

Using the Linux Workstations

Mark Lunt

Contents

1	Preliminaries	1
1.1	Purposes	1
1.2	Installing Software on the Workstations	2
1.3	User Accounts	2
1.4	Software to Connect to the Workstation	2
2	Connecting to the Linux Workstation with Cygwin	3
2.1	Command-line access	3
2.2	Running Graphical Programs	3
2.3	Running a Window Manager	3
3	Connecting to the Linux Workstation with PuTTY and Exceed	4
3.1	Command-line access	4
3.2	Running Graphical Programs	5
3.3	Running a Window Manager	6
4	Using the Workstation	6
4.1	Mounting Drives	6
5	Performance	8
5.1	Stata Flavours	8

1 Preliminaries

1.1 Purposes

The basic purpose of the Linux workstations is to run analyses that would be either impossible or prohibitively time-consuming if they were run on your own computer. It would be counter-productive to run lots of relatively minor jobs on the workstations, since that would slow down the major jobs without saving much time on the minor jobs.

Currently, there are no plans to back up the workstations. Anything on the workstations should be considered temporary. Any data for analysis should be on either L, P or S drive: these can all be accessed from the workstation. You may want to copy a dataset to the workstation for processing, if that is more efficient than working with it across the network, but any scripts for processing the data and results that you want to keep should be stored on one of the network drives.

It is expected that some jobs on the workstations could be quite lengthy, but at the same time maintenance is required. Most updates can happen easily, but a new kernel requires the machine

to be rebooted. New kernels seem to come out every week or two, so the machine will need rebooting regularly. Currently, we plan for maintenance to take place on a Wednesday, so you can expect the machine to be rebooted then (although it will not be necessary *every* Wednesday). If you have a job that will still be running when maintenance is due, have a word with Owen: it may be possible to leave it running to completion, if he knows it is necessary.

1.2 Installing Software on the Workstations

The workstations run Red Hat Enterprise Linux (64-bit version), so any packages that are available for that OS can in theory be installed. Other software that is not specifically packaged for Red Hat can also be installed, although it may be a little trickier: discuss this with Owen. However, the machines are intended specifically for data analysis, so a request to install, for example, a music player would be unlikely to be granted. For an up-to-date list of software available, see the intranet.

Although a full set of development tools are available, making it possible to install software in your own home directory, this is probably best avoided. If you find the software useful, it is likely that others will as well, so it would be far better to have a copy installed centrally for everyone to share. The exception to this would be if you are writing / testing your own software: this is best kept to your own home area until it is working as intended, then you can ask Owen to install it for all users.

1.3 User Accounts

Before you can use the linux workstation, you need to have an account on it. Owen will arrange this: send him an email giving your Central Services Username. Also ensure that he knows that you need to access the command “`sudo mount`”, so that you can mount your P drive and L drive. He will let you know when your account has been created.

You will have your own home directory on the workstation, but with a limited quota of space. As has been said before, it would be better to store data and scripts on a network drive rather than on the workstation directly. However, it may make life easier to have a copy of C:/ado in \$HOME/ado.

1.4 Software to Connect to the Workstation

In order to connect to the workstation, you need to use `ssh`. This is available through cygwin, which is installed on the M drive. Alternatively, there is a free program called PuTTY which Owen can install on your PC if you wish. If you wish to use graphical programs, then you need an X-server. There are a number of possible solutions:

- Cygwin can provide an X-server, but it is not officially supported. It works at the moment, but that could change.
- The University has a site licence for Exceed, which is a commercial (and supported) X-server.
- If you have a linux virtual machine on a personal computer you are using to log into the workstation, you could use that as an X-server. This is a very heavyweight solution, and we have no intention of installing virtual machines on the Unit’s computers, but the appearance is better than the default cygwin X-server.

In addition to the software required, you need a hole in the firewall on your PC to enable it to communicate with the workstation. Owen will arrange this when setting up your account

2 Connecting to the Linux Workstation with Cygwin

Once you have your account, you can connect to the workstation. I will assume that you are using cygwin to make the connection.

2.1 Command-line access

To get command line access, you need to start a `bash` shell and run `ssh` in it. You need to run `M:/cygwin/cygwin.bat`, which will start the shell running (either navigate to it in Windows Explorer and double click, type it in at the Start \Rightarrow Run command-line, or (preferably) create a shortcut to it on your desktop). Then you can type

```
ssh mdeasUUU@130.88.236.140
```

replacing the “mdeasUUU” above with your username. You will then be prompted for your password, which you will have received from Owen. You will then be logged into the workstation.

You can type

```
pwd
```

to check: the response should be `/home/mdeasUUU`, where “mdeasUUU” again represents your username.

2.2 Running Graphical Programs

If you wish to run graphical programs on the workstation, you need to run an X-server on your own computer. You start this by running `M:/cygwin/usr/X11R6/bin/startxwin.bat` (again, a shortcut is recommended). This will open an xterm window, in which you can enter the command

```
ssh -Y mdeasUUU@130.88.236.140
```

The `-Y` tells `ssh` that you are running an X-server and that all X-windows information should be sent to it. You may wish to add the option `-C` as well: this compresses the data before it is sent, reducing the network usage and possibly speeding things up a little as a consequence. You will again be prompted for your password: enter it and you are logged on. To check, you can type

```
xstata &
```

and you should see stata opening on your desktop.

2.3 Running a Window Manager

Using the method outlined in section 3.2, each program run on the workstation appears as a new window on your PC, and there is no visual difference between programs running on your PC and programs running on the workstation. This can be confusing if you are running the same program (e.g. emacs) on both the PC and workstation. Also, all programs must be started from the command-line: you do not have access to the workstation desktop, with its shortcuts and menus.

The above gives you access to all programs on the workstation, but you have to start them from the command-line. It is possible to work with a full desktop environment as if you were sat at the machine, but be warned: it is desperately slow. Working from the command-line is far better when working remotely: graphically, you have to wait until the remote machine responds to your mouse movements, and it's easy to get ahead of yourself if you are not careful. Getting ahead of yourself when typing commands is not a problem.

Nonetheless, it is possible to run a window manager on the remote machine, and access your programs from menus rather than from the command-line. Another advantage of this is that

all the programs running on the workstation appear together: it can be confusing when they appear as windows on your windows PC, you forget which programs are running where.

So, to get a full managed session on the workstation, start a cygwin command line as before by running `M:/cygwin/cygwin.bat`. Within this window, type

```
xinit &
```

to start the X-Server¹ *without a window manager*. You will get a large ugly window, with a terminal window in the top left corner. In the terminal window, type

```
ssh -Y mdeasUUU@130.88.236.140
```

and enter your password when prompted. You are now connected to the workstation. To start the window manager, type either

```
startkde
```

or

```
gnome-session
```

depending on whether you prefer Gnome or KDE for your desktop environment. A number of error messages will appear in the terminal window, but eventually the window manager will start. You now have a single window on your PC which contains everything you would see if you were logged in directly at the workstation.

3 Connecting to the Linux Workstation with PuTTY and Exceed

3.1 Command-line access

Command line access is provided by PuTTY. The first time that you connect via PuTTY, you need to set it up. However, you can store all of the settings you need and load the configuration each time you wish to connect.

If you do not have a configuration saved, or this is the first time you have connected to the workstation, you can create one using these steps:

- (i). Start PuTTY
- (ii). On the first page you see (“Basic options for your PuTTY session”: you can return to this page by clicking on “Session” in the left hand pane), enter 130.88.236.140 in the “Host Name (or IP address)” textbox. Leave the Port at the default 22.
- (iii). Set the connection type to SSH
- (iv). Click on “Data” underneath “Connection” in the left pane (if “Data” is not visible, click on the + sign next to “Connection” and it will appear).
- (v). Put your username into the “Auto-login username” textbox
- (vi). Select “Default Settings” then click on “Save” to save this setting as your default. If you don’t want it as your default, enter another name in the “Saved Sessions” textbox before clicking Save

Now, you can click on the “Open” button, and you will open a terminal window on the workstation. In future, when you start PuTTY, these defaults will be entered for you (if you saved them as “Default Settings”, otherwise you will need to click on the name you gave this

¹If the response to `xinit` is `xinit: command not found`, you need to enter the command `PATH=/usr/bin/X11:$PATH;export PATH`

configuration and click “Load” before you click “Open”). You will be prompted for your password: enter the password that Owen gave you for the workstation and you should be connected. You can type

`pwd`

to check: the response should be `/home/mdeasUUU`, where “mdeasUUU” again represents your username.

3.2 Running Graphical Programs

To run graphical programs on the workstation and have them display on your PC, you need to configure PuTTY again, and also set up Exceed. To configure PuTTY:

- (i). Start PuTTY, and in the left hand pane, click on the + sign next to SSH (if “SSH” is not visible, click on the + sign next to “Connection” and it will appear).
- (ii). Click on “X11”
- (iii). Click in the checkbox labelled “Enable X11 Forwarding”
- (iv). Again, click on “Session” then “Save” to save this configuration, either as the default or under a different name, as you prefer.

To configure Exceed:

- (i). Go to Start ⇒ All Programs ⇒ Hummingbird ⇒ Exceed ⇒ Xconfig
- (ii). Under “Network and Communication”, ensure that the mode is set to “Passive”
- (iii). Under “Security . . .”, select the “File” radiobutton, leave the filename as `xhost.txt`, then click on Edit. This will open the file for editing: beneath the comments at the top, add a new line containing the single word `localhost`. Click on the big green tick to confirm the changes. This ensures that only users on your PC can access windows on your PC.
- (iv). Under “Display and Video”, ensure that the “Window Mode” is set to “Multiple” and the “Window Manager” is set to “Default to Native”

Now you should be able to run graphical programs on the workstation, with them appearing in a window on your PC. To test this:

- (i). Start Exceed (Start ⇒ All Programs ⇒ Hummingbird ⇒ Exceed ⇒ Exceed). A small window will appear in your taskbar.
- (ii). Start PuTTY and click on “Open”
- (iii). Enter your password on the workstation when prompted
- (iv). After logging on, type `xclock &` (without the inverted commas). You should see a clock appear in the top left of your screen.

Now, any X-based program you run (such as `xstata`) will appear as a separate window on your PC.

3.3 Running a Window Manager

If you wish to run a window manager on the workstation, as described in section 2.3, this is again possible using Exceed and PuTTY. However, there are some changes that need to be made to the Exceed configuration:

- (i). Go to Start \Rightarrow All Programs \Rightarrow Hummingbird \Rightarrow Exceed \Rightarrow Xconfig
- (ii). Go to “Display and Video”
- (iii). Change “Window Mode” from “Multiple” to “Single”
- (iv). If you wish, you can set the size of your workstation desktop using the “Root Size (pixels)” options.

Now, start Exceed as usual, and connect to the workstation with PuTTY as usual.

4 Using the Workstation

4.1 Mounting Drives

You need to get data to the workstation in order to be able to do any work. The easiest way to do this is to mount your P drive and the L drive. First, you need to make a space to put them: the commands to do this are

```
cd ~
mkdir pdrive
mkdir ldrive
```

The command to mount the P drive will be different for every person, but takes the form

```
sudo mount -t cifs //XXXXXX-XX.ds.man.ac.uk/HOME ~/pdrive -o
    username=mdeasUUU@ds.man.ac.uk,uid=NNN,gid=NNN}
```

Note that

- (i). This needs to be all entered on one line: the line break above is purely for typographic convenience.
- (ii). You need to replace “//XXXXXX-XX.ds.man.ac.uk/HOME” in the above command with the location of your P drive. You can find this out by going to The University IT Account Manager page² in a web browser, logging on, then clicking on “Accounts”.
- (iii). You need to replace “mdeasUUU” in the above command with your username.
- (iv). You need to replace NNN in the above command with your UID (User ID number) and GID (group ID number) on the linux workstation. You can find this out using the command

```
cat /etc/passwd | grep mdeasUUU
```

where “mdeasUUU” is again your username. The result will be a record containing 7 fields, separated by colons. The fields are
 - a) Your username mdeasUUU
 - b) an 'x', to show that you have a password set

²<https://iam.manchester.ac.uk/welcome/menu>

- c) UID
- d) GID
- e) Your full name
- f) Your home directory
- g) Your default shell (`/bin/bash`)

The command to mount the L drive is similar:

```
sudo mount -t cifs //ss3.ds.man.ac.uk/VOL5 ~/ldrive -o
username=ARCERU/mdeasUUU,uid=NNN,gid=NNN
```

Note that

- (i). Again, the command must be entered on one line.
- (ii). The location of the L Drive is the same for everybody (but which directories on the drive they can see is determined by their user name, so each person has to mount their own copy of the drive).
- (iii). Again, mdeasUUU and NNN need to be replaced with your username, UID and GID.

If you have access to the Welcome Trust Case Control Consortium drive, you need somewhere to mount it: `mkdir sdrive` will provide it. The command to mount is it similar to that for your P drive:

```
sudo mount -t cifs //vdm06-g1.ds.man.ac.uk/MHS_Research_01\$ ~/sdrive -o
username=mdeasUUU@ds.man.ac.uk,uid=NNN,gid=NNN}
```

To facilitate mounting the drives, there is are two scripts called `mount_drives` and `mount_drives_s` in `/usr/tmp`. You need to copy one of these scripts to your home directory (`mount_drives` if you do not have access to the S drive, `mount_drives_s` if you do) with the commands

```
cp /usr/tmp/mount_drives ~
```

or

```
cp /usr/tmp/mount_drives_s ~/mount_drives
```

then edit it to contain the correct username, home drive address, UID and GID. For editing, Emacs is available, but if you are not familiar with it, `gedit` may be simpler to use. Once it has been edited, it can be run with the command

```
~/mount_drives
```

Note that if you run the above commands, either separately or as a script, you will be prompted for two passwords:

- (i). Your password on the workstation, so you can access `sudo`
- (ii). Your password on the network, so you can access the drive.

Unfortunately, `sudo` sometimes asks for a password and sometimes does not, depending on how long it is since you last used it. And the password prompt is identical for `sudo` and `mount`, although they need different passwords. You are therefore strongly recommended to change the password that Owen gave you to the password you usually use on the network. The command to do this is `passwd`: you will be asked to enter your existing password, then enter your new password twice.

5 Performance

5.1 Stata Flavours

I've run a couple of analyses on my desktop computer, and on the different types of stata installed on the workstation. The BSRBR reports do-file mainly writes text files, the preparation do-file mainly does data manipulation, and the multiple imputation do-file is mainly doing data analysis. The results are in Table 1.

Stata Version	BSRBR Reports	Multiple Imputation	BSRBR (Prepare)
Desktop	700	149	145
stata	156	59	52
stata-se	172	59	55
stata-mp	145	38	79
xstata	211	79	59
xstata-se	223	77	56
xstata-mp	198	59	83

Table 1: Comparison of Timings Between Stata Flavours

Clearly, running jobs on the workstation is much faster than running them on a desktop machine. And stata-mp is usually faster but not dramatically faster, than stata or stata-se. However, the graphical versions of stata are consistently slower than the command line versions, so if you have a long program to run, run it as a batch job, i.e.

```
stata -b do my_do_file_name.do
```

Notice also that stata-mp was slower than stata or stata-se at preparing the BSRBR data. Although all analysis commands are quicker in stata-mp, there are some data management commands which are slower. In particular, `reshape` is slightly slower, and `joinby` is dramatically slower. Also, `bootstrap` is slower, but overall stata-mp *may* be quicker when bootstrapping depending on which command you are bootstrapping: you should profile your do-file to find out which stata runs it best.

This strengthens the argument that you should separate data preparation from data analysis in your do-files:

- Currently, the workstation cannot use ODBC, so it cannot extract data from databases. The do-file to do this must be run on a windows PC.
- If we could run `prepare.do` on the workstation, it would possibly run faster on stata or stata-se, rather than stata-mp, but `analyse.do` would be faster on stata-mp. Keeping them separate helps with efficiency.