UNIVERSITY of York

# Using git to download and update BOUT++

Peter Hill

York Plasma Institute

## Outline

- What is git?
- Getting git
- Basic git usage
- Getting BOUT++
- Compiling BOUT++
- Running examples
- Contributing to BOUT++

## What is git?

### Version Control System (VCS)

- Version control systems record changes to a file/set of files over time
  - Not just software! This talk is under git
  - Allows you revert files back to a previous state, compare changes over time, see who last modified something, etc.

### Local VCS

- Naive versioning: separate folders for each version
- Slightly better: local database of changes

## What is git?

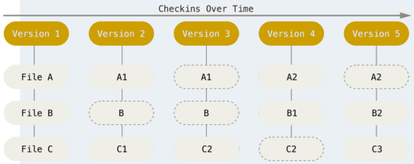
### Centralised vs Distributed VCS

- Centralised VCSs: CVS, Subversion
  - Have a single server than contains all the versioned files
  - Can see what other people are working on
  - Easier to administer a centralised VCS than local databases on each client
  - If server goes down, can lose access to project history, etc.
  - If central database is lost, everything not backed-up is lost
- Distributed VCSs: git, mercurial
  - Clients don't just checkout latest snapshot of files, repository is fully mirrored
  - If server goes down, any client repo can be copied back to server to restore it
  - Multiple remote repos work pretty well

UNIVERSITY of York

## How does it work?

### Snapshots



- git thinks of its data like a set of snapshots of a miniature filesystem.
- Every time you commit, it takes a picture of what all your files look like and stores a reference to that snapshot

## git is local

- Vast majority of operations are local
    - Doesn't need to talk to remote servers to get e.g. history
- This means you can continue to work offline, including committing changes to the database
- Checking out a copy of the repository means you have a full copy
    - You can copy your local version onto a USB stick and hand it to someone
    - They now have access to the project history, can make changes, etc.

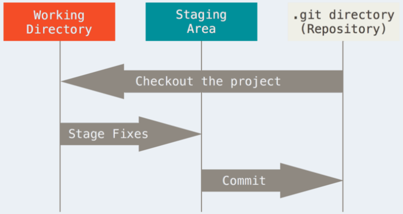
## git has integrity

- Everything in git is check-summed
- References are to checksums
  - git can immediately detect if data gets lost in transit or files are corrupted
- Checksums are done using SHA-1:

  `24b9da6552252987aa493b52f8696cd6d3b00373`
- git stores everything, not by name, but by hash value of its contents

## The Three States

- Important to understand correctly
- Three main states that files can be in:
  1. Committed: data stored in repo
  2. Modified: file is changed but not committed
  3. Staged: modified file marked to go into next commit



UNIVERSITY *of York*

## Getting git

### Linux

Get git through your package manager:

```
sudo yum install git
sudo apt-get install git
sudo zypper install git
```

### Mac

Install Xcode Command Line Tools, then try to run `git`. If you don't have it installed already, it will prompt you to install it.

More up-to-date version: http://git-scm.com/download/mac

Alternatively, use GitHub for Mac: http://mac.github.com

## Getting git

### Windows

Official build: http://git-scm.com/download/win Note that this is actually “Git for Windows”

Alternatively, use GitHub for Windows: http://windows.github.com

### From source

Download tarball from either https://github.com/git/git/releases or https://www.kernel.org/pub/software/scm/git

Compile and install, then you can get git via git!

## Getting a repo

- Two choices: import an existing project into git, or clone an existing git repo from somewhere else

### Initialise a repository

- Go to project's directory:

```
git init
```

- Creates a new subdirectory names .git containing all the necessary repo files
- Nothing in project is tracked yet

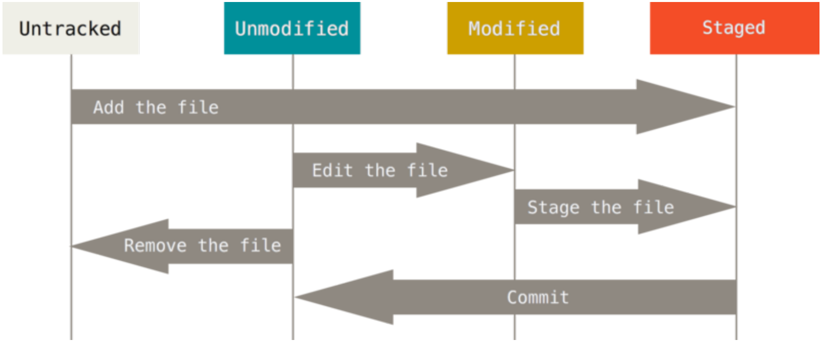
## Cloning an existing repository

```
# Clones into ./BOUT-dev
git clone https://github.com/boutproject/BOUT-dev.git
# Clones into ./BOUT++
git clone https://github.com/boutproject/BOUT-dev.git BOUT++
```

- Doesn't just get a working copy - git receives a full copy of nearly all data that the server has
- This used the HTTPS protocol, but you can also use SSH like `user@server:path/to/repo.git`
- To use SSH on GitHub, add your (public!) SSH key to your GitHub profile

UNIVERSITY *of York*

## Lifecycle of a file



## Getting help

```
git help <command>
git <command> --help
git <command> -h
man git-<command>
```

## Checking status

```
git status
git status -s
```

## Tracking new files

```
git add <filename>
git add *.cxx
```

## Staging modified files

```
git add <changed file>
```

## Ignoring files

.gitignore

## Viewing changes

```
git diff
git diff --staged
git diff <filename>
git difftool --tool=meld
```

UNIVERSITY of York

## Committing changes

```
git commit
git commit -m "Useful commit message"
git commit -a -m "This commits everything"
```

## Removing files

```
git rm <filename>
```

- Doesn't delete file! Just removes it from git repo

## Moving files

```
git mv <old file> <new file>
```

- This is synonym for

  ```
  mv <old file> <new file>
  git rm <old file>
  git add <new file>
  ```

## Viewing project history

```
git log
```

- Huge amount of options here
- Add to your .gitconfig

UNIVERSITY of York

## Getting BOUT++

Use git to clone the repository:

```
# Clones into ./BOUT-dev
git clone https://GitHub.com/boutproject/BOUT-dev.git
# Clones into ./BOUT++
git clone https://GitHub.com/boutproject/BOUT-dev.git BOUT++
```

# Compiling BOUT++

Basic configure and make:

```
./configure --with-lapack --with-netcdf
make
```

Then check examples work:

```
cd ./examples
./test_suite_make
./test_suite
```

## Submitting a bug report

Open an issue on GitHub

## Contributing to BOUT++

1. Create a new branch

   ```
   git branch my-new-feature
   git checkout my-new-feature
   ```

2. Make changes
3. Run test_suite if necessary
4. Commit (with a nice message!)

   ```
   git add README.md
   git commit -m "Fiddle with contributing section in README"
   ```

5. Push to GitHub

   ```
   git push
   ```

6. Submit pull request

## Acknowledgments

Some material from *Pro Git*, Second Edition by written by Scott Chacon and Ben Straub and published by Apress. Available here: https://git-scm.com/book