Imperial College London – Department of Computing

COMPM0101 - Introduction to Remote Learning

Tutorial Exercise: Working With DoC Systems

This exercise gives you a brief introduction to the systems you will need to use for practical work in the Department of Computing (DoC). The department provides computers that you can use to do your work, and other services that you will be interacting with.

CSG

All departmental resources (the machines in the lab and the systems described below) are managed by the DoC *Computing Support Group* (CSG). If you have any problems or questions about the operation of any DoC hardware or systems, then visit the CSG helpdesk in Huxley room 305, or email

doc-help@imperial.ac.uk

CSG provide some extremely helpful guides to working in DoC and getting the best out of the available resources. You can find these guides at

http://www.imperial.ac.uk/computing/csg/

They are highly recommended, and I will refer to some of the essential ones later in this document.

The Exercise

To complete this exercise you will need to obtain some provided code, make some minor changes, and then submit your work. The work involved in doing this should be quite minimal — this is not a programming exercise. The exercise is intended to show you one of the common ways in which work that involves code is distributed and submitted in the department. It is worth taking some time now to get to know the tools involved.

Stage 1: Obtain the Specification from CATE

Actions

As you have already done in this case, for each exercise you need to:

- Go to https://cate.doc.ic.ac.uk, find your exercise timetable, and the entry that represents the exercise.
- Download the specification.

Explanation

- Every exercise has an entry in CATE. These are shown in a timetable format to help you plan your time. All *assessed* exercises should show in the timetable by the start of the course.
- The CATE entry has multiple uses. The title of the exercise is a link to the specification. More files might be available via the **G** ("given files") button.
- Work is also submitted to CATE. Normally, this will be by directly uploading a file or files. The submission or "hand in" page is accessed using the **H** button. This exercise uses a different mechanism, but the submission stills end up appearing in CATE.
- CATE also has an **N** button for each course, where you will find course notes.

Stage 2: Log in to a Lab Computer

Actions: (If You Are In The Lab)

- When you sit down at a lab machine you will get a Linux log-in screen. Use your college username and password to log in.
- Open a terminal. The terminal should open at your DoC *home directory*. Have a look around.

Actions: (If You Are Working Elsewhere)

- Connect securely to a lab machine from your computer by following the CSG guide on SSH access:

  https://www.imperial.ac.uk/computing/people/csg/guides/remote-access/ssh/
- To make the "2nd hop" of your connection, you can run the recently created `sshtolab` script. This will pick a free machine for you, so you do not need to specify one. (Use of `sshtolab` has not yet made it into the CSG guide, but see "Further Reading" below.)
- **Make sure** you reach an actual lab machine (do two hops) — do not stay on a shell server. Running programs on shell servers is a good way to make CSG unhappy. You won't like them when they are unhappy!
- The session should open at your DoC *home directory*. Have a look around.

Further Reading

- A very detailed guide to remote working for DoC students has been created by one of the department's Teaching Scholars, Nuri Cingillioglu. It includes videos and goes far beyond the basics of ssh:

https://www.doc.ic.ac.uk/~nuric/teaching/remote-working-for-imperial-computing-students.html

Explanation

- There are in excess of 250 workstations for undergraduate and taught postgraduate (MSc) students to use.
- Details of the machines, including the hostnames you can connect to can be found at:

  https://www.imperial.ac.uk/computing/people/csg/facilities/lab/workstations/
- The lab machines are dedicated DoC machines — not general purpose Imperial College ones. They provide acccess to your DoC personal file space ("home directory") and software that is used in the courses you will be taking.
- You may do a lot of work on your own computer this year, but the lab machines are still available — they are all accessible remotely — and using them might be essential for certain tasks.
- All the lab computers, except the few Apple Macs, run a CSG-maintained version of Ubuntu 20.04 Linux.
- You have a personal, networked DoC home directory. When you log in to any lab PC you will automatically be connected to your home directory, which will be *mounted* in the local file system at `/homes/<your-id>`.
- There is information about your home directory storage on these CSG pages:

  http://www.imperial.ac.uk/computing/people/csg/services/file-storage/

  http://www.imperial.ac.uk/computing/people/csg/guides/file-storage/quota/
- Most (if not all) practical assignments are either OS-independent, or expected to be completed in Linux. If you have not used Linux before you will need to familiarise yourself with it. Getting through this exercise will get you started. Take the opportunity now to open a web browser so you can look up some basics.

Stage 3: Obtain the Provided Code

Actions

- A *Git repository* (repo) for the exercise has been created for you on the DoC GitLab server. The url is https://gitlab.doc.ic.ac.uk/lab2122_autumn/lab_intro_101_⟨your-id⟩.git. (You need to replace `<your-id>` with your college username.)
- *Clone* the repo into your home directory. The clone command will create a directory called `lab_intro_101_<your-id>` under your home directory.
- `cd` into the new directory.

Explanation: Git (https://git-scm.com/)

- This exercise uses the command line tool `git` to work with a provided *code repository*. One of the ways work is issued and submitted in DoC is through such Git repos.
- Git is a very widely used *version control* system, and many of you will be familiar with it already. If you are not, then now is the time to find out about it, as you will be using Git extensively throughout your degree. A good resource is 'Pro Git', which is available here as an e-book:

  https://git-scm.com/book/en/v2

  Chapter 2 covers all the basics (like how to create your own repo, or clone an existing one). Chapter 3 is also essential reading (but not required until later in this exercise).
- You can easily clone code onto your own computer and work from there, if you have the right environment. Do this exercise a lab machine, so you find out how.

Stage 4: Edit the Files

Actions

- Modify the provided files so that the script `hello.rb` ouputs a personalised message.

Explanation

- Before any modifications the script should already be runnable. It will generate a non-personal message. Try it.
- Running the script with the `-h` option will print a useful help message. Try it.
- If you cannot run `hello.rb`, try to figure out why and solve the problem. Possible reasons are:
  - the file has the wrong *permissions* set;
  - you have not learned how to execute a file in Linux using the command line / shell / terminal.
- There are various text editors available that can be used via a terminal. If you have not used an editor like this before, the easiest to start with is probably `nano`. If you have, then try opening your favourite — it is probably there.
- The script is written in Ruby. You are not expected to know any Ruby. This is not really a programming task.

Stage 5: Commit Your Changes and Update the Server

Actions

- Commit the changes you have made to the repo in your home directory.
- Push your commit(s) back to the ‘origin’ repo on the GitLab server.

Explanation: GitLab (https://about.gitlab.com/)

- CSG maintain a GitLab service for the department. This gives a centralised platform where the ‘origin’ version of a git-based project can be hosted. Some exercises, like this, may be issued as a GitLab project that has already been set up for you.
- There is a web interface for the DoC GitLab service at

  https://gitlab.doc.ic.ac.uk

  where you can also create your own projects.
- More information about the CSG GitLab service is available here:

  https://www.doc.ic.ac.uk/csg/guides/version-control/gitlab

Stage 6: Check Your Solution

Actions

- Log in to the LabTS system at:

  https://teaching.doc.ic.ac.uk/labts
- Click the link for this exercise (’Lab_Intro_101’).
- Run the LabTS tests for one of the versions of your code and check the results.

Explanation

- LabTS was developed within DoC by Dr Mark Law, Dr Mark Wheelhouse and Dr Tristan Allwood. It is another part of the infrastructure used to manage coding exercises. It is integrated with the GitLab system and can access and run the code that you have pushed to the server.
- The LabTS web site shows all the exercises that have been set for your class. For each exercise it allows you to run a set of tests provided by the creator of the exercise, and then make a submission.
- LabTS is provided to allow you to check your work before submitting. The same system (but not necessarily the same tests!) will be used to generate test output to help assess

your work. So, using LabTS means you can see whether your code actually runs and produces output in the test environment.

- LabTS should not be used to do the bulk of your testing during development. It is for doing occasional 'sanity checks'.
- LabTS does not mark your work. Testing is not the same as marking, and passing all the tests does not imply that your code is faultless. Markers will be looking at the code itself. They will have the test output to help them understand what you have done and award a mark. So, make your code presentable. In fact, make it beautiful.

Stage 7: Make a Submission

Actions

- Choose an appropriate commit and submit it from the LabTS web page.
- Go to CATE and confirm that the submission was successful.

Explanation

- It is your responsibility to submit your work properly, by the deadline. Unsubmitted commits on the GitLab server or in your local repository will not be accepted for marking.
- Only commits that appear in CATE are submissions.
- You can update an existing submission to a new version any time before the exercise deadline. Just submit the new commit using LabTS in the normal way. This will overwite the previous submission.

Stage 8 (Homework Extension): Make Another Clone

Actions

- Clone your repository to your personal computer.
- Make some more updates to the files and push the changes to GitLab.
- Experiment with creating a second branch in your repository, commit some changes to the new branch and then merge the new branch with the initial 'master branch'.

Explanation

- The GitLab server is accessible from outside Imperial, so you can clone, pull and push directly from your own computer.
- The most powerful feature of Git is the ability to make 'cheap' branches in your local repository. These give you much more flexibility when it comes to managing a project. See Chapter 3 of the Pro Git e-book for details.
- For the current exercise, you will only be able to run and test your work locally if you have a Linux system with Ruby installed.

Congratulations

You have completed your first exercise — well done. (Did you submit before the deadline??)

Assessment and Feedback

This exercise carries no credit and will not be assessed.