1 What is a Branch?: Git References

1.1 Lecture

- Each commit defines a history through its parent(s)
  - A **reference** to a commit says that its history is important
    - Makes it findable so you can easily make use of it
    - Tells Git not to garbage-collect it
- 3 kinds of **references** in Git, plus `HEAD`
  - Tags
  - Branches
  - `HEAD` (technically a **symbolic reference**)
  - Remotes
- All reference objects (usually commits)
  - Can be thought of as (reference name, object hash) pairs
- Generally contained in `.git/refs/<reference-type>/<reference-name>`
  - Each **reference** file *only* contains the commit hash being referenced
    - Makes creating and deleting references really cheap
      - Aligns with one of Git's design goals: "strong support for nonlinear development (thousands of parallel branches)" (https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git).

1.1.1 Tags

- Two kinds: annotated and lightweight

Lightweight Tags

- **Location:** `.git/refs/tags/<tag-name>`
- Just object references

Annotated Tags

- **Location:** `.git/refs/tags/<tag-name>` and `.git/objects/<hash>`
- Objects as well as references because they contain metadata as well as a name and hash
  - Metadata: tagger, time of tagging, message

1.1.2 Branches

- **Location:** `.git/refs/heads/<branch-name>`
- Just like lightweight tags, except they can move when they're checked out.
  - Updated when you make a new commit and they're in `HEAD`

1.1.3 `HEAD`

- **Location:** `.git/HEAD`

- A symbolic reference (most of the time)
  - Refers to a reference (almost always a branch) instead of directly to a commit
- `HEAD` specifically will always refer to a branch (symbolic ref) or commit (normal ref)
- A repository is said to be in “detached HEAD state” when its `HEAD` refers to a commit
- Refers (directly or indirectly) to the currently checked-out commit or branch
  - I *think* that this means that `HEAD` is the commit whose tree the Staging Area is populated from
- Whenever a new commit is made via `git commit`, its parent is the commit `HEAD` refers to

1.1.4 Remotes

- **Location:** `.git/refs/remotes/<remote-name>/<branch-name>`
- Represent other repositories’ branches at last `git fetch`/`git push`
- Read-only: Can `git switch` to them (in detached HEAD state), but can’t commit on them with `git commit`

1.2 Lab/Homework

- Follow along with 10.3 in Pro Git (ignoring the Remotes section)