

CS2042 - Unix Tools

Fall 2010

Lecture 6

Hussam Abu-Libdeh

September 20, 2010

ssh

You can use “secure shell” (ssh) to connect to a remote machine.

```
ssh [username@]<remote machine name or IP address>
```

- If the username is omitted, local username will be used.
- Remote machine has to be configured to accept ssh connections:
 - ssh daemon (service) has to be running and listening on an open port (by default 22)

Executing remote commands

ssh can be used to execute commands on the remote machine

Example

```
ssh ha232@csug01.csuglab.cornell.edu ls
```

this will execute `ls` on `csug01.csuglab.cornell.edu` and output the result to the screen before `ssh` terminates the connection

- You can use the `-f` flag to put `ssh` into the background before executing the remote command
- You can use the `-Y` flag to forward X11 (graphical windows/user interface) to the local machine

Run firefox on the remote machine

```
ssh -Y ha232@csug01.csuglab.cornell.edu firefox
```

- You can use an identity file to authenticate with the remote machine instead of using your username/password.
- An identity file allows you to authenticate yourself with a “*pass phrase*” (which could be empty).
- Identity files are typically a pair of public/private keys used for asymmetric key cryptography (e.g., RSA).

Identity files

To use identity files,

- 1 Create an identity file using `ssh-keygen`

create identity files using RSA encryption

```
ssh-keygen -t rsa
```

- 2 Append the generated public key file (by default `~/.ssh/id_rsa.pub`) to the `~/.ssh/authorized_keys` file on the remote machine.
- 3 `ssh` to the remote machine using the `-i` flag to use the identity file, and specify the private file corresponding to the public file appended at the remote machine

```
ssh hussam@example.com -i ~/.ssh/id_rsa
```

ssh configuration file

You can configure ssh to use customized settings when connecting to a particular host without having to set the corresponding flags every time. The file `~/.ssh/config` contains these settings.

Sample config

```
host office
hostname mishmish.cs.cornell.edu

host tiger
hostname tiger.cs.cornell.edu
user hussam
ForwardX11 yes
IdentityFile ~/.ssh/id_rsa
```

Here, ssh office connects to mishmish.cs.cornell.edu and ssh tiger connects to tiger.cs.cornell.edu with username hussam and identity `~/.ssh/id_rsa` and enable X11 forwarding.

sftp

- Transfer files securely between local and remote machines.
- Operates over an encrypted ssh transport.
 - same connection settings as ssh
- Uses an interactive console to interact with the user
 - unless the `-b [batchfile]` option is used to use batch files
- Useful sftp commands:
 - `help` : to see a list of commands and help on them
 - `put` : upload a file to the remote machine
 - `get` : download a file from the remote machine
 - `cd` / `pwd` : change directory / print current directory on remote machine
 - `lcd` / `lpwd` : change directory / print current directory on local machine

scp

- Copy files securely over a network using an encrypted ssh transport.

copy file to remote machine

```
scp file hussam@remote_machine:
```

copy file from remote machine

```
scp hussam@remote_machine:file .
```

- The ':' is necessary after the remote machine name. A path on the remote machine starting from the user's home directory can be specified after the colon ':':

copy directories using the -r flag

```
scp -r pics_dir hussam@remote_machine:
```


So far we have been only talking about text processing. However that is not all what we can do and automate! For example, we can use tools to automate graph plotting (gnuplot) or manipulating images (the ImageMagick suite).

gnuplot

`gnuplot` is a command-line program that can generate 2D and 3D plots of functions, data, and data fits. It is frequently used for publication quality graphics as well as educational purposes. It runs on all major operating systems (GNU/Linux, Unix, Windows, Mac OS X, and others).

- Run `gnuplot` with no command line arguments to get an interactive console.
- Run `gnuplot plot_script` to execute the plotting commands from the given script file.

Important gnuplot commands

- `help` : great help and documentation for the many gnuplot features and commands
- `plot` : plot data or a function

Plot the function `sin(x)`

```
plot sin(x)
```

- `set xrange / yrange` : set the range on the x or y axis.

```
set xrange [0:5]
```

- `set xlabel / ylabel` : set the text label on the x or y axis

```
set xlabel 'Time (s)'
```

- `set title` : set the graph title

```
set title 'Awesome performance graph!'
```

Plotting options

- `gnuplot` can read data files and automatically separate data from multiple columns

Plot data using the first and second column from a file

```
plot "data.csv" using 1:2
```

- Plot multiple data series

```
plot "data.csv" using 1:2, "data.csv" using 1:3
```

- Plot the series using a line

```
plot "data.csv" using 1:2 with lines
```

- Plot the series using boxes

```
plot "data.csv" using 1:2 with boxes
```

There are many things that gnuplot can do. To learn more about a particular command or function, use the built in help (which is searchable).

To learn more about the plot command

```
help plot
```

- Many very useful gnuplot tips here:
<http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>
- gnuplot project homepage:
<http://www.gnuplot.info/>