

INTERNATIONAL UNIVERSITY OF JAPAN
Public Management and Policy Analysis Program
Graduate School of International Relations

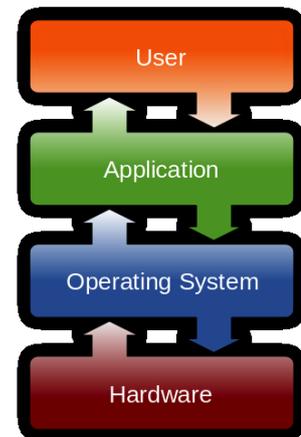
ADC5030401 (2 Credits)
Introduction to Electronic Government
Winter 2019

This document introduces UNIX and Secure Shell (SSH) to those who are new to UNIX, remote connection (telnet), and file transfer.

1. Operating Systems, Kernel, and Shell

An operating system (OS) is a core component of the system software of a computer system. An operating system manages computer hardware resources and mediates between computer hardware and (application) programs. Specifically it manages process and programs, memory (allocation), file systems, input/output devices, network, security (access control), and interface with users.

Kernel controls most basic processes and functions of a operating system, while a *user interface* (or *shell*) allows users to interact with (request to and get services from) the computer. A shell can be independent of kernel of the operating system (In Microsoft Windows, kernel and shell are integrated or inseparable). A command-line interface is a classic shell but graphic user interfaces (GUI) due to its intuitiveness becomes popular. Its use of *icons* in GUI was pioneered by Alan Kay at Xerox PARC (Palo Alto Research Center) in California. GUI examples include X-Window System or X11 (1984), Microsoft Windows (1985), KDE (1996), and GNOME (1999). (The above image was downloaded from Wikipedia)



2. UNIX History

UNIX is a multitasking, multi-user computer operating system developed in 1969 by a group of AT&T employees at Bell Labs who also developed a computer language C (Wikipedia). It was written in assembly language and then C in 1973. The Novell and then Open Group owns its trademark. UNIX has influenced other operating systems so significantly that some people say that UNIX is the only original operating system (other operating systems are just UNIX's variants). The cornerstone was *UNIX System V* that was commercialized by AT&T in the 1950s and later triggered the GNU project led by Richard Stallman in 1983. See http://www.unix.org/what_is_unix/history_timeline.html for the detailed UNIX history.

3. UNIX-Like

UNIX has evolved into various UNIX-like variants (including free UNIX) that interact and help each other to improve themselves.

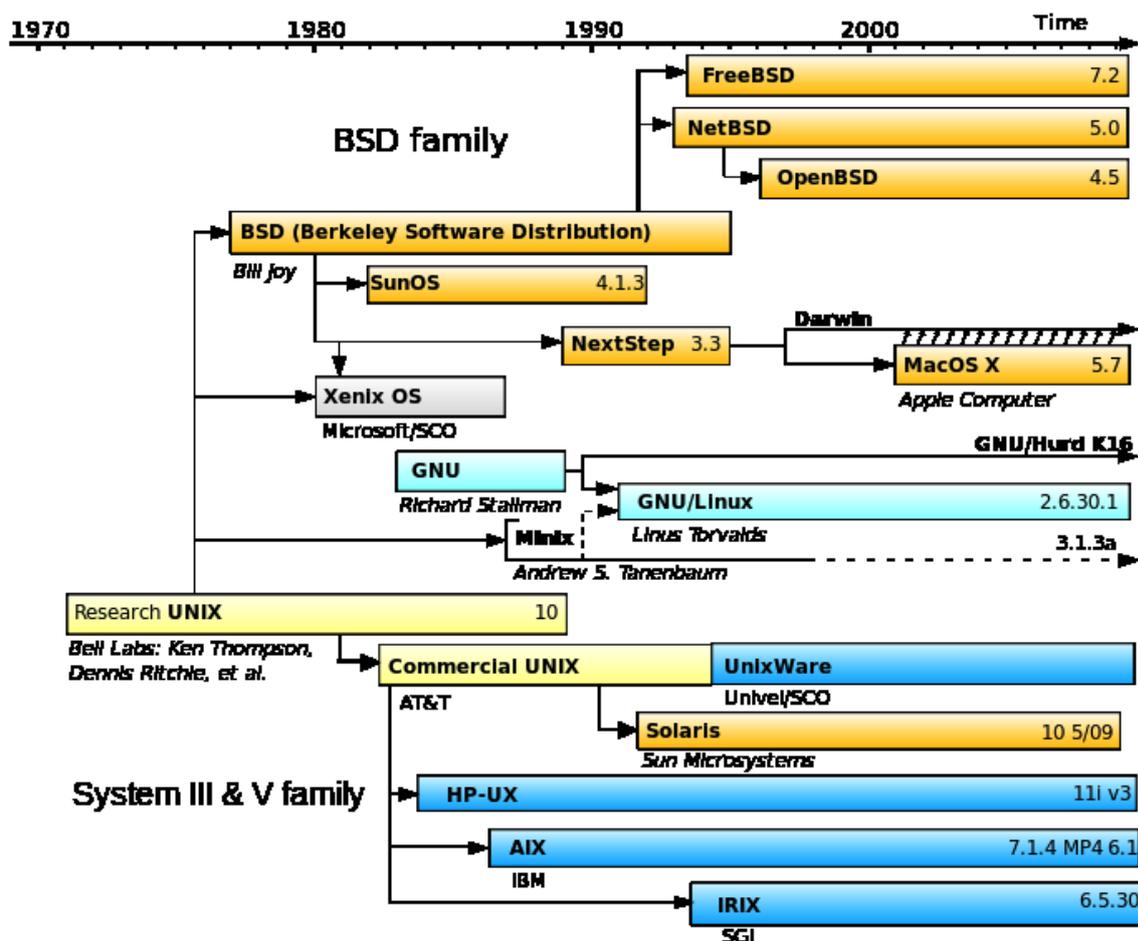
AIX, HP-UX, and IRIX are descendent of UNIX System V and run exclusively on IBM (International Business Machines Co.), HP (Hewlett-Packard), and SGI (Silicon Graphics Inc.) hardware, respectively.

BSD (Berkeley Software Distribution) family was introduced by the University of California at Berkeley and includes NetBSD, FreeBSD, and OpenBSD. Steve Jobs developed NeXTSTEP on the basis of BSD at his NeXT Inc. Tim Berners-Lee developed World Wide Web on a NeXTSTEP machine.

MacOS X (not previous releases) is a UNIX-based graphic user interface operating system that was developed from Steve Jobs' NeXTSTEP (OpenStep API).

Linux (kernel) was developed by Linus Torvalds in 1991. GNU/Linux has been under GNU General Public License and used by many Linux distributors such as Red Hat, Fedora, SUSE, openSUSE, Debian, Ubuntu, Knoppix, Slackware, Mandriva, and Gentoo.

SunOS/**Solaris** was developed by Sun Microsystems and becomes now Oracle Solaris. Solaris was a piece of proprietary software that supports primarily Sun's SPARC workstations.



Source: Wikipedia ("Linux")

Microsoft Windows is a proprietary operating system that supports Intel x86 architectures. Its earlier releases until Windows 98 were running on the top of MS-DOS before Windows

2000 was built on the NT (New Technology) technology. MS-DOS was originated (renamed) from Q-DOS (Quick and Dirty Operating System) or 86-DOS of Seattle Computer Products.

4. Understanding UNIX Command-line Interface or Shell

UNIX shell as an interface interprets your instruction or a set of instructions (computer program or computer software) so that you can tell your computer what you want it to do. A shell translates what you want into what computer can understand. UNIX has such shell as C-Shell (csh), Korn Shell (ksh), Bourne Shell (bsh), and Bourne Again Shell (bash) and each shell has different set of commands. You can choose the interpreter (shell) that can best support your task. The Linux Ubuntu server by default provides bash for users.

Each shell has its own *prompt* (e.g., \$ and %), a cursor, that waits for you type in UNIX commands. Of course, you may change the prompt to include current path and directory. See <http://kb.iu.edu/data/agvf.html> for UNIX shell.

In order to execute a command, type in a UNIX command and then hit **Enter**. A shell interprets the command and tells the computer to execute it. A UNIX instruction consists of a command, arguments, and/or options that were separated by spaces (at least one space is needed to separate two items).

For instance, the following command shows your working directory. Note that \$ below is a UNIX prompt; you don't need to type in.

```
$ pwd
```

The following command lists files stored in the current directory. You may provide options (e.g., -al below) that begins with -.

```
$ ls -al
```

A UNIX command may be followed by relevant arguments. The first command copies an existing file `unix.txt` into a new file `linux.txt` and the second to the directory `/home/kucc625/it` with the same name.

```
$ cp unix.txt linux.txt  
$ cp unix.txt /home/kucc625/it
```

You may type in more than one UNIX command in a line. Use semicolon (;) to separate commands as if you are writing C codes.

```
$ ls -al; cp unix.txt linux.txt
```

UNIX has its built-in command manual to be evoked by the `man` command. The following command pops up a help screen (definition, usage, options, etc.) for the command `ls`.

```
$ man ls
```

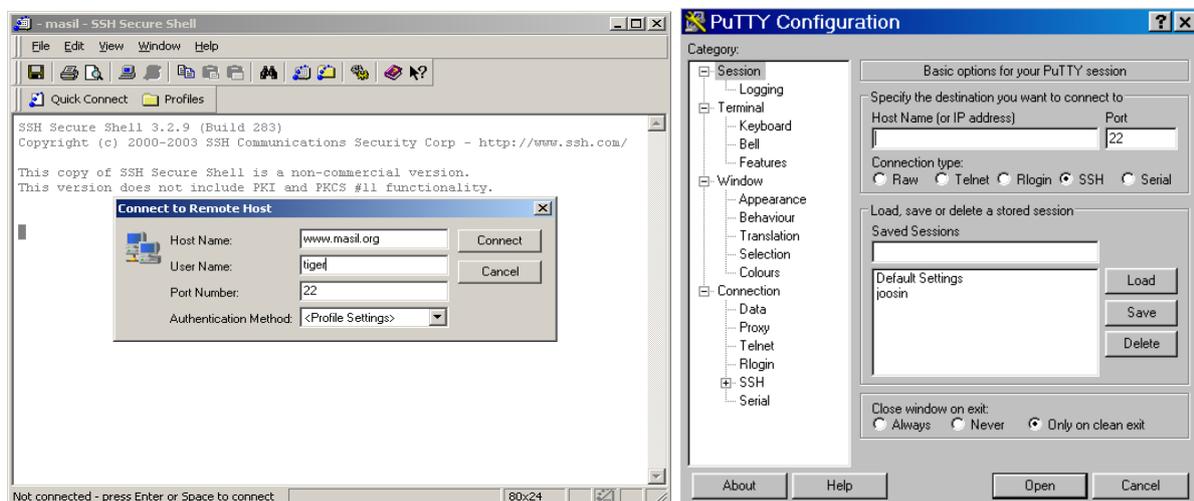
If you want to stop the current process and get UNIX prompt back, press **Ctrl+C** (holding **Control** key and then press **C**). When you finish job on the session (terminal or telnet), run `exit` to quit the session.

\$ exit

5. Accessing a Linux Machine: Secure Shell

Assuming there is a well built Linux server (Web and database server), how do you access it? Current Linux supports secured telnet only (classical telnet is not supported due to its vulnerability). The *telnet* is one of Internet services and a network protocol that allows you to access command-line interface on a remote machine (Linux server in this case). You can control the remote machine, once you can successfully access the machine using (secured) telnet.

In a Linux or Mac machine, you just need to run the secured shell command (`$ ssh ...`) because Linux and Mac are UNIX-like! If you are using Microsoft Windows, install SSH client (<http://www.ssh.com/>) from <https://shareware.unc.edu/> or PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>).



Once secured telnet client is ready, check the following access information to be used by telnet.

- IP address or domain name of the remote machine
- Account name (user ID) in the remote machine
- Password or passphrase of the network account
- Port number (by default, 22 for telnet)
- Any special option that the remote machine requires

The Linux server has your account name (nickname) and password (ask the instructor). This remote machine requires SSH connection; (classical) telnet is blocked.

Now, run Secure Shell Client (Click **Start** and then browser to the SSH Client) and click **Quick Connect** to open the **Connect to Remote Host** dialog box. Type in IP address to **Host Name:** and account name to **User Name:** and then click **Connect**. You will be asked to type in password (passphrase) shortly. See the left figure above.

Or run PuTTY from the Windows menu (Click **Start** and then browser to the PuTTY) to open PuTTY window. Provide IP address and then click **Open**. Provide your account name

after `login as:` and then hit **Enter** to type in its password (passphrase). See the right figure above.

If you provide correct account name and its password, you will get the UNIX prompt on the PuTTY window. Then you are ready to use a Linux server.

6. Basic Commands

There are several commands to check where you are (e.g., `pwd`) and show current status (e.g., `who`, `ifconfig`, `ps`, `du`, `df`). First, try to run `who` to find out who else is logged on this server.

```
$ who
```

Try to run `ifconfig` to list network configuration.

```
$ ifconfig
```

Run `ps` to list the current processes that are running on the server. The option `-ef` or `-aux` provides more detailed information in the standard format.

```
$ ps
$ ps -ef
```

Run `du` and `df` to display disk usage and free disk space of the file systems.

```
$ du
...
$ df
Filesystem                1K-blocks    Used Available Use% Mounted on
/dev/mapper/joosin-root 133066088 1790824 124515840   2% /
udev                      3918012      4    3918008   1% /dev
tmpfs                     1584028      904    1583124   1% /run
none                       5120         0         5120    0% /run/lock
none                       3960068      0    3960068   0% /run/shm
/dev/sda1                  233191     48216    172534   22% /boot
```

Run `date` to check current date and time. The `cal` command displays a calendar.

```
$ date
$ cal -y 2014
...
$ cal -m 2
```

```
    February 2014
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28
```

If you need a calculator, run the command and then provide mathematical expression. In order to quit the calculator, type in `quit` and the hit **Enter**.

```
$ bc
```

The following command clears the terminal screen and locates the prompt at the top of the screen.

```
$ clear
```

7. Navigating UNIX File System

UNIX has a quite efficient hierarchical file system beginning from the *root directory*. UNIX strictly regulates locations of files stored in its file system; you may not save your documents in any place other than your account home directory (`/home/your_account/`).

In UNIX, there is no drive name like `C:/` and `D:/`. UNIX has a file system that can combine multiple different types of secondary devices (e.g., EIDE, SCSC) into one integrated entity (In fact, current Linux server uses one EIDE HDD and one SCSC HDD).

The following `ls` command (“show all files and directories located at the root in the long format”) lists default directories at the root. `/` and `~` denote the root directory and user’s home directory.

```
kucc625@joosin:~$ ls -l /
drwxr-xr-x  2 root root  4096 Feb  3 18:32 bin
drwxr-xr-x  4 root root  1024 Feb  3 18:42 boot
drwxr-xr-x 16 root root  4340 Feb  6 17:03 dev
drwxr-xr-x 96 root root  4096 Feb  6 17:03 etc
drwxr-xr-x 11 root root  4096 Feb  6 16:00 home
drwxr-xr-x 18 root root  4096 Feb  3 18:32 lib
drwxr-xr-x  2 root root  4096 Feb  3 18:29 lib64
drwxr-xr-x  3 root root  4096 Feb  3 18:12 media
drwxr-xr-x  2 root root  4096 Aug  4 2012 mnt
drwxr-xr-x  2 root root  4096 Feb  3 18:10 opt
dr-xr-xr-x 118 root root    0 Feb  6 17:02 proc
drwx-----  3 root root  4096 Feb  3 18:26 root
drwxr-xr-x 18 root root   620 Feb  6 21:21 run
drwxr-xr-x  2 root root 12288 Feb  3 18:42 sbin
drwxr-xr-x  2 root root  4096 Mar  6 2012 selinux
drwxr-xr-x  2 root root  4096 Feb  3 18:10 srv
drwxr-xr-x 13 root root    0 Feb  6 17:02 sys
...
```

UNIX’s major directories are:

- `/bin/`: UNIX’s executable files are located
- `/boot/`: booting files
- `/dev/`: Devices (e.g., HDD, USB)
- `/etc/`: Configuration files (e.g., network)
- `/home/`: Users’ accounts and their home directories
- `/lib/`: Shared library files
- `/sbin/`: Binary files for system administrator (e.g., `ifconfig`)
- `/usr/`: Important programs and online manual (e.g., X Window System and C/C++)
- `/var/`: Variable-length files (e.g., log file)

Use a period to indicate current directory and two periods for the *parent directory* (one directory up from the current directory). Therefore, the following two equivalent `ls` commands, for example, list contents in the *subdirectory* or *child directory* `web/policy`, two

directories down from the current directory. The option `-al` displays hidden files in the long format. The last command lists all files that have the extension `doc`.

```
$ ls ./web/policy -al
$ ls web/policy -al
$ ls web/*.doc
```

The `cd` command changes from current directory (location) to another. When you login the Linux server, your default location is `/home/account` (account means your account name). Consider the following examples.

```
$ cd /
$ cd ../
$ cd ../../
$ cd ../kucc625/web
$ cd /etc/network
```

The first command changes your location to the root directory; The second goes to the parent directory of `/home/account` or simply `/home`; The third leads to the root directory; The first and third gives you the same result in the server. The fourth command first goes up to the parent directory (`/home`) and then gets down to its subdirectory `/home/kucc625/web`. The final command directly jumps to the *absolute path* (as opposed to relative path) `/etc/network`.

The `ls` and `cd` commands are most frequently used by users to navigate and view file systems.

8. Working with Files and Directories

It is very important to name files and directories properly. UNIX and Linux is casesensitive. Consider the following guidelines.

- 1) Use characters (a through z and A through Z), numbers (0 though 9), or underscore `_`. Do not use special characters, such as space, `-`, `$`, `#`, `@`, `&`, and `~`.
- 2) Begin with a letter.
- 3) Avoid reserved words or keywords, such as `cp`, `mv`, `cd`, `ls`, `cat`, etc.
- 4) Give something meaningful to indicate what the variable is for.
- 5) The shorter the better, although UNIX allows up to 256 characters
- 6) Use lower cases unless necessary or required.
- 7) Use group names so that you can take advantage of wildcards (e.g., `score?` and `score1-score9`).

	Good	Bad, If not Invalid
1)	<code>gnp_2002</code>	<code>gnp of 2002</code> ; <code>gnp-2002</code> ; <code>gnp#2002</code> ; <code>gnp~2002</code>
2)	<code>score1</code> ; <code>gnp_2003</code>	<code>1st_score</code> ; <code>2003_gnp</code>
2)	<code>interest</code>	<code>_interest</code>
3)	<code>gender</code> ; <code>education</code>	<code>double</code> ; <code>int</code> ; <code>using</code> ; <code>logit</code> ; <code>glm</code> ; <code>ttest</code> ; <code>tabulate</code>
4)	<code>invest_2003</code>	<code>x</code> ; <code>y</code> ; <code>z</code> ; <code>xxx</code> ; <code>yyy</code> ; <code>zmdje</code> ; <code>ej93nx6</code>
5)	<code>iRate_2003</code>	<code>real_interest_rate_of_in_2003</code>
6)	<code>income</code> ; <code>sales_IBM</code>	<code>INCOME</code> ; <code>InCoMe</code> ; <code>sales_ibm</code>
6)	<code>iRate_US2003</code>	<code>irate_us2003</code> ; <code>IRATE_US2003</code>
7)	<code>score1</code> ; <code>score2</code> ; <code>score3...</code>	<code>math</code> , <code>physics</code> , <code>history</code> , <code>management...</code>

You may copy files into new files using the `cp` command. Wildcards `?` and `*` are used to indicate any single character and number, and any characters and numbers, respectively.

```
$ cp unix.txt linux.txt
$ cp u*.txt notes/unix
$ cp *.* notes
```

The first command above copies `unix.txt` to `linux.txt`. Once the command is executed, you **MAY NOT** cancel it in UNIX; therefore, be cautious whenever you copy and remove files. The second copies all text files whose names begin with `u` (including, for example, `us.txt` and `unix1.txt`) to a subdirectory `notes/unix` under the current directory. The final command copies all files to a subdirectory `notes`.

If you want to rename or move files, use the `mv` command as follows. The second command below move `unix.txt` to a subdirectory `notes`. You may ignore `./` if the directory `notes` exists; If not, UNIX consider `notes` as a file rather than a directory.

```
$ mv unixx.txt unix.txt
$ mv unix.txt ./notes
```

The `rm` command below removes the files with filename `windows` regardless of their extension.

```
$ rm windows.*
```

If you want to create or remove a directory, use `mkdir` and `rmdir` commands, respectively. Note that `rm` removes files, not directories.

```
$ mkdir oss
$ rmdir oss
```

If you want to combine text files, use `cat` with `>` operator for redirection. The following command combines three text files and save the result into `linux.txt`.

```
$ cat chapter1.txt chapter2.txt chapter3.txt > linux.txt
```

9. Understanding File Ownership and Permission

In UNIX, all files and directories have their owner, user-group, and permission specified. These pieces of information tell you who own the file (or directory), in which user-group the owner is, and to what extent that the owner, user-group, and other users can handle the file (directory). Let us take a look at the following.

```
drwxr-xr-x  2 kucc625 root  4096  Aug 14 12:36 images
-rw-r--r--  1 kucc625 root 10090  Dec  3 15:09 index.html
```

The first part (called *mode field*) represents the permission. The second represents the number of memory block taken by the file (or directory). The third and fourth are owner and user-group followed by file size in bytes, date and time of last modification, and file (directory) name. For instance, the file `index.html` was owned by `kucc625` in user-group `root`. It was last modified on December 3 to have a file size of 10,090 bytes.

The mode field consists of four parts. The first letter indicates if the item is a directory (d) or a file (-). The a set of three letters tells owner's right to handle the file (directory), whereas the next two sets of three letters are for user-group and other users. *w* and *r* mean write and read, respectively, while *x* says that the file is executable. For instance, owner's permission *rw**x* below says that the user (owner) can read, write, and execute the file. The user group can read and execute the file but cannot write the file. Other users can only read the file.

d	rw	x	r	-	x	r	-	-
	User		User-group		Other users			

User and user-group can be changed by `chown` and `chgroup` commands by a *superuser* (root user). File permission can be changed by `chmod`. You may use `=` to indicate users (owner), and `+` and `-` to add or remove the permission. Look at the following examples.

```
$ chmod u=rw,g=r,o=r index.html
$ chmod u+x,g+w,o-r index.html
```

The first command gives the most common permission type used for Web documents, `-rw-r--r--`, meaning that the owner can read and write the document, and user-group and other users (e.g., Web site visitors) can read the file only. The second command adds execution permission to the owner and writing to the user group, and removes reading permission from other users.

File permission is summarized by three digits (e.g., 644 and 755) for the sake of convenience. The first digit summarizes owner's permission, while the second and third digits represent user-group and other users' permissions. 4, 2, 1 are respectively assigned to read (*r*), write (*w*), and execute (*x*). For instance, the permission `-rw-r--r--` is presented by 644. And 755 says that the owner can read, write, and execute the file ($7=4+2+1$), while user-group and other users can read and execute the file ($5 = 4 + 0 + 1$). Let us change the file permission as follows.

```
$ chmod 644 index.html
$ chmod 644 *.html
```

10. Working with a Text File: nano and cat

UNIX supports text editors such as `vi`, `vim`, `pico`, `emacs`, and `nano`. `vi` is the classical text editor provided by UNIX/Linux but is not easy for beginners (or Microsoft Windows users) to use. Ubuntu by default supports `nano`. At the Linux prompt, type in `nano` and file name to be created or modified.

```
$ nano linux.txt
```

Then, the `nano` editor pops up for you. You may see major short-cuts at the bottom of the editor. When you finish editing, press `Ctrl+X` (holding **Control** key and then press **X**) to save the change you made.

If you want to view the context of a text file, use the `cat` command followed by the file name. If you want to display the contents in the paginated form, use `more` instead.

```
$ cat sftp.txt
```

```
$ more sftp.txt
```

The `head` and `tail` commands respectively display the first and last several lines of the file.

```
$ head login.php
$ tail domain.pl
```

The `wc` command displays the numbers of lines, words, and characters used in a file.

```
$ wc egov.html
```

11. Piping and Searching

UNIX allows connects multiple commands using `|` (vertical bar) so that the output of a command becomes the input of the next command. This process is called *piping*. For example, if the list of files of a directory is very long, you can add `more` to display the list page by page.

```
$ ls -al | more
```

If you want to find a particular string in a text file, use the `grep` command after a command such as `ls` or `cat`.

```
$ ls -al | grep "linux.*"
$ cat linux.txt | grep "kernel"
$ ps -ef | grep "mysql"
```

If you want to find files, use the `find` command with proper options.

```
$ find . -name "*.html"
$ find /home -size +10M
```

The first command above lists all files in the current directory and subdirectories whose extension is `html`. The second displays the list of all files under `/home` whose file size is greater than 10 Mega bytes.

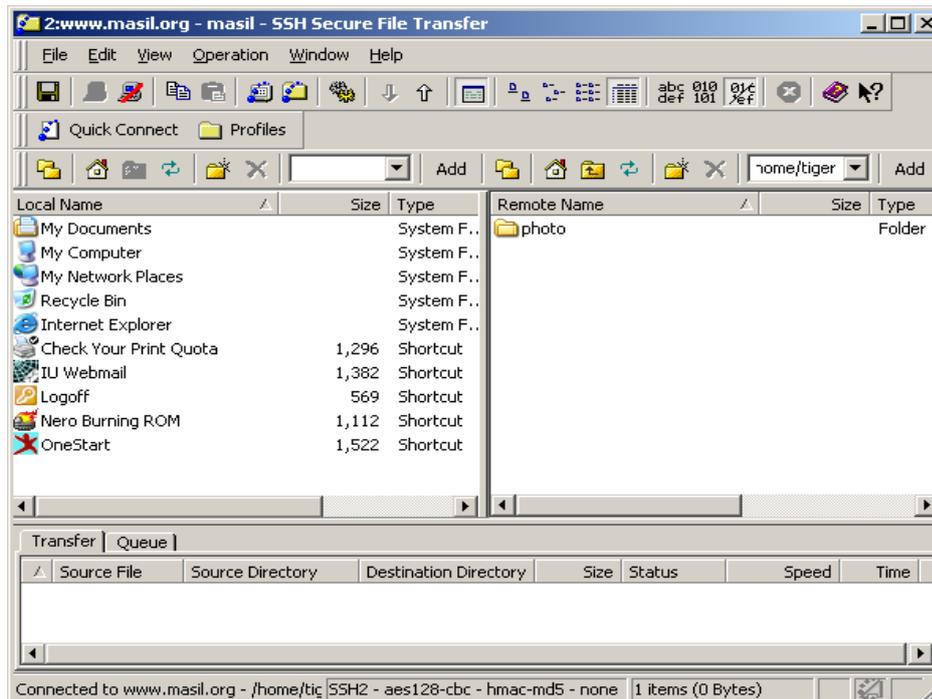
12. Uploading and Downloading Files: Secured FTP

When your Web document is ready, you need to send it to your account in the Web server. This process is called *uploading*. Since the traditional command-line file transfer protocol (*FTP*) is neither convenient nor secured, the secured FTP applications are used nowadays. Some of examples are SSH Client and WinSCP (<http://winscp.net/>) that provide a GUI environment for you.

SSH integrates secured telnet and FTP. One you make a remote connection, you can switch from one to the other. Look at the two icons  right below the menu bar; Left icon is for telnet, and right icon is for FTP.

Let us click the right icon to switch mode from secure telnet to FTP. SSH Secure File Transfer will pop up. Left files and directories shown under **Local Name** are what you have in your computer, while right files and directory under **Remote Name** are what are stored in the remote host computer. You may upload and download by dragging files from the left to

right and vice versa. You may navigate directories, copy files and directories, create directories, and remove files and directories as you do in Windows Explorer.



Visit following Web pages to learn how to use WinSCP.

- <http://winscp.net/eng/docs/screenshots>
- http://www.siteground.com/tutorials/ssh/ssh_winscp.htm
- <http://webmaster.iu.edu/tools-and-guides/publishing-to-webserve/winscp.phtml>

Once you install such applications, you need to provide server connection information as you did in secured telnet above. In SSH, secured telnet and FTP are integrated. Once you make a FTP connection, you can send and receive (download) files as if you are using the Windows Explorer.

13. Accessing Your Account on joosin.org

Your UNIX account is already created on the server (joosin.org), which is located in the MLIC server room. Your account name is your nickname such as indika and mist. Therefore, provide joosin.org (host name), your nickname (account or user name), and 22 (default port number) when connecting to the remote Web and database server.

Your Web documents need to be placed under www that was created for you. For instance, when mist logs on the machine, he will find the following files under the directory www. index.html is the default HTML document that the Web server recognizes.

```
mist@joosin:/home/mist/www$ ls -al
drwxrwxr-x 2 mist mist 4096 Feb  6 21:53 .
drwxr-xr-x 4 mist mist 4096 Feb  6 17:17 ..
-rw-r--r-- 1 mist mist 3630 Feb  2 22:51 index.html
-rw-r--r-- 1 mist mist 16578 Feb  2 22:51 kcm.jpg
```

End of this document.

<http://www.sonsoo.org>