**zLinux Basics on System z**

**Objectives:**

* Understand how z/VM exploits virtualization and allows Linux to be run as a virtual guest
* Be able to effectively work in the Linux environment
* Be able to do essential system tasks
* Be able to work with files including creation, editing, coping, etc.
* Be able to write basic Linux scripts

Background: <http://www.perpetualpc.net/srtd_commands_rev.html>; Mod1\_Intro.ppt

**Accessing Linux on the Mainframe**

1. Access the Secure Shell client on your desktop:

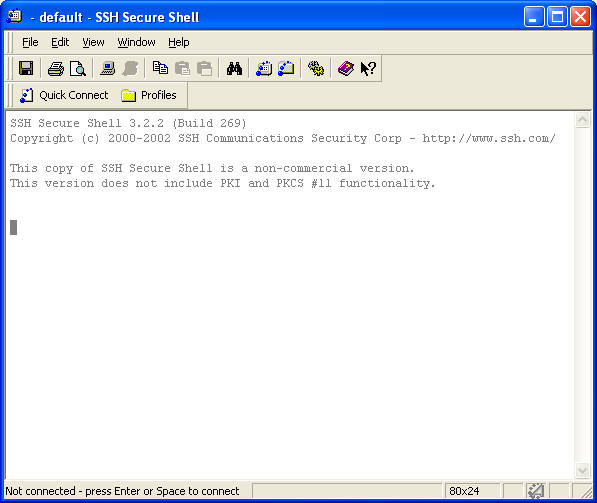
Click on Start

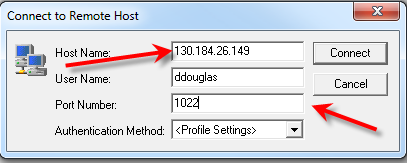
Click on All Programs

Click on SSH Secure Shell

Click on Secure Shell Client

The following window will appear:



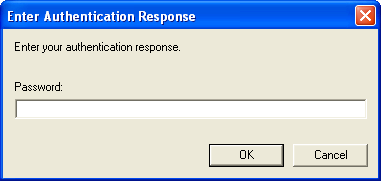
Click on Quick Connect. The following window will appear.

On campus, enter 130.184.26.149 as the Host Name. Enter your user id—***must be lower case.***

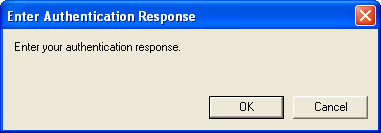
Off campus, you may need to enter the host of comp.uark.edu and your user id and the port must be 1022.

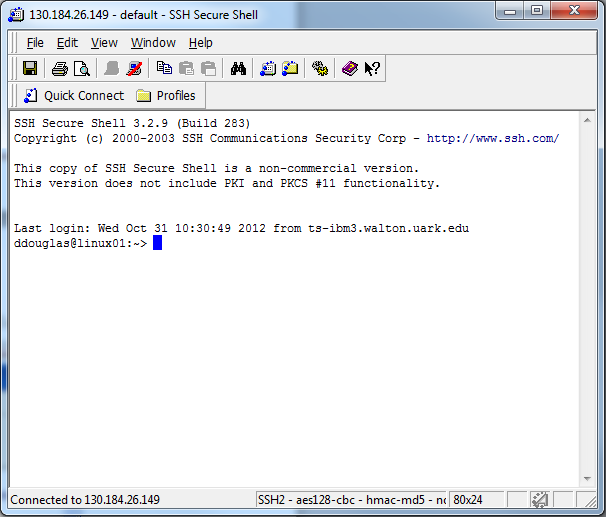
Click on Connect.

Enter the password “Initial1” (that’s the word Initial followed by a 1) and click OK



You will see this box. Click OK to open the default shell (bash)





Command line prompt

Type passwd and press enter to change your password.

To exit Linux, type exit and press return. Close the SSH Secure Shell window.

**LINUX Tour—Part I**

1. Type **date**. What shows on the screen?
2. Type **w.** What shows on the screen?
3. Type **who**. What shows on the screen?
4. Type **who –q**. What shows on the screen?
5. Type **finger**. What shows on the screen?
6. Type **man cal**. What happened? This command displays the manual or help pages for the cal command. Use **Shift-Q** to exit the man command.
7. Type **cal**. What shows on the screen?
8. Type **cal 2011**. What shows on the screen?
9. Type **cal 12 2012**. What shows on the screen?
10. Display the calendar for October 2012. What command did you use to do that?
11. Look up the month and year you were born – on what day of the week were you born?
12. You can use LINUX to send mail. Try this first to yourself and then to someone else.
    1. Type mail and then your user name or someone else’s name.
    2. Enter the subject, this is a test
    3. Enter hi this is a test. Enter as many lines as you want.
    4. Press Ctrl-D to send. You will see EOT and will be returned to the prompt.
13. To check your mail:
    1. Type mail and then press enter
    2. If you have 1 message, then enter the number 1 and press enter.
    3. To reply, press r and then return.
    4. Press Ctrl-D to send.
    5. To delete, press d and then the number.
    6. Press Ctrl-D to exit the mail environment.
14. Note: If you get stuck in mail, press Shift-Q and you will receive a question mark prompt. Type exit and press enter to exit mail.
15. Type **clear** and press enter. What happened? Is it really gone?
16. Type **ls –l**. (Both are the letter l and not the number one). What do you see?
17. Type **cal > test**. Press the enter key. The > symbol moves the results of the command into a text file. In this case, the text file is named test.

Type **cat test**

1. Type **cal >> test** and press the enter key. The >> symbol appends the results of the command into the designated text file—named test in this example.
2. Press the up arrow twice. What do you see? (should see ls –l). Press enter.  
   Why are some files in blue and others are not in blue?
3. Type **more test**. What happens?
4. Type **cat test**. What happens? The cat command is used to display the contents of a file. It is derived from the term concatenate which means to join or fuse together.

Type **cal 2008>>test** Type **cal 2009>>test**

Type **cat test**

Type **cat test |more** Type **cat test|less**

1. Type **jpico** and press enter. You are in an editor. Type a few lines and then press **Ctrl-X** (This is designated as ^X at the bottom of the editor). Type **Y** to save and give it a name of **testfile**.
2. What would we type to see the contents of the file we just created?
3. What would we type to edit the file we just created? Try it. Add a new line and then save. Examine the contents of the file – is the new line there?
4. Type **cat test testfile**. What happened? Is it permanent? This is an example of using the cat command to concatenate two files.
5. Type **pwd**. What shows on the screen?
6. To move up one level (to the parent) subdirectory, type **cd..** and press enter.
7. Type **pwd**. What shows on the screen? This is the full pathname of the current working directory.
8. Type **cd..** to move back up one more level (change to the parent directory of the current directory). On what level are you located? cd is the command that is used to change directories.
9. Type **ls –l** to see all of the files.
10. Type **echo “This is fun”** and press the enter key.
11. Type **echo “Enter text here”** and press the enter key.
12. Type **echo “This is so much fun” \!** and press the enter key.

Type **echo “This is so much fun!”** and press the enter key.

1. Type **echo –e “Enter text here \c”** and press the enter key.   
   -e enables character and \c suppresses the trailing new line.
2. Type **echo $HOME** What shows on the screen?
3. Move back to your home directory**: cd /home/userid** where userid is your userid. Type **pwd** to make sure you are in your home directory.
4. Type **cd /** Be sure to include a space between the cd and the /. Where are you now?
5. Move back to your home directory: **cd ~** Type **pwd** to make sure you are in your home directory.
6. Type **ls –l.** Do you see your two test files?
7. Type **du** and press enter to see the disk usage.
8. Type **top** and press enter to receive information about the CPU – the most intensive processes currently running. This is frequently refreshed (measured in seconds). To stop this, Press Ctrl-C.
9. Type **free** and press the enter key to display statistics about memory usage (RAM and swap) and buffers used by the kernel. free –t displays totals.
10. grep (general expression) is used to search in file(s) for file lines that match a regular expression pattern. The syntax is **grep pattern files**. For example, if you named the file you created in step 17 test, then the command **grep Tu test** would tell Linux to search the file test for lines that have the pattern Tu. Try it and see what happens.
11. Use the editor to add Tu to the file you created in step 19 called testfile.
12. type grep Tu \* and press enter. What happened?

Remember to enter Exit to leave Linux!!!

**LINUX Tour - II**

Follow these steps to create a subdirectory under your home directory and copy the two test files (test and testfile) into it:

1. First, **type pwd** to make sure you are in your home directory. You should see /home/userid
2. To create the subdirectory junkfiles under your home directory, type **mkdir junkfiles**
3. Type **ls –l** to see that the subdirectory was created. Notice it is in blue!
4. To copy a file, the cp command is used. (To move a file, the mv command is used.) We are already in the subdirectory containing the files we want to copy and the destination subdirectory is under the subdirectory we are in, so typing **cp test junkfiles** copies the file test into the subdirectory junkfiles.
5. To copy the file testfile using the complete path for both source and destination subdirectories, the command is:

**cp /home/userid/testfile /home/userid/junkfiles**

1. Go to the subdirectory junkfiles and type ls (**cd junkfiles** and then enter **ls**). Do you see the two files there?
2. Go back one subdirectory (**cd..**)
3. Type **pwd** to make sure you are in your home directory.
4. Use **mv** to move a file. Note that is also can be used to rename a file; for example

**mv** test testa

changes the name of the file, test, to testa. Of course, you can move a file to another location while keeping retaining the original file name.

1. To remove a file, the rm command is used. To remove the file named test, type **rm test**. Remove the file testfile also. If we had used move instead of copy, would we have to remove these files?
2. Type **ls**. Are the two files gone?
3. To remove a subdirectory, the command rmdir is used. Type **rmdir junkfiles**. What happens? Can you figure out what to do? (see below)
   1. **cd junkfiles**
   2. **rm test**
   3. **rm testfile**
   4. **cd ..**
   5. **rmdir junkfiles**

**OR rm –r junkfiles**

**OR rm junkfiles -r**

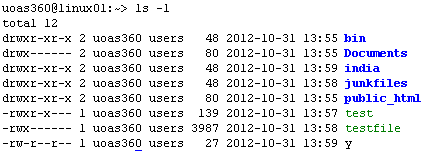
**The ls –l command and Permissions in Linux**

When the ls –l command is entered, a long listing (-l) of the directory is listed.

The output contains the following information:

* Type of file (This is the first character and specifies if the file is a directory, file, character device, or block device)
* Permissions (The next 9 characters in three groups of three. Discussed below)
* Links column – the number of links or shortcuts to the file
* Owner column – the ID of the person who owns the file.
* Group column – the name of the group
* Size in bytes
* Date and time the file was created or last edited
* Linux filename – can be a maximum of 256 characters long.

An example of the output is below:



The first character is the type field. Possible values include:

d – file is a directory

- – file is a file

c – file is a character device

b – file is a block device.

**Permissions:**

After the first character for the type field, the next nine characters represent the permissions for the file. They are grouped into three groups of three:

1. The first three represent the owner’s permissions
2. The second set of three represent the permissions for the group, and
3. The last three represent the permissions for other users.

The characters used in defining permissions and what they mean are:

r – file can be read

w – file can be written to

x – file can be executed if a file. If a directory, x means that the ls command can be performed to see its contents.

- - permission is not granted.

Example:

-rwxr-x--- 1 uoas360 users 139 2012-10-31 13:57 test

-rwx------ 1 uoas360 users 3987 2012-10-31 13:58 testfile

Notice the two file listed above.

The first character of “-“ indicates that they are both files.

The next three characters (rwx) represent the permissions for the owner. In both cases, the owner can read both files, write to the files, and execute the files.

The next three characters represent the permissions for the group. Examine the file named test. The characters are r-x indicating that members of the group named users can read and execute the file named test, but cannot write to the file named test. With --- as the last three characters of permissions on files means the members cannot read, write to, or execute these files; the outside world cannot read, write to, or execute them.

Changing Permissions:

The chmod command is used to change permissions of a file. Its format is:

chmod nnn filename

where each n represents a number from 0 to 7. The first n is the permission number for the owner, the second n is the permission number for the group, and the third n is the permission number for other users. The table below shows the permissions (read, write, and/or execute) and numbers to which they refer:

|  |  |
| --- | --- |
| **Permissions** | **Number** |
| --- (no permissions) | 0 |
| --x (execute only) | 1 |
| -w- (write only) | 2 |
| -wx | 3 |
| r-- | 4 |
| r-x | 5 |
| rw- | 6 |
| rwx | 7 |

If a file named program1 needed to have read, write and execute for the user owner, read and execute permissions for the group, and no permissions for all others, you would type:

chmod 750 program1

APPENDIX I

**Key combinations in Bash** (from <http://tille.garrels.be/training/tldp/ch02s02.html>)

| **Key or key combination** | **Function** |
| --- | --- |
| **Ctrl**+**A** | Move cursor to the beginning of the command line. |
| **Ctrl**+**C** | End a running program and return the prompt, see [Chapter 4, *Processes*](http://tille.garrels.be/training/tldp/ch04.html). |
| **Ctrl**+**D** | Log out of the current shell session, equal to typing **exit** or **logout**. |
| **Ctrl**+**E** | Move cursor to the end of the command line. |
| **Ctrl**+**H** | Generate backspace character. |
| **Ctrl**+**L** | Clear this terminal. |
| **Ctrl**+**R** | Search command history, see [the section called “The grep command”](http://tille.garrels.be/training/tldp/ch03s03.html#sect_03_03_03_04). |
| **Ctrl**+**Z** | Suspend a program, see [Chapter 4, *Processes*](http://tille.garrels.be/training/tldp/ch04.html). |
| **ArrowLeft** and **ArrowRight** | Move the cursor one place to the left or right on the command line, so that you can insert characters at other places than just at the beginning and the end. |
| **ArrowUp** and **ArrowDown** | Browse history. Go to the line that you want to repeat, edit details if necessary, and press **Enter** to save time. |
| **Shift**+**PageUp** and **Shift**+**PageDown** | Browse terminal buffer (to see text that has "scrolled off" the screen). |
| **Tab** | Command or filename completion; when multiple choices are possible, the system will either signal with an audio or visual bell, or, if too many choices are possible, ask you if you want to see them all. |
| **Tab** **Tab** | Shows file or command completion possibilities. |

**Introduction to Linux**

**A Hands on Guide**

[**http://tille.garrels.be/training/tldp/index.html**](http://tille.garrels.be/training/tldp/index.html)

**The Linux Documentation Project**

[**http://tldp.org/**](http://tldp.org/)