An Introduction to HPC and Scientific Computing – Using Repositories and Good Coding Practice
– Practical Sessions –

Ian Bush *ian.bush@oerc.ox.ac.uk*

Contents

1 Introduction

In these exercises we will look at using the git revision control system

2 Getting Started

In this exercise we will ask you to run through a short online introduction to git. Open a web browser on your local machines, go to

https://try.github.io/levels/1/challenges/1

and work your way through the tutorial found there. This will not only cover what was in the lecture, but also touch on subjects such as branches, and show you how to use a remote repository.

3 Git on Arcus-b

Now we shall see if we can use our new knowledge to create and manage a simple repository on arcus-b.

Log onto arcus-b and

1. First set up the software environment with `module load git`
2. Create a new directory using `mkdir` and change into it
3. Now create a repository by using `git init`
4. Write a program in C that write "Hello world to the screen"
5. Using `git add` and `git commit` add file this to your repository
6. Examine the output of `git status` and `git log` and make sure you understand them
7. Modify you program to say hello to you rather than to the world in general
8. Update the version in the repository using the appropriate commands
9. Update your program again to not only greet you, but also to print your age and check this into the repository
10. Write a function in C to calculate the square of an integer. Put this in a separate file to the one with the main program in
11. Add this new file to the repository
12. Modify you main program to call the function which squares a number to print out the square of your age. make sure you can compile and run this all correctly

13. Check the appropriate files into your repository

14. Finally you decide that printing out the square of your age makes you feel much too old, and you want to go back to the version where it just printed your name and your age as is. Use the appropriate git commands to go back to this version.

4 Accessing a remote Repository

Finally we shall breifly look at using git to get access to a piece of software. To do this we shall use the `git clone` command, which make a copy of a remote repository to give you a local version. We shall look at QuEST, a library developed by my RA Ania Brown which is used to simulate Quantum computers. See

https://quest.qtechtheory.org/

for more details.

To access QuEST all you need do is issue the command

```
git clone https://github.com/aniabrown/QuEST.git
```

This will download the repository and create a new local one in the directory

1. change into QuEST and issue a `git status` command. Is the result what you expect?

2. Use `git ls-files` to see all the files in the repository

3. There is an example program using the library in this directory. Issue `make` and then `./demo` to see a simple example of QuEST at work

4. QuEST is written in C, cd into the QuEST directory and have a look at the source. Not in particular how constant the style is throught out all the code.