Linux Tutorial – Navigating the Diamond Cluster

Aims

The aim of this tutorial is to get a novice Linux User up to speed at navigating the Diamond cluster and getting started with data processing. Feel free to skip over anything that you already know and remember that there is often more than one way to achieve the same process.

Shortcut: if you are an expert at Linux, you can skip over everything except for the things in red boxes. Please do these so we can check that everyone has the correct permissions and abilities.

Remember that commands are case-sensitive and are almost always in lower-case. In this tutorial, anything that is for you to type is written in bold such as the **tree** command here.

There is deliberately some repetition in this tutorial, as it will help you get confident with all the commands.

Please experiment! (we are scientists after all) – if you accidentally set off something that takes a long time, all you need to do is **Ctrl + C** and that will halt the current command. If you can suggest ways to improve this tutorial, please don't hesitate to contact me – felicity.bertram@diamond.ac.uk

You should already have a Fed ID, and know what it is – please ask if you don't know what yours is. It forms part of the address for your home directory, and it is what identifies you on the system. Every experiment at Diamond operates in a "Proposal", and the Proposal ID for this course is **MX30951**.

To Start

Once in an NX session, open a terminal by clicking on "Activities" in the top left corner of the screen, and then clicking on the Terminal icon which is the 5th option down (a black box with a >_ in it – hover over an icon and it will tell you what it is

See my slides from my presentation for more information about how you do this.

Basic Navigating

Prompt

When you have an open terminal, you will see a "Prompt" which consists of the following:

[{User}@{server} {location}]$

So for a User with a Fed ID of abc12345, it's likely to appear as:

[abc12345@r-sc-serv-02 ~]$

Because you are in your home directory, the location is represented as a ~ ("tilde" symbol). Diamond has lots of servers, so don't worry if your server ID is different to this.

Print Working Directory

The first thing we are going to do is find out where you are, so we want to type the command for "**P**rint **W**orking **D**irectory" (this is where you currently are) and then press enter (always press enter after a command):

pwd (and after all of these commands, you press **[Enter]** to make them start)

This should be /home/{your Fed ID} – but remember, once the main course starts you should not be working in your home directory as you will quickly run out of space there – but we will use it for some of this tutorial.

To find out what is in this folder, there are two main commands you can choose:

- **ls**
- **tree**

The command for list is:

ls (small L not capital i – remember most commands are lower case)

This will produce a list of all the directories within the current directory, in alphabetical order. If there is nothing in the current folder, nothing will appear. Note that it won't tell you what is within any of these folders.

The command for tree is:

tree

This will produce a tree diagram of every folder within the other folders in your current location. It can be easier to find things this way, but it can quickly become such a long list (mine is almost 2,000 items now), that you have to do a lot of scrolling to find what you are looking for. One way to shorten the view of the tree is to use:

tree -d

This will show a tree of all the directories, without listing the files that may be within them. There are actually lots of options, like the added **-d** that can alter how the tree is displayed, and what information it contains. To find out about all of them, you can access the manual for it.

Using the Manual

There is a manual option for Linux commands, and you access it with the **man** command. So, for instance, if we want to see all the options for the **tree** command, then we want to input:

man tree

This will open all the information about the tree command. You don't need to know all of it, just your favourite additions, like the **-d** used above. However, Linux offers options that may be useful to all sorts of people, doing all sorts of things, and these options allow you to tailor it to how you like it. To navigate around the manual, use the following keys:

[Enter] key = move down 1 line

[Space] key = move down 1 page

g = move to the top of the page

G = move to the bottom of the page (one of the few times you use a capital)

h = if you need help with the manual itself, then **h** will tell you even more ways to interact with it

q = quit

Finding Your Data

Short-cut: If you know how to navigate to /dls/i04-1/data/2021/mx30951-8 then just do it – this is where your data will be after the course

To find your data, you need to navigate away from your home directory. To do this, you will need to use the "**C**hange **D**irectory" command, which is:

cd

In its simplest form, you can just navigate to where your data is, which is stored in a uniform path which looks like this:

/dls/{beamline}/data/{year}/{visit number}

So, in this instance you can go straight there by typing:

cd /dls/i03/data/2021/mx30951-8 (note there is a space between the cd and the file path)

However, you will come across much more complex paths than this, and if you type one character wrongly, it won't work, and will return a "No such file or directory" message. There is no helpful flexible search for something similar. Therefore, you want to be able to explore folder by folder as well. If you have navigated straight to the data, then we need to reset you back to your home directory for you to follow the next steps. To jump straight to your home directory simply type:

cd (with nothing after it)

To move up one folder, in this instance from /home/{your Fed ID} you need to type:

cd .. (once again there is a space between the cd and the ..)

If you then use the **pwd** command again, it will tell you that you are now in /home (go on, try it). You will also notice changes to the prompt as you move from directory to directory. If you want to see what is in the /home folder then you can use the **ls** command once again (in this instance it will list folders for all Fed IDs currently using that node). We want to go up again, so type **cd ..** once again. This should mean that when you use the **pwd** once again, the location is simply / This is also called "root" as it is the top of the directories.

Remembering that the path for any data collected at Diamond is /dls/{beamline}/data/{year}/{visit number} you now want to navigate into /dls. To do this type:

cd dls (cd /dls also works)

pwd → this will show that you are in /dls now, not /

ls → this will show what folders there are in /dls – most are a list of lots of beamlines, not just MX

cd /i03 → you are now in the beamline directory – note that it needs to be a little i in the file path

This course is using multiple beamlines from Diamond, though most individual participants will have all of their data on one beamline. However, it is not unknown that during a course we may go back and do additional data collections if needed, and they may not all be on the same beamline – so be prepared to navigate around if needed.

cd data I know that this is the next step in the path to your data, but if you want to see what the options are, from /dls/i04-1 then you can use the **ls** command to check what the available options are. And any time you like, you can use **pwd** to tell you where you are.

You can also jump two steps at once, so first let's go back two steps by putting in the following:

cd ..

↑ (to save typing the same thing twice, you can use the up arrow to take you to the previous command)

Pressing the up arrow (↑) multiple times, will take you back one command at a time. Most commands are longer than 5 characters, so this can save a lot of typing.

You can then go back in two steps at once:

cd /i04-1/data

Now we can move another 2 steps, as we do actually know the year and visit number (the last 2 directories in the standard path to your data:

cd /2021/mx30951-8

Now this isn't your data, this is some sample data that is there for you to check that you can find it, before the beamline day. This will enable you to get comfortable navigating around the Diamond cluster, and be used in

the scripting examples that Graeme is going to going over after lunch, and you can also experiment with using ISPyB, where the data also appears, when Marco speaks later on.

Kill Command

As described at the front of this tutorial, if you start a process off, and it's taking too long, you can kill the process by using:

Ctrl and **C**

So try it out by setting off the **tree** command in this visit folder. After the visit was finished last Thursday, there were 63,424 directories containing 159,098 files (and there may have been additional auto-processing since then) so it will take your computer time to do this – at least a couple of minutes of continuous scrolling, or more. And it isn't actually necessary to look at the list as a whole. So now kill the process by typing **Ctrl** and **C**

This example serves to show how useful the scripting that Graeme is going to teach you this afternoon can be, when faced with data on this scale – you don't want to sift through that sort of volume manually.

Autocomplete

To set yourself up for the next step, type **cd ..** to take you back to /dls/i03/data/2021

When you are navigating, if you start to type an option, you can do a few letters, and then press tab to autocomplete. So, if you start typing:

cd mx3 and then press **[Tab]**

It will autocomplete to this visit. However, from the same place, if you pretend you are accessing another proposal, start by typing:

cd mx295 and then press **[Tab]**

It will start to autocomplete – but I've set this up as a trick question as there are six visits in that folder which start *mx295*. You can see these in the folder by using the **ls** command. Therefore, the autocomplete will only take you as far as **cd mx2950** so if you hit **[Enter]** after that, it won't work (and again, unlike Windows, it won't try to help you out to find what we are looking for). However, this can be a very useful time-saver.

Wildcards

You can however include a wildcard in your search string (in this instance *****) in the directory address, for instance:

ls -d /dls/*/data/2021/mx30951-8

What this will do, is look for any directories which fit this criterion, and lists them. It can take a bit of time, as it's searching quite a lot of options, and at this time of year there have been a lot of visits. But in this instance there will only be one of them. In this example the **-d** directs the search for a directory – if you omit these 2 letters, you instead get a list of the contents of that directory.

Now try instead

ls -d /dls/*/data/2021/mx29507*

And you will see a mixture of 8 directories, which are incidentally from the course before this one, showing different beamlines

Wildcard in Linux are:

***** Matches any number of characters (including none) – as we have used above

? Matches a single character in a specific position

[] Matches characters within the brackets, so **[abc]** will look for a, b or c being at that position

! Excludes characters inside the brackets, so **[!abc]** will look for any letter other than a, b or c

As we turn towards the New Year, then searching for **202?** in the string would bring up data from 2020, 2021 or 2022 – you could be more specific and say **202[12]** and this will search for either 2021 or 2022 but not 2020 (nor 2023, 2024…).

How to locate visits when you can't remember the number

Take yourself back to your home directory before starting this step, so type:

cd (bet you could remember that one now)

Then use the **groups** command, so type:

groups (typing **id -Gn** also has the same effect – just one example of the redundancy in the system)

This will list all of the groups that you are in – which depending on how much use you have had at Diamond will either be a short list of the visits that you are named in on this course, or it could be a longer list of other visits that you have previously been involved in. This is because you automatically get put into the Linux group for visits that you are named on – either if you helped collect the data, or if you were given data access to it afterwards. So you should all find one of your groups is mx30951-8

However, this listing of groups doesn't tell you which beamline the visit was on, which is a problem with Diamond's data storage system, which is (as a reminder):

/dls/{beamline}/data/{year}/{visit number}

However, using wildcards, you can do a search:

Ls -d /dls/i*/data/2021/mx30951-8

And you will find the data that you need.

Practice

Now practice – try some different searches out

Clear

If your terminal becomes clogged up with all this practicing, then you can wipe the slate clean with:

clear

You will find everything has vanished, and you can start again. But not from the beginning, as if you do **pwd** again, you will find that you are still in the folder that you had just navigated to. You just don't have all the mess of switching around before this on display.

History

This command is particularly useful if someone has just shown you how to do something, and you need more time to make notes on it. Or if you have just used the **clear** command and want to check back on what you just did. Typing the command:

history

Produces a list of the history of commands on that session. Note that if you are working hard, you'll end up with a very long list of commands. Therefore, you can use:

history 10

This gives you the most recent 10 commands (you can use any number here, where I have used 10)

Remember you can also scroll through previous commands, one at the time, by pressing the up arrow (↑), as an alternative method.

Make a new folder/Directory

Initially experiment with this in your home directory, and then when you are confident, there is a task for you to go and add files and folders to the course folder. So, to make sure you are in your home directory, where you can make all the mess you want, simply use:

cd

And you will find you are back in your home directory.

The command to **m**a**k**e a new **dir**ectory is:

mkdir test

This will make a folder called "test" in whichever directory you are currently working in (remember **pwd** to **p**rint which **w**orking **d**irectory you are working in).

For people unused to Linux, it can be unnerving that nothing seems to happen at that point, but you can use the **ls** or **tree** command to see the folder "test" in the list. Alternatively, if you want a bit more feedback on what is going on, using the "**v**erbose" mode is very useful, and quicker. We get this by adding **-v** to the command, so let's make a second folder:

mkdir -v test2

The new folder will be created, and then a line will appear below saying "mkdir: created directory 'test2'" which can be reassuring. You can also use the **ls** command to show you what is currently there, if you wish.

Make a New File

We are going to work in these folders (still working in your home directory), so you want to move into it test:

cd test

Now we want to create a file, and in this instance, we are going to create a .txt file The easiest way to do this is using the **>** command, so try:

> test1.txt

Then of course, you want the **ls** command to see it in position, just to check that it has worked.

If you want to create more than one file at a time, you can use the **touch** command (it also works for only creating one file), so please input this:

touch test2.txt test3.txt

Then the lovely **ls** command should list all 3 files. At the moment they are empty, but they do exist.

You need to have write permissions to the file. Otherwise, you will receive a permission denied error. You will definitely have permission in your home directory, where you are currently practicing – but be aware of this when you start moving into visit folders.

Read & Write Content Into A File

Cat Command

You can also use the con**cat**enate command to write text to a file:

cat > test1.txt

And you will find that your prompt vanishes, and your cursor sits on the left-hand side. You can now type as much as you want and pressing **[Enter]** will simply send you to the next line. So, try adding some exciting text, maybe even write a poem (or just opt for "Hello World!"). Then when you have finished, press:

[Control] and **d** (at the same time)

And you will be back where you started with the Command Prompt, but the document will have content in it, if you read it. The way to read it is simply:

cat test1.txt (without the > sign)

Nano – Text Editor

Now we want to add some text to this file. There are various text editors that you can use, and for this example we will use one called **nano**. Therefore, the prompt is:

nano test2.txt

This opens a text editor window, into which you can type whatever text you want – so write anything you like. Commands within nano are at the bottom of the screen. Of particular note, to save the words you have written, you need to use **Ctrl** + **o** not Ctrl + S which might be your Windows reflex. Then to exit it is **Ctrl** + **x** which is more intuitive. Nano does query whether you want to save any changes you haven't already saved.

To read what is in a file using nano, then it's just the same **nano test1.txt** command from before to reopen the file.

Move Files

To **mov**e a file, you use the **mv** command, then say which file or directory you want to move, and then where you want to move it to. So in your test directory, create a new file named "filetomove":

> filetomove.txt

Then create a new directory to move it into:

mkdir newdirectory

Then to actually move it:

mv filetomove newdirectory

You should find that you can now move into newdirectory and then list the contents to see filetomove is now inside it.

Over-write Files

I have warned you about deleting files in my talk, as there is no undo and no recycle bin, and you can also delete files by overwriting them, without Linux giving you any warning at all. So firstly, check that there is text in **test1.txt** by reading it as above. Then recreate it with:

> test1.txt

And read is again, and you will find it is blank. You have been warned…

Delete files

When deleting files, it is really important that you remember that there is no undo, and no recycle bin. No going back. Hence, we have you practice in your home directory. As you have practiced, you will have created a whole load of files – now it's time to tidy up a bit…

You use the **rem**ove command, so to delete test1.txt put in:

rm test1.txt

Then use the **ls** command to check that it has gone.

If you want to remove multiple files at once, then try

rm test2.txt test3.txt

If you want a bit of clarity about what it is doing, you can use the **v**erbose setting, so:

rm -v test4.txt

Will then tell you "removed 'test3'"

Practice

I suggest you repeat this writing and reading process and deleting until you are comfortable with it.

Show Us You Can Do It

Now please go to this directory - /dls/i03/data/2021/mx30951-8/processing/testing

1. Create a directory within it, titled with your name

2. Create a text file in that directory, titled "confidence" containing a statement about how confident you are at Linux

If you have any difficulties with this, then let us know.

Permissions

There are different permissions to do things to different files, and if you want to know if you have permission to either read a file, or alter a file, you can do so by listing with -l:

ls -l

From the output:

- First character = file (-) or directory (d) (there are others, but I'll gloss over that)
- Next 3 characters – permission of the owners
    - r = read permission
    - w = write permission
    - x = execute permission
- Characters 5-7 – permission of the group
- Characters 8-10 – permission of others

The rest of the information tells you how big the file is and when it was last modified – and of course the filename.

Have an explore around to see what you do and don't have permission to see in beamline directories and so on.

Also, have a read of the ls manual as there are loads of different ways to list things, which will help you find them:

man ls

Module Load CCP4 Workshop

To create an environment that works within Diamond, for this course, it's important to enter:

module load ccp4-workshop

This will move you to a working folder /dls/temp/{your Fed ID}

Once you start running big jobs, you can access additional computing power by using:

clusterme

Which will then add even more power to your terminal – but don't worry, jobs are prioritised, so you can't use too much power here.

Loading GUIs

You can access various GUIs and programs from the command line. So for instance, to start up ADXV you type:

module load adxv

adxv

And the program will start in 3 separate windows. You can look at the data by navigating to:

/dls/i03/data/2021/mx30951-8/gw/20211118/TestInsulin

Pick a crystal to examine, for instance ins_1 and go into it's f

Shout in the Slack channel if you have more

Here is the full manual for using ADXV:

https://www.scripps.edu/tainer/arvai/adxv/AdxvUserManual.pdf

Access the Tutorials

Dials

The main DIALS tutorial will go through the GUI, but your can also do the same tutorial in the Command Line, and anyone who has already had experience of DIALS GUI may wish to try it this way instead. Using the Command Line will also enable more advanced use cases.

https://dials.github.io/documentation/tutorials/index.html

XDS

https://xds.mr.mpg.de/

https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Tutorial(First_Steps)

https://smb.slac.stanford.edu/facilities/software/xds/

SHARP/autoSHARP

https://www.globalphasing.com/sharp/wiki/

Bonus Extras

Simple things you can type into the Command Line to

cal – shows this month's calendar (and do **man cal** to find out more options on this command)

date – shows the current date and time

Finally

I want to add to this tutorial, so if you've got this far, please let me know how you find it (negative feedback is definitely sought as well as positive) then please email it to me – felicity.bertram@diamond.ac.uk