General Linux Tips and Tricks – Revision 1 (08/30/06)

1 Using Linux

1.1 Basic Commands

`ls` List the contents of a directory.

`cd <dir>` Change the current directory.

`pwd` Print the current directory.

`less <file>` View a text file.

`xemacs <file> &` Start the xemacs editor in a separate window on a file.

`xemacs -nw <file>` Start the xemacs editor in the current window on a file.

`man <prog name>` Get help on a program, output will come up in a pager.

`rm <file name>` Remove a file.

1.2 Using a pager

Here is a list of basic commands to use the common pagers `less` and `more`.

`<space>` Advance to next screen full of information.

`b` Go back one screen full of information.

`q` Quit pager and return to shell.

`<return>` Advance one line.

`/<regular expression>` Search for next instance of the regular expression. You can search for strings containing only letters, numbers and underscores simply by using the string as the regular expression.

1.3 Using a Shell

These instructions assume that you are using bash. You can check this by typing the command `echo $SHELL`.

To set an environment variable, the command is `export <var-name>=<value>`. This sets the value for the current shell.

To set an environment variable for all new shells that you create, add the line `export <var-name>=<value>` to the file ~/.bashrc

To avoid typing the same command over again, you can use up and down arrow to cycle through the previous commands you have entered.

You can type `<tab>` in many situations at the shell to complete things for you or get a list of possible completions in case it is ambiguous.

1.3.1 UNIX Paths

Unix paths all start at the root directory (/). The current directory is indicated by period (.). The parent of the current directory is indicated by two periods (..). The components of a path are separated by forward slashes (/).

1.4 Using Xemacs or Emacs

Type text using the keyboard. Use the arrow keys and such to move around.

Type `C-x C-s` to save the current file (or use the button or menu options). `C-x` means press the "Ctrl" key, press the "x" key, release the "x" key, then release the "Ctrl" key.

Type `C-x C-c` to quit xemacs.

1.5 Using ssh/scp

We do not use `telnet` or `ftp` on our machines because they are insecure protocols (unless using kerberized version) which send passwords and other sensitive information in cleartext. Instead, we use the ssh tools `ssh` and `scp`.

`ssh` is a secure replacement for `telnet` that also securely forwards X11 requests. This allows access to remote shells and the ability to run X11 programs on the remote machine with the display sent to the local machine. To connect to a remote machine use `ssh [<user>@]<machine name>`. The user name can be omitted if it the same on both machines. The brackets indicate an optional part of the command.

`scp` is a secure alternative to `ftp`. `scp` uses a syntax similar to `cp`. To specify a remote file use the syntax `[<user>@]<machine>:<filename>`. The user name can be omitted if it is the same on both machines. The filename is a path relative to your home directory unless it starts with a slash (in which case it is relative to the root directory). The rest of the syntax is the same as `cp`. See `man scp` for more details.

1.5.1 Using Public/Private Key Authentication

Normally `ssh` and `scp` authentic by asking for your password. You can set up ssh to use public/private key authentication instead. This creates a trust relationship between the machines, so please only use this between machines within the lab. Here are the steps needed to setup public/private key authentication so that remote machine trusts the local machine:

1. Open two xterms.

2. Connect to the remote machine using `ssh` in one window.

3. Run `ssh-keygen -t rsa` on the local machine (you only have to do this once per machine). It's best to set a password when prompted. This password is for securing your private key on that machine so that someone else could not use it even if they gained access to the machine.

4. On the local machine, run `scp .ssh/id_rsa.pub user@remote:` You'll still need to type your login password for the remote machine.

5. On the remote machine type `cat id_rsa.pub >> .ssh/authorized_keys`

6. Now you can ssh into the remote machine and you will need to type your SSH password instead of the login password for that machine.

7. Now, create a file in the directory .kde/Autostart called ssh-add and put the following contents into it:

   ```
   #!/bin/sh
   /usr/bin/ssh-add < /dev/null > /dev/null
   ```

   and make it executable with the command: `chmod 700 .kde/Autostart/ssh-add`

8. Execute the script for the first time with the command: `.kde/Autostart/ssh-add` It will ask you for your SSH password with a graphical prompt.

9. You should now be able to ssh to the remote machine from the local machine without entering a password. ssh-add will now be invoked whenever you log into the local machine and will cache the password for the remainder of the session.