

Goals for This Lecture:

- Understand how to log in remotely with SSH
- Understand how to transfer files with SCP
- Learn how to log in from SINC site windows PC's with XWIN32
- Understand the long form listings produced by the ls command
- Understand the basics of directory permissions
- Understand how to change directory permissions using the chmod command

Logging in remotely with SSH

- It is often highly desirable for you to login to a different Unix system from the one that you are currently logged into. Let's examine how to do this.
- Avoid using commands like **telnet**, **rsh**, and **rlogin**. They are no longer secure. These commands send your username and password over the network in an unencrypted fashion. Using them is asking for trouble. Someone can easily steal your username and password.
- Use an encrypted method such as secure shell, a.k.a. **SSH**
- **SSH** uses public key encryption to make sure that your username and password are sent over the network in a secure fashion
- A very good alternative is the secure shell command **ssh**

SSH usage

- The syntax for this command is easy:

```
>ssh -l username remote_machine_name
```

- You can execute your commands as you normally would
- When you are ready to log off of the remote machine just execute the command **logout**
- If you have problems connecting with **ssh** try **ssh1** (same syntax) instead. The remote machine may not support level 2 **SSH**
- The Matlab public machine addresses are listed in the **/etc/hosts** file on any matlab machine:
 - ella.mathlab.sunysb.edu
 - bird.mathlab.sunysb.edu
 - duke.mathlab.sunysb.edu

Copying files with SCP

- Copying of files from one machine to another can be accomplished using the SSH **scp** command
- Syntax:
>**scp user1@machine1:frompath user2@machine2:topath**
- This syntax can be simplified if either the "from" file or the "to" file are located on the current machine
- Example:
>**scp speech.doc gbush@whitehouse.gov:/home/gbush/speech1.doc**
- Example:
>**scp bclinton@whitehouse.gov:speech2.doc speech3.doc**
- You will be prompted for your password on the remote machine

ASCII & Binary files

- Files on Unix systems can consist of two basic types of data:
- ASCII data -- bytes of data containing codes recognized as alpha-numeric characters
- Binary data, -- bytes of data which are not recognized as alpha-numeric characters
- ASCII = American Standard Code for Information Interchange
- Example of ASCII files: text files, HTML files, files containing FORTRAN code
- Example of Binary files: files containing compiled executable code, image files, some data files
- You can find out what kind of file you are dealing with using the **file** command.
- Try using the file command on the various files in your directories.

Long form Listing Fields

- `>ls -al`
total 44
drwxr-xr-x 3 dswesty users 4096 Feb 11 20:32 ./
drwxr-xr-x 18 dswesty users 4096 Feb 6 22:00 ../
-rwxr-xr-x 1 dswesty users 23918 Nov 26 16:28 ftest
-rw-r--r-- 1 dswesty users 163 Nov 26 16:28 ftest.f
-rw-r----- 1 dswesty users 152 Nov 26 16:24 ftest.f~
drwxr-xr-x 2 dswesty users 4096 Feb 11 20:32 zcode/

- The second field is the number of hard links associated with the listed object (not important)
- The third field is the owner of the listed object
- The fourth field lists the group associated with the file
- The fifth field lists the size of the object in bytes
- The sixth field list the date of the last modification of the file
- The seventh field lists the time of the last modification of the file
- The eighth field lists the name of the file or directory

Directory permissions

- When you do an `ls -l` command in your current working directory you see a long form listing that looks like:

- ```
>ls -al
total 44
drwxr-xr-x 3 dswesty users 4096 Feb 11 20:32 ./
drwxr-xr-x 18 dswesty users 4096 Feb 6 22:00 ../
-rwxr-xr-x 1 dswesty users 23918 Nov 26 16:28 ftest
-rw-r--r-- 1 dswesty users 163 Nov 26 16:28 ftest.f
-rw-r----- 1 dswesty users 152 Nov 26 16:24 ftest.f~
drwxr-xr-x 2 dswesty users 4096 Feb 11 20:32 zcode/
```

- The output data from this long form listing contains numerous fields that list properties of the files and directories
- A "d" in column one means that the listed object is a directory
- Columns 2-10 list the permissions associated with each listed object

# Directory permissions

- `>ls -al`

```
total 44
drwxr-xr-x 3 dswesty users 4096 Feb 11 20:32 ./
drwxr-xr-x 18 dswesty users 4096 Feb 6 22:00 ../
-rwxr-xr-x 1 dswesty users 23918 Nov 26 16:28 ftest
-rw-r--r-- 1 dswesty users 163 Nov 26 16:28 ftest.f
-rw-r----- 1 dswesty users 152 Nov 26 16:24 ftest.f~
drwxr-xr-x 2 dswesty users 4096 Feb 11 20:32 zcode/
```

  - There are three basic types of permission associated with a file: read **r**, write **w**, and execute **x** permissions
  - The permissions come in three sets: for the user who owns the file (the first set), the group (second set), and the other (the third set)
  - Columns 2-4 list the permissions for the owner
  - Columns 5-7 list the permissions for the group
  - Columns 8-10 list the permissions for the world
- The file owner can change permissions using the **chmod** command
- Example: `>chmod og-rwx /home_b1/dswesty` removes all permissions for the group (**g**) and others (**o**)
- Directories must have both read and execute permissions in order for them to be fully accessible