Research Computing: Practical session 2

1 Basic Networked Linux usage

Try the following exercises:

1. ssh to a CSC server such as `apollo.lsc.phy.private.cam.ac.uk`.
2. Change your password.
3. Log out and back in again to make sure you've remembered your password.
4. Print the full directory name of your home directory.
5. List the current directory.
6. List `all` directories (including hidden ones) in your home directory.
7. Can you list the home directory of your neighbour? Or `pmb39`? If not, why not? (Hint: Use `ls` to check file-permissions.)
8. How much disk-space do you have in your home directory?
9. What disk-space is available in `/local/data`?
10. How much disk-space is available on the file-server `maia`?
11. Can you run `xclock` or `xeyes`?
12. Try to get one of these to run (if you are on Windows or Mac OS X and it doesn't work quickly, don't spend too much time on this).
13. Set up an ssh-key (with passphrase) from your laptop to `apollo`.
14. Try connecting to `cerberus1` instead. Do you need to enter your password? If not, why not?
15. What CPU(s) and how much memory does `cerberus1` have?
16. Run all the following commands in a `screen` session on `apollo`.
17. Copy any text file from your laptop to your CSC home directory.
18. Edit the file on `apollo` and copy it back to your laptop.
19. Print the value of the `PATH` variable.
20. What programs are running on `apollo`?
21. Who is logged in to `apollo` or who has logged in recently?
22. What is the default version of `g++` (C++ compiler)?
23. Can you find a different version in `/lsc/opt` and add it to your `PATH`?

2 Bash scripting

Create a Bash script that does the following:

1. Takes a single argument corresponding to a file suffix, e.g. `c` for files with suffix `.c`.
2. Finds all files in the current directory with that suffix
3. Displays the number of files to the user.
4. Makes a tar-ball of all of these files called `Files_c.tar.bz2` (for example).
5. You should include suitable error checking, and exit early from the script if necessary.

Now test it by checking out the source-code for `screen`:

```
git clone https://git.savannah.gnu.org/git/screen.git ./screen
cd screen

$ find_endings.sh c
There are 42 files with suffix .c

$ find_endings.sh
Please specify a single suffix as an argument to this script

$ find_endings.sh C
There are 0 files with suffix .C

$ find_endings.sh pl
There is 1 file with suffix .pl
```

Create a Bash script that displays some characteristics of all the machines listed as arguments. For example:

```
$ analyze_machines.sh melete lovelace cerberus1
========== melete ===========
Has 24 cores and 64038MB of memory.
There are currently 4 users, and 5 recent users
The 15min load-average is 0.07.
========== lovelace ===========
Has 64 cores and 257394MB of memory.
There are currently 2 users, and 13 recent users
The 15min load-average is 0.01.
========== cerberus1 ===========
Has 64 cores and 63859MB of memory.
There are currently 1 users, and 12 recent users
The 15min load-average is 32.23.
```