Help Topic: The Git Clone and Checkout Commands

Spring 2022

Oscar Viquez, oviquezr@mit.edu
Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

Cloning a Repository with the `clone` Command

The `git clone` command is likely the first command you'll need to know since this is how you get a local copy of a Git repository from a server to your machine. The basic command is simple, but there are a few other useful ways of using it that we discuss here. More info can always be found by Googling "Git book" and reading the full PDF online free, or just typing `git help clone` anytime on the command line.

Basic use of the `git clone` command

The basic syntax for the `clone` command is:

```
$ git clone <repository> [<directory>]
```

For example, the course software can be checked out with:

```
$ git clone https://github.com/moos-ivp/svn-mirror.git moos-ivp
```

Note that you can provide a name for the locally checked out directory that may be different than the repository name. In the example above, if you omitted the "moos-ivp" as the last command-line argument, the checkout would have produced a file named "svn-mirror" on your local machine.

Checking out an older revision with the `checkout` Command

One of the core strengths of a version control system is the ability to undo changes, or at least go back to an older version if need be. This can be done with the `git checkout` command by providing information on the command line about the version you want to go back to. For example, you can check out a copy of the course software as it looked on February 27th, 2020, in the following way.

```
$ git clone https://github.com/moos-ivp/svn-mirror.git moos-ivp-time-travel
$ cd moos-ivp-time-travel
$ git checkout 1701edd418b6cab7733d67c8a4c34f11799d973e
```

The third line in the set specifies a particular commit ID to check out. The commit ID is a unique SHA-1 hash created when the commit is created. You can get the commit ID you are looking for by using the `git log` command. This is described in a separate topic.

Another way to checkout the repository as it looked on February 27th, 2020, is to stack git commands using bash command nesting. The `git rev-list` command lists commits in reverse chronological order, and accepts options such as date boundaries and limited list length. The preceding commit ID can be obtained via the following:

```
$ git rev-list --max-count=1 --before=2020-02-28 master
1701edd418b6cab7733d67c8a4c34f11799d973e
```

Nesting one command into there other would look like this:

```
$ git checkout $(git rev-list --max-count=1 --before=2020-02-28 master)
```

Note: The final argument in the `git rev-list` example above corresponds to the target branch. Branches are an important feature of the git workflow in complex projects and are described in a separate topic.