The Linux Command Line and Some Useful Keywords for our Workshop

Extended Notes for Statistical Computing on Hammer & LionX Machines

Research Computing and Cyberinfrastructure Group (RCC)
Pennsylvania State University

1 Instruction

In this extended note, you'll be introduced to a number of command line tools, technologies and terms that are useful to know for our workshops on Statistical Computing. My goal is to provide a more cohesive, up-to-date review to today's computing tools and prepare the attendees for the coming workshops. I'll provide references, links and keywords whenever possible and I hope you'll find it helpful.

2 An overview of useful terms

Redhat Linux: is an OS, one of the most popular Linux distributions. The Enterprise 6 is the license that is installed on most of our Clusters.[1]
Bash: bash is a reliable ubiquitous shell/terminal for typing in your commands. Right click on the Redhat Desktop and you can open one by selecting Terminal from the list.
Vi, vim, emacs or gedit (only on Hammer): these are Text Editors to provide you with an interactive computer programming environment. To use one, type in their name in a terminal:

[1]Most of the credit for the material should go to Dr. Balaji S. Srinivasan Lecturer at Stanford.

Listing 1: Run gedit on Hammer in the background

```
gedit  newfile.txt &
```

Virtual machine: In practice, while you are submitting your job onto our cluster you are usually not just using a single physical computer, but rather a virtualized piece of a multiprocessor computer. Breakthroughs in operating systems research have allowed us to take a single computer with (say) sixteen processors (e.g. LionXG clusters)and make it seem like sixteen independent computers, capable of being completely wiped and rebooted without affecting the others.
A good analogy for virtualization is sort of like taking one apartment complex with sixteen rooms and converting it into sixteen independent apartments, each of which has its own key and climate control, and can be restored to default settings without affecting the others. While these "virtual computers" have reasonable performance, they are less than that of the native hardware. In practice, virtualization is not magic; extremely high workloads on one of the other "tenants" in the VM system can affect your room, in the same way that a very loud next door neighbor can overwhelm the soundproofing- that is in interactive mode (think Hammer) [Not as true while you are on the batch mode (like LionX machines where resources are secured for when assigned you but still others load balance affect your queue time so the analogy still holds to some degree).

IP address: a dotted address representing a machine on the internet, e.g. 171.65.1.2. For example to get the IP address of the hammer, or check if they are live, open a terminal and type in:

Listing 2: Run ping on Hammer

```
ping hammer21.rcc.psu.edu
```

and you'll be checking on Hammer Unit 21 (we have 24 units).
Hostname: The name of a server, e.g. hammer.rcc.psu.edu or lionxj.rcc.psu.edu
DNS: The system which maps a hostname like hammer.rcc.psu.edu to an IP address like 171.65.1.2
SSH : Secure Shell, mentioned earlier, a means of connecting to a remote computer and execute commands.
Public Key: (If you have an account on our newer machines like cyberstar

you've heard about it) When you hand out your public key, it is like handing out a set of special envelopes. Anyone has your public key can put their message in one of these envelopes, seal it, and send it to you. But only you can open that message with your private key. Both of the "keys" in this case can be thought of as long hexadecimal numbers. Cyberstar asks for your public key to confirm your identity in future interactions; essentially for all future communications they send you a message that only you can open (with your private key). By decryption this message and sending it back to them, they know that you must have the private key and so they authorize the operation.

3 The Command Line

In general, the best way to understand the command line is to learn by doing. So, I suggest that you log into your Hammer Account, Open a terminal and type these into the terminal. You can use man + command name (e.g. man ls) to get the manual for that command.
Before showering you with the most used commands there are three key concepts about Unix based systems (like Linux Redhat which is our OS) that I'll go over very briefly. I suggest you consult with the following references to become reasonably skilled in the process of "learning by doing":

- Beginning the Linux Command Line, by Vugt and Sander van (online edition is available through PSU library)
- Sobells Linux Book (Applicable to both Linux and OS X systems, most popular)
- The Command Line Crash Course (free online tutorial introduction to the command line)

3.1 Concept One: Modular Design

Unix follows the Modular Design Philosophy: Everything in Unix is a module, the combination of all these modules developed/being developed by UNIX developers community has made it what it is now. Connecting this to our workshop on Intro to R, R started with the same design mentality

back in the S-days: everything is a module/library/package, parented and maintained by different developers until now.

3.2 Concept Two: The Three Streams

Most bash commands accept inputs as a single stream of bytes called STDIN or standard input and yields two output streams of bytes: STDOUT, which is the standard output, and STDERR which is the stream for errors. You can always manage and use these stream. #PBS -l oe basically takes care of your output and error streams when you submit a job in Batch.

- STDIN: this can be text or binary data streaming into the program, or keyboard input.
- STDOUT: this is the stream where the program writes its data. This is printed to the screen unless otherwise specified.
- STDERR: this is the stream where error messages are display. This is also printed to the screen unless otherwise specified.

3.3 Concept Three: Piping

It's a technique of connecting one command's STDOUT to another command's STDIN. For example:

Listing 3: piping

```
curl -s rcc.its.psu.edu | head -n 2 &> sample_piping.txt
```

what is done above, with the | symbol is piping. The | is called a pipe, and the technique of connecting one command's STDOUT to another command's STDIN is very powerful. It allows us to build up short-but-powerful programs by composing individual pieces. If you can write something this way, especially for text processing or data analysis, you almost always should...even if it's a complex ten step pipeline. The reason is that it will be incredibly fast and robust, due to the fact that the underlying GNU tools are written in C and have received countless hours of optimization and bug fixes. It will also not be that hard to understand: it's still a one-liner, albeit a long one. For more examples you can check here.

4 Some useful linux commands:

```
cd - change directory
alias - set a command alias to save typing
rm - remove a file (check options irf)
mv - move a file
mkdir - create a directory
pwd - print the current working directory
env - list all environment variables
ls - list files in current directory (check option -l )
ln - create symbolic links
rsync - synchronise a local file with a remote file
wget - download a file (only for publicly available files)
curl - Only for single URLs, understands more protocols
ping - test network availability
less - used to view large files by paging it
cat - File viewer, but does not have pagination features of less
head - view first few lines of a file (check option n)
tail - view last few lines of a file
cut - Pull out columns from a file
paste - paste data into columns
nl - print our the line number
sort - sort lines in a file
uniq - determine unique elements in a file
wc - line, word and character count
split - split large files
man - single page manual files for commands
info - provide more details, examples, etc than what man provides
uname - lists system information
hostname - name of machine
whoami - name of current user
ps - list current running processes
kill - kill a process (check option -9)
top - list processes based on criteria
sudo - act as root user for one or more commands
su - become root user
tar - archival utility
```

```
gzip - compression utility
find - non-indexed file search
locate - indexed file search. Requires updatedb
df - determine disk space
du - determine file's disk usage
grep is a text and file parser that uses regular expressions.
sed is a string substitution command. Used to do a find and replace.
awk is a useful scripting language for tab-delimited text.

A list of useful bash shortcuts:

CTRL+C : Abort command
CTRL+L : Clear the screen (command clear will do something similar)
CTRL+D : Exit the command prompt
CTRL + Windows Button:  it takes you to your local machine
Two important Operators:
Ampersand & allows you to run a command in the background
Backticks ` allows you to use STDOUT from commands
```

References

[1] RCC webiste: rcc.its.psu.edu