INFO I211 - Information Infrastructure II

Spring 2014

Lab 1: connecting to `burrow` SoIC Linux servers using `ssh`

This lab's purpose is to familiarize with using `ssh` to connect to the School of Informatics and Computing Linux systems.

Important Documentation from the SoIC Help Desk

Read the following School of Informatics and Computing Knowledge Base documents about available Linux systems:

- SoIC Help - What Linux systems are available?
- SoIC Help - Available Systems - Remote Use

Important Documentation from the UITS IU Knowledge Base

Read the following University Information Technology Services (UITS) at Indiana University - Knowledge Base documents about `ssh` connections and `ssh` software:

- At IU, what SSH/SFTP clients are supported and where can I get them? There are two main categories of ssh-based connections we're going to be using:
    1. secure-shell (ssh) terminal connections for interactive sessions, e.g. to write and run Python programs on remote Linux servers. Typically, we'll use PuTTY on Windows systems and OpenSSH on Mac OS X systems.
    2. secure-copy (scp) file-tranfer connections to transfer files to and from remote Linux servers, e.g. programs that we wrote on our desktop/laptop systems, etc.
    For more details, read the document:
- UITS IU Knowledge Base - What are SSH and SSH2?

Tasks

We suggest you do the following:

1. Use **PuTTY** or OpenSSH to login to the main Burrow Linux system. All enrolled I211 students should have a Burrow account activated now. The DNS name for the central Burrow server is silo.soic.indiana.edu.
2. Use **`WinSCP`** or `scp` to test uploading a local file to your account on the Burrow server (e.g. a local text file), and to test downloading a remote file from your accout on the Burrow server to the local filesystem.
3. **CGI accounts**: Make sure your *burrow* account is set up to run CGI scripts on the SoIC web server. Follow the School of Informatics and Computing Knowledge Base document on this topic: "How do I run CGI scripts on the SoIC web server?", then log into one of the Linux systems (like silo.soic.indiana.edu) and run the `make-cgi` script.

4. **CGI directory**: Once you are logged into silo.soic.indiana.edu, verify that the permissions set by the *access mode bits* for the directory where CGI scripts are located, i.e. the `~/cgi-pub/` directory, allow execution access to the CGI server. For this, type the following commands on the remote connection:

```
cd ~/cgi-pub/
chmod ugo+x .
```

5. **Create a test HTML page**: Create a simple file named `hello.html` in your `cgi-pub` directory, for example using the **pico** text editor:

```
cd ~/cgi-pub/
pico hello.html
```

   where the content of the hello.html file could be something like this:

```
<html><head><title>Hello HTML</title></head>
<body>
    Hello HTML, from I211!
</body></html>
```

6. **HTML file permissions**: Once you saved the `hello.html` file from the pico text editor, exit the pico text editor and set the correct file permissions on the `hello.html` file, so that it can be read by the web server program:

```
chmod ugo+r hello.html
pigi
```

7. **HTML web page test**: In a web browser window, go to the following URL address to verify that your `hello.html` page is now visible to the world (use your own username, not the word "username" in the address):

```
http://cgi.soic.indiana.edu/~username/hello.html
```

Last updated: *Mitja Hmeljak* January 16, 2014