git will warn him that there is a conflict and won't allow the commit to be pushed into the main repository. Bob has to update his local repository and fix the conflict. He'll get something like this:

```java
class Foo {
    public void printAWittyMessage() {
        System.out.println("Alice is the coolest");
    }
}
```

The `<<<<<< HEAD:<some git nonsense>
System.out.println("Alice is the coolest");
======
System.out.println("Alice is teh suk");
>>>>>> blahdeblahdeblah:<some more git nonsense>
}
```
Part 2. GitHub and friends
- Hosting services for software development
- Management using git
- Issue tracker
- Other bells and whistles
Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.
Join GitHub

Create your account

Username *
chooseyourusername

Email address *
youraddress@yourprovider.com

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lower case letter. Learn more.

Email preferences

☐ Send me occasional product updates, announcements, and offers.

Verify your account
Email preferences

- [ ] Send me occasional product updates, announcements, and offers.

Verify your account

Pick the spiral galaxy

Create account

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.
Part 3. Basic workflow of git
How Git works

Here is a basic overview of how Git works. If you will probably work with git:

1. Create a "repository" (project) with a git hosting tool (like Bitbucket)
2. Copy (or clone) the repository to your local machine
3. Add a file to your local repo and "commit" (save) the changes
4. "Push" your changes to your master branch
5. Make a change to your file with a git hosting tool and commit
6. "Pull" the changes to your local machine
7. Create a "branch" (version), make a change, commit the change
8. Open a "pull request" (propose changes to the master branch)
9. "Merge" your branch to the master branch

Get started with Git
● Creating a repository
Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template
Start your repository with a template repository's contents.

No template

Owner *  Repository name *

alexdesiqueira  /  upglo-git-webinar-2021

Great repository names are short and memorable. Need inspiration? How about curly-rotary-phone?

Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you’re importing an existing repository.

☑ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set 🍀main as the default branch. Change the default name in your settings.

Create repository
• Cloning your repository to your PC
README.md

upglo-git-webinar-2021
Clone

HTTPS  SSH  GitHub CLI

https://github.com/alexdesiqueira/upglo

Use Git or checkout with SVN using the web URL.

Download ZIP
This command will clone the repository to your PC.
• Creating branches and adding your changes
Working on different branches

- main
- add_dog
- remove_shoe

👟🍏🐶
👟🍏
This command will create a new branch on your repository.
Press ‘q’ to come back to the terminal.

1. `git status`
   On branch my-awesome-branch
   Changes not staged for commit:
   (use "git add <file>..." to update what will be committed)
   (use "git restore <file>..." to discard changes in working directory)
   
   modified:   README.md

   no changes added to commit (use "git add" and/or "git commit -a")

2. `git diff README.md`
   diff --git a/README.md b/README.md
   index 582a282..95457b3 100644
   --- a/README.md
   +++ b/README.md
   @@ -1 +1,6 @@
   -# upglo-git-webinar-2021
   \ No newline at end of file
   +# upglo-git-webinar-2021
   +
   +This is a test repository for our UpGlo webinar, given on Apr 28, 2021.
   +Feel free to add your text below:
Adding changes, checking status and committing added files

1. `git add README.md`
2. `git status`
   - On branch my_awesome_branch
   - Changes to be committed:
     - (use "git restore --staged <file>..." to unstage)
     - modified: README.md
3. `git commit -m 'Adding info on the webinar'`
4. `git push origin my_awesome_branch`
   - Enumerating objects: 5, done.
   - Counting objects: 100% (5/5), done.
   - Delta compression using up to 8 threads
   - Compressing objects: 100% (2/2), done.
   - Writing objects: 100% (3/3), 369 bytes | 369.00 KiB/s, done.
   - Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
   - remote: Create a pull request for 'my_awesome_branch' on GitHub by visiting:
     - https://github.com/alexdesiqueira/upglo-git-webinar-2021/pull/new/my_awesome_branch
   - To github.com:alexdesiqueira/upglo-git-webinar-2021.git
     * [new branch] my_awesome_branch -> my_awesome_branch

Sending changes to the repository
git has a “staging area”

- A place to review your modifications before finishing the process
• Opening a pull request
upglo-git-webinar-2021
You can add a title, comments, labels, ...

... and see the changes again
“Merge” available if you have enough permissions for the repository.
Adding info on the webinar #1

alexdesiqueira merged 1 commit into main from my_awesome_branch now

alexdesiqueira commented 10 minutes ago

No description provided.

Adding info on the webinar

alexdesiqueira merged commit 4083e5a into main now

Pull request successfully merged and closed

You’re all set—the my_awesome_branch branch can be safely deleted.
• Pulling changes back to your PC
Gets back to the “main” branch and pulls the latest changes

```
$ git checkout main
Switched to branch 'main'
Your branch is up to date with origin/main'.
$ git pull origin main
remote: Enumerating objects: 100%, 1/1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 648 bytes | 648.00 KiB/s, done.
From github.com:alexdesiqueira/upglo-git-webinar-2021
  * branch     main    -> FETCH_HEAD
  8c19844..4083e5a main    -> origin/main
Updating 8c19844..4083e5a
Fast-forward
README.md | 7 +++++++-
1 file changed, 6 insertions(+), 1 deletion(-)
```
Part 4. Using git without the command line
Add event handler to dropdown component

Aneurin Shepherd and Markus Olsson committed c79e71c 1 changed file

Co-Authored-By: Markus Olsson <nisk@users.noreply.github.com>

app/src/ui/toolbar/dropdown.tsx

```typescript
145 // -145.6 +145,10
146 export class ToolbarDropdown extends React.Component {
147
148 this.state = {
149   clientRect: null
150 }
151
152 private get isOpen() {
153   return this.props.dropdownState === 'open'
154 }
155
156 private dropdownIcon(state: DropdownState): Octicon {
```

GitHub Desktop
Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

Get it at GitHub Desktop
Also, check its documentation
**Measurement results**

The file with measurement results (default: `results.csv`), returned by this pipeline, contains the following measurements:

- `image_id`, the filename of the processed image.
- `left_wing (mm)`, distance from the tip of the left wing to the left shoulder.
- `right_wing (mm)`, distance from the tip of the right wing to the right shoulder.
- `left_wing_center (mm)`, distance from the left wing to the center.
- `right_wing_center (mm)`, distance from the right wing to the center.
- `wing_span (mm)`, distance from the tip of the left wing to the tip of the right wing.
- `wing_span (mm)`, distance from the tip of the left shoulder to the tip of the right shoulder.
- `position`, position of the butterfly being `right-side_up` or `upside_down`.
- `gender`, gender of the butterfly being `female` or `male`, returned if `position` is `right-side_up`.

**Example**

Example data can be found at [https://github.com/machine-shop/butterfly-wings-data](https://github.com/machine-shop/butterfly-wings-data). For this example, clone the repository alongside `butterfly-wings`.

```bash
git clone https://github.com/machine-shop/butterfly-wings.git
git clone https://github.com/machine-shop/butterfly-wings-data.git
```

Resulting files:

```
$butterfly-wings$
```

```
$pipeline.py$
```

```
$image1.jpg$
```

```
$image2.jpg$
```
**SOURCE CONTROL**

- **Branches:**
  - Show All
  - Show Remote Branches

**Graph**

- **main** (origin) (origin/HEAD)
  - Merge pull request #1 from alexdesiqueira/my_awesome_branch

- **myAwesomeBranch** (origin)
  - Adding info on the webinar
  - Initial commit: 19 Apr 2021 19:06

**COMMENTS**

- Last fetched 3 hours ago

**FILE HISTORY**

**LINE HISTORY**

**BRANCHES**

**REOMTES**

**STASHES**

**TAGS**

**SEARCH & COMPARE**

**main**
Use Git version-control tools in Visual Studio Code

22 min • Module • 7 Units

🌟🌟🌟🌟 4.8 (177)

Beginner Developer Visual Studio Code

Utilize the tight integration of Visual Studio Code with the Git source-control management system.

Learning objectives

In this module, you will:

- Perform common GitHub tasks by using the Command Palette in Visual Studio.
- Monitor the status of your work
- Commit your files to your repositories from the Source Control view

Available at the [Microsoft documentation](https://docs.microsoft.com)

---

Working with GitHub in VS Code

Using GitHub with Visual Studio Code lets you share your source code and collaborate with others. GitHub integration is provided through the [GitHub Pull Requests and Issues](https://github.com) extension.

Install the GitHub Pull Requests and Issues extension

To get started with the GitHub in VS Code, you’ll need to create an account and install the [GitHub Pull Requests and Issues](https://github.com) extension. In this topic, we’ll demonstrate how you can use some of your favorite parts of GitHub without leaving VS Code.

If you’re new to source control and want to start there, you can learn about VS Code’s source control integration.

Available at the [VSCode documentation](https://code.visualizer.com)
Ending
Your tasks:

- Open your GitHub account (if you don’t have one already)
- Clone the repository [github.com/alexdesiqueira/upglo-git-webinar-2021](https://github.com/alexdesiqueira/upglo-git-webinar-2021)
- Create your own branch and add a new line with some text on README.md
- Make a commit to that repository, adding your changes
Some pain until reaching the solution
Two lessons:

- Everything you make on git can be reverted
- Having good friends is great 😊
  - Git documentation, StackOverflow, google/bing/duckduckgo, ...
Going further

Git

- Git: [https://git-scm.com/](https://git-scm.com/)
- Download and install git here: [https://git-scm.com/downloads](https://git-scm.com/downloads)

Git tutorials

- [Software Carpentry’s “Version Control with git”](https://software-carpentry.org/githt/)
- Several tutorials on git: [https://git-scm.com/doc/ext](https://git-scm.com/doc/ext)
  - Highly recommend [Git Immersion](https://git-scm.com/doc/ext) on that list
  - [Git for Designers](https://git-scm.com/doc/ext) is a good introduction, too
- [Google’s Tech Talk: Linus Torvalds on git](https://www.youtube.com/watch?v=cKZsKkxM4JQ)
- [Matthew Brett’s “Learn git right for a long and happy life”](https://www.youtube.com/watch?v=cKZsKkxM4JQ)
  - Please read this one! 😊
Going further

GitHub and friends

- GitHub: https://github.com/
  - Open your account at https://github.com/join
- GitLab: https://gitlab.com/
  - Open your account at https://gitlab.com/users/sign_up
- Bitbucket: https://bitbucket.org/
  - Open your account at https://id.atlassian.com/signup?application=bitbucket

Graphical User Interfaces for git

- GitHub Desktop
- Microsoft’s “Use Git version-control tools in Visual Studio Code”
Thank you!