香港城市大學
City University of Hong Kong

Lab 1: Getting Started with Linux

CS4296/CS5296
Cloud Computing

專業 創新 胸懷全球
Professional · Creative
For The World

Some slides are adapted from tutorial slides of Augustine Abaris at Boston University: Introduction to Linux (Available at here).

What is Linux?

- **Linux** (/ˈlɪnəks/ LIN-əks) is a family of open source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds.
- Linux is typically packaged in a Linux distribution. Distributions include the Linux kernel and supporting system software and libraries.
- Popular Linux distributions include Debian, Red Hat, SUSE, Fedora, and **Ubuntu**.

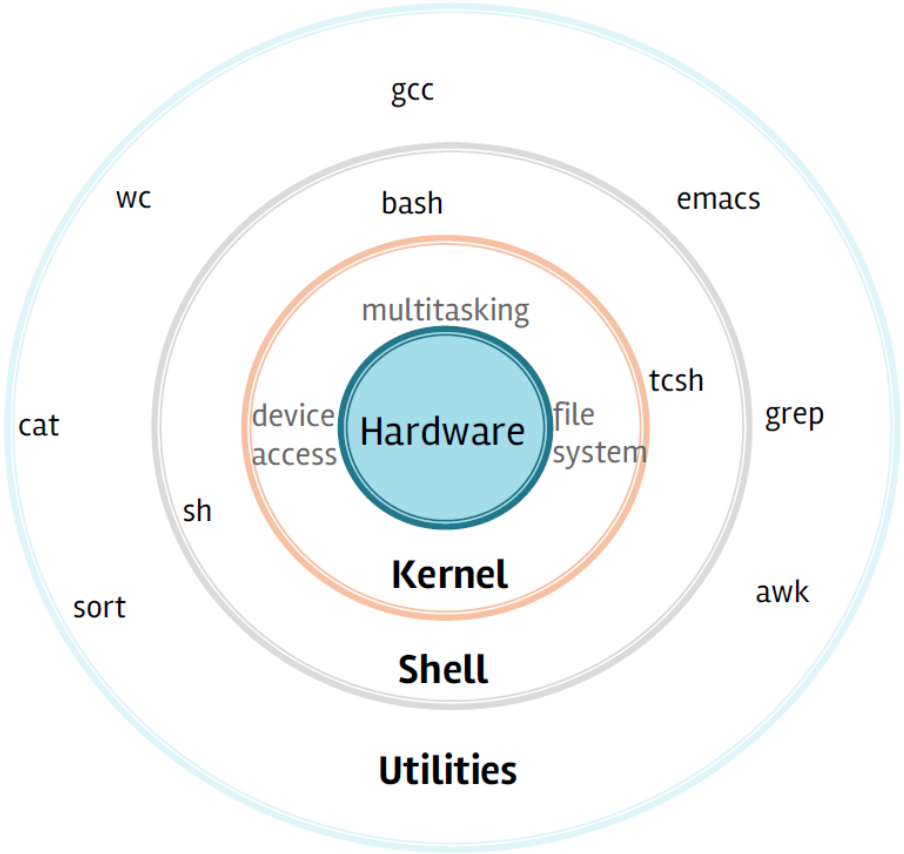
Base distributions

Derived distributions

CityU

What is Linux? (cont'd)

- Bird's eye view



Where is Linux

- World Wide Web
  - 67% of the world's web-servers run Linux (2016)
- Research/High-Performance Compute
  - Google, Amazon, NSA, 100% of TOP500 Super-computers [1]
- Modern Smartphones and devices
  - Android phones
  - Amazon Kindle
  - Smart TVs/Devices
- The most common OS used by CityU researchers when working on a server or computer cluster. 👍👍👍

1. Linux Runs on All of the Top 500 Supercomputers, Again! https://itsfoss.com/linux-runs-top-supercomputers/

Why Linux

- Free and open-source.
- Powerful for research datacenters
- Personal for desktops and phones
- Universal
- Community (and business) driven

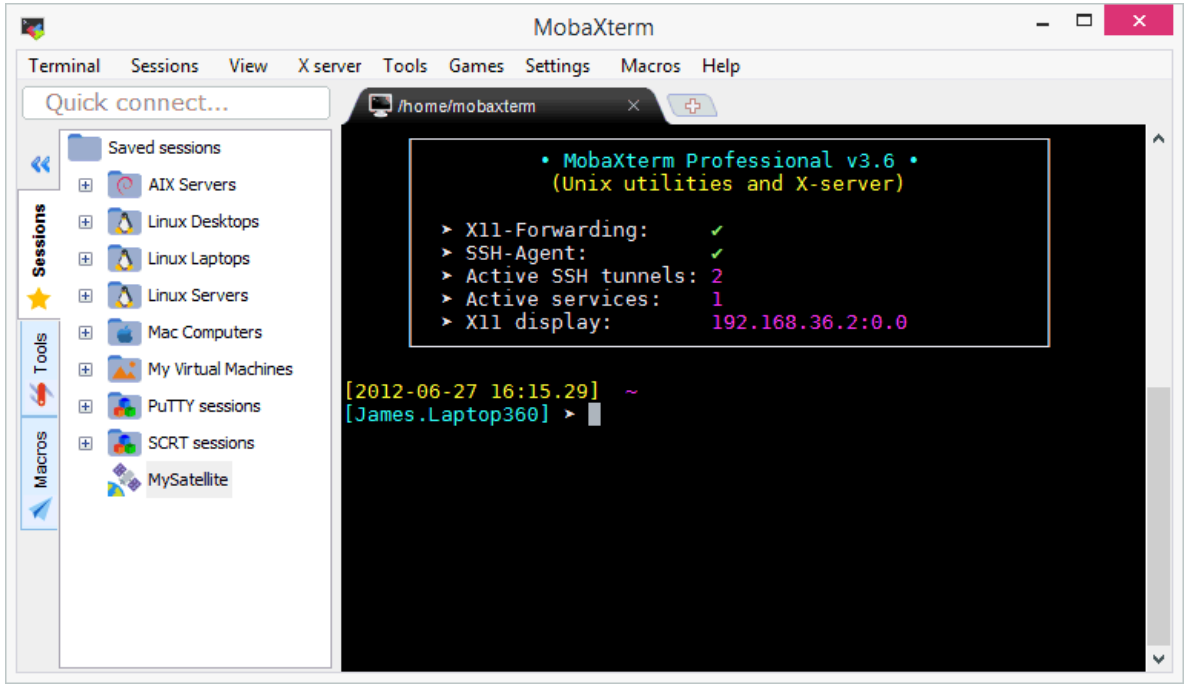
- We'll do labs and projects on Amazon Web Services (AWS) cloud, using Linux virtual machines.

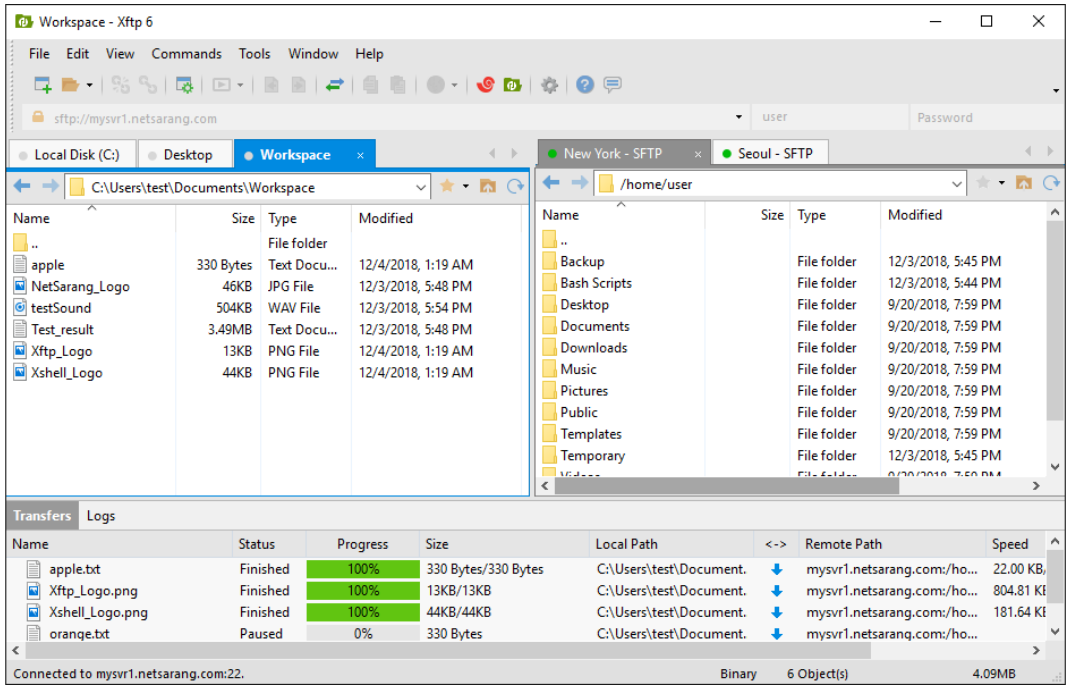
Connecting

Let’s use Linux

Connection Protocols and Software

- Remote Connections: Secure Shell (SSH)
- Data Transfer: Secure File Transfer Protocol (SFTP)





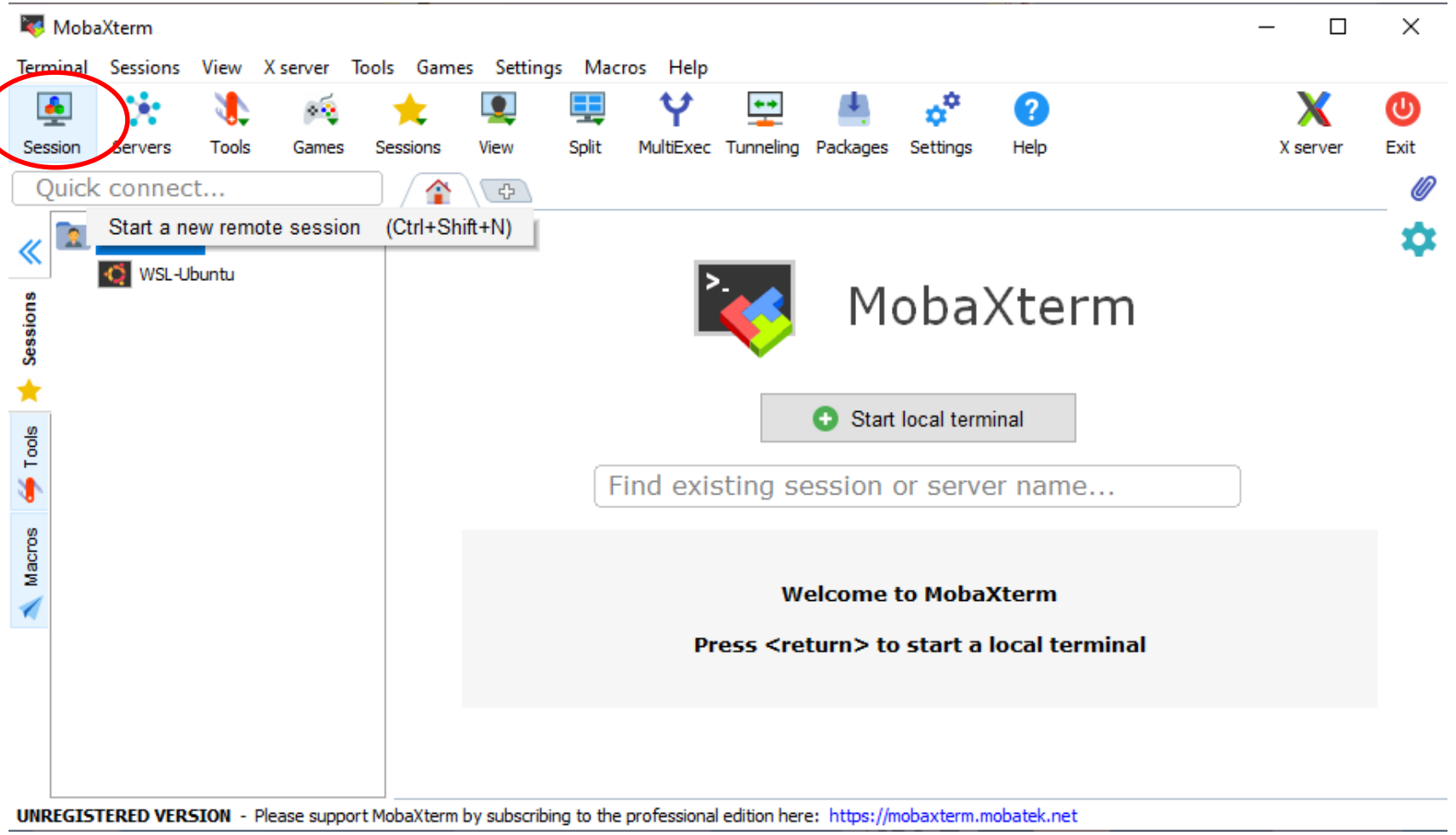
Connecting from Different Platforms

	SSH	SFTP
Microsoft Windows	MobaXterm, Xshell, PuTTY	MobaXterm, Xftp, FileZilla, Cyberduck
Apple macOS	Terminal (Built in), iTerm2	Cyberduck, ForkLift3, FileZilla
Linux	Terminal (Built in)	Various (Built in)

Note:
[1] For **CS4296** students: MobaXterm is available on all CSLab Windows machines. You can launch MobaXterm from *CSLab Menu*, which is on the Windows desktop.
[2] For **CS5296** students: PuTTY is available on the Work Desk Menu in the CSC Teaching Studios.

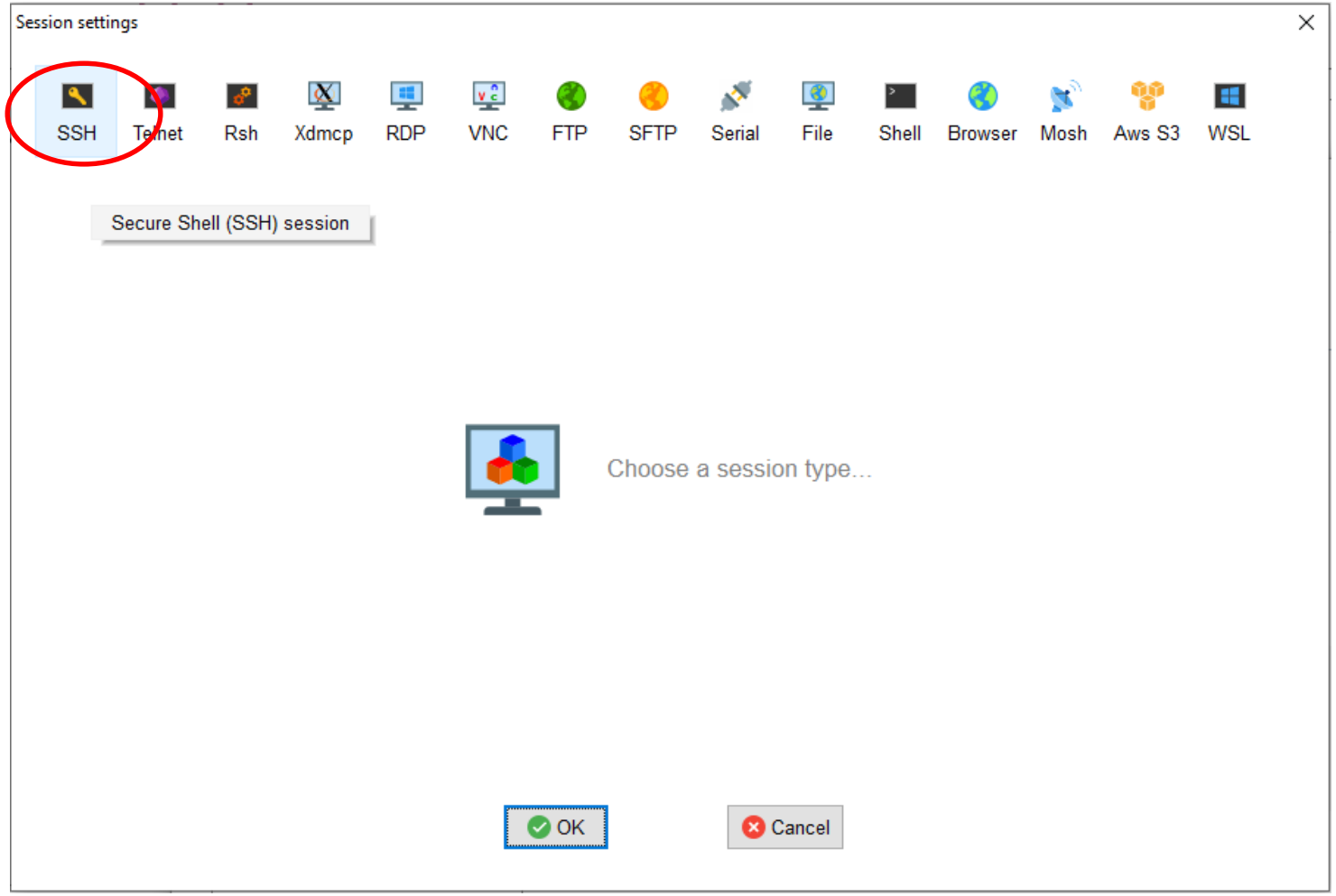
Microsoft Windows (MobaXterm)

- Launch MobaXterm from *CSLab Menu*, which is on the Windows desktop and start a “New Session”:



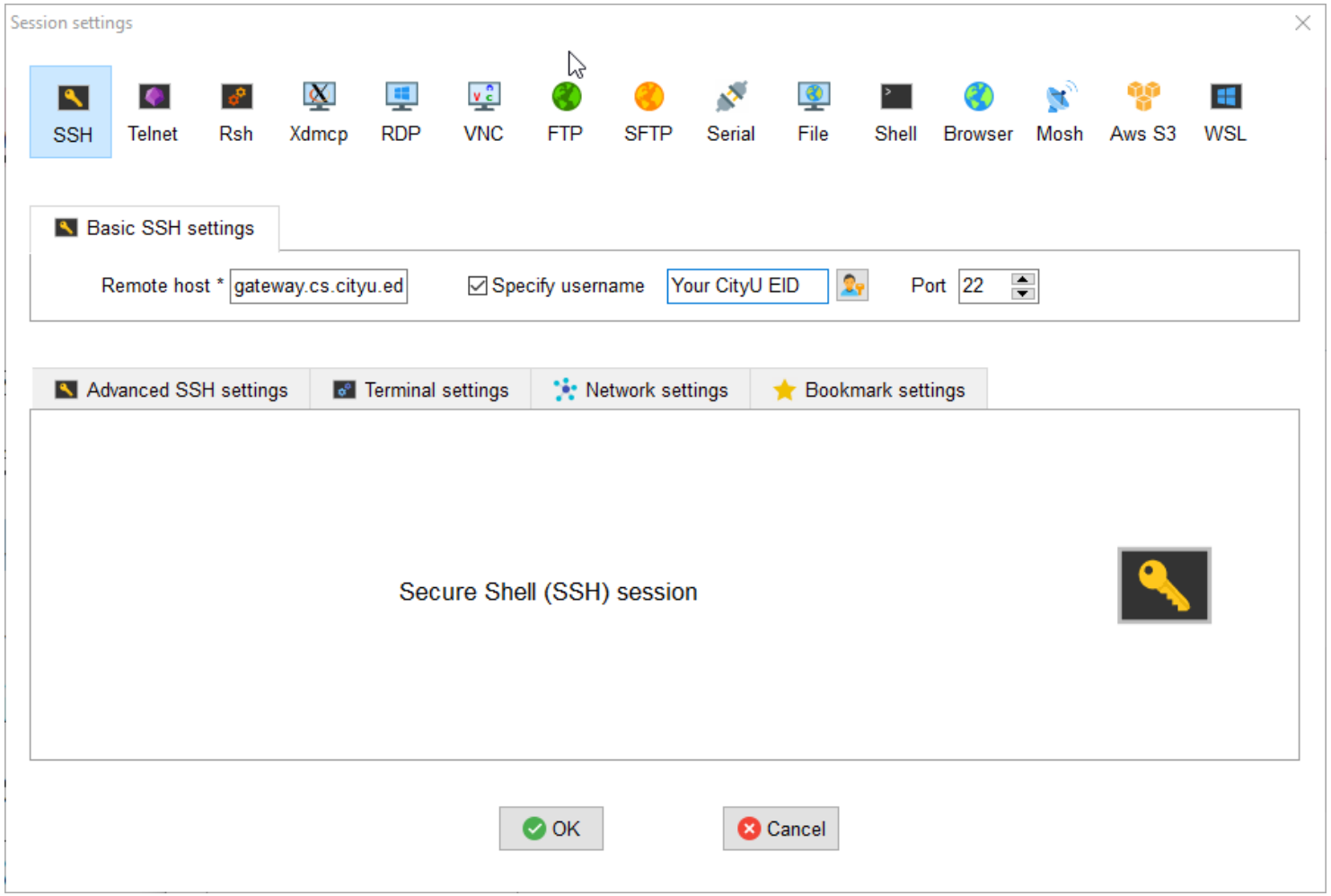
Microsoft Windows (MobaXterm)

- Select “SSH” as the session type:



Microsoft Windows (MobaXterm)

- Specify “gateway.cs.cityu.edu.hk” as the remote host, your EID (e.g., cctom2) as the username, and click “OK”

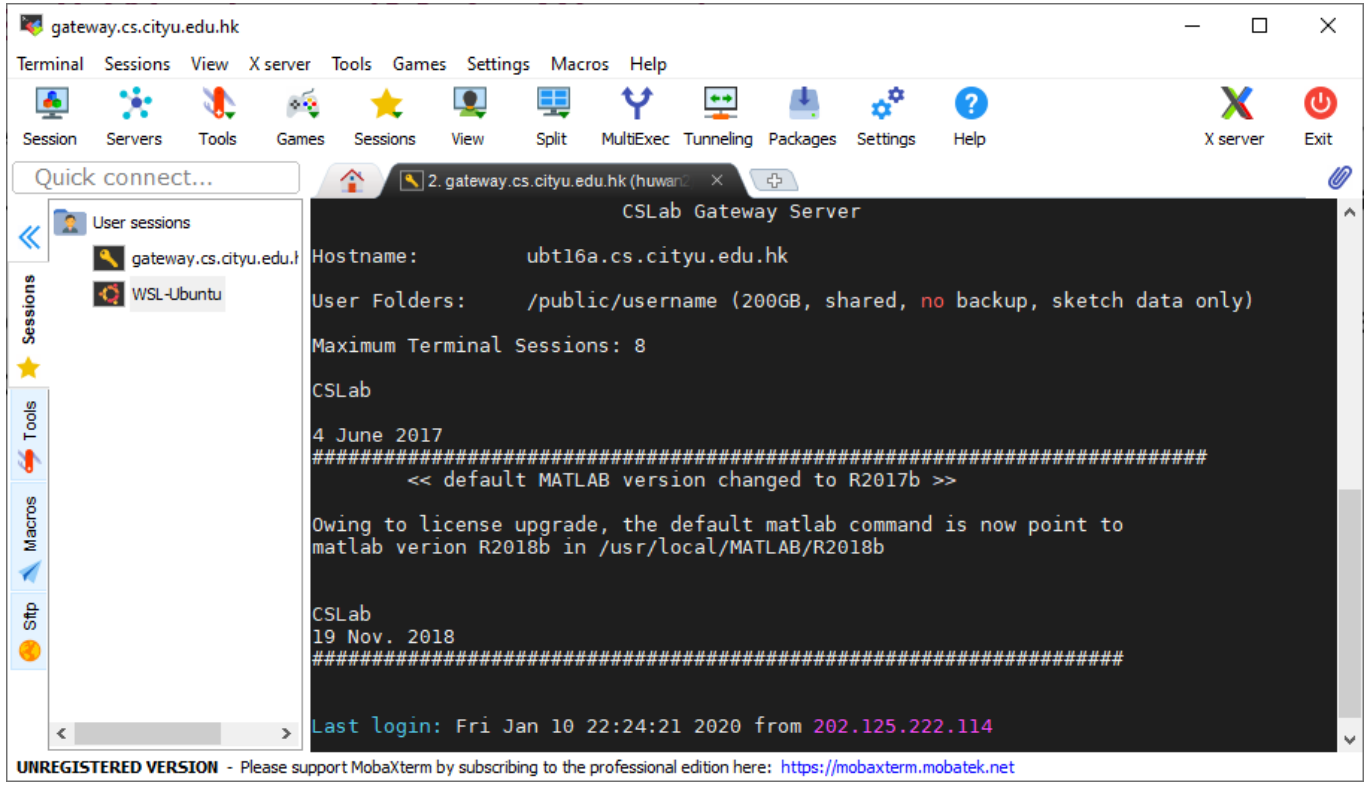


Microsoft Windows (MobaXterm)

- In the terminal window you will get a prompt to enter your password (Note that the characters in your password will not be displayed when you type them as a security precaution).

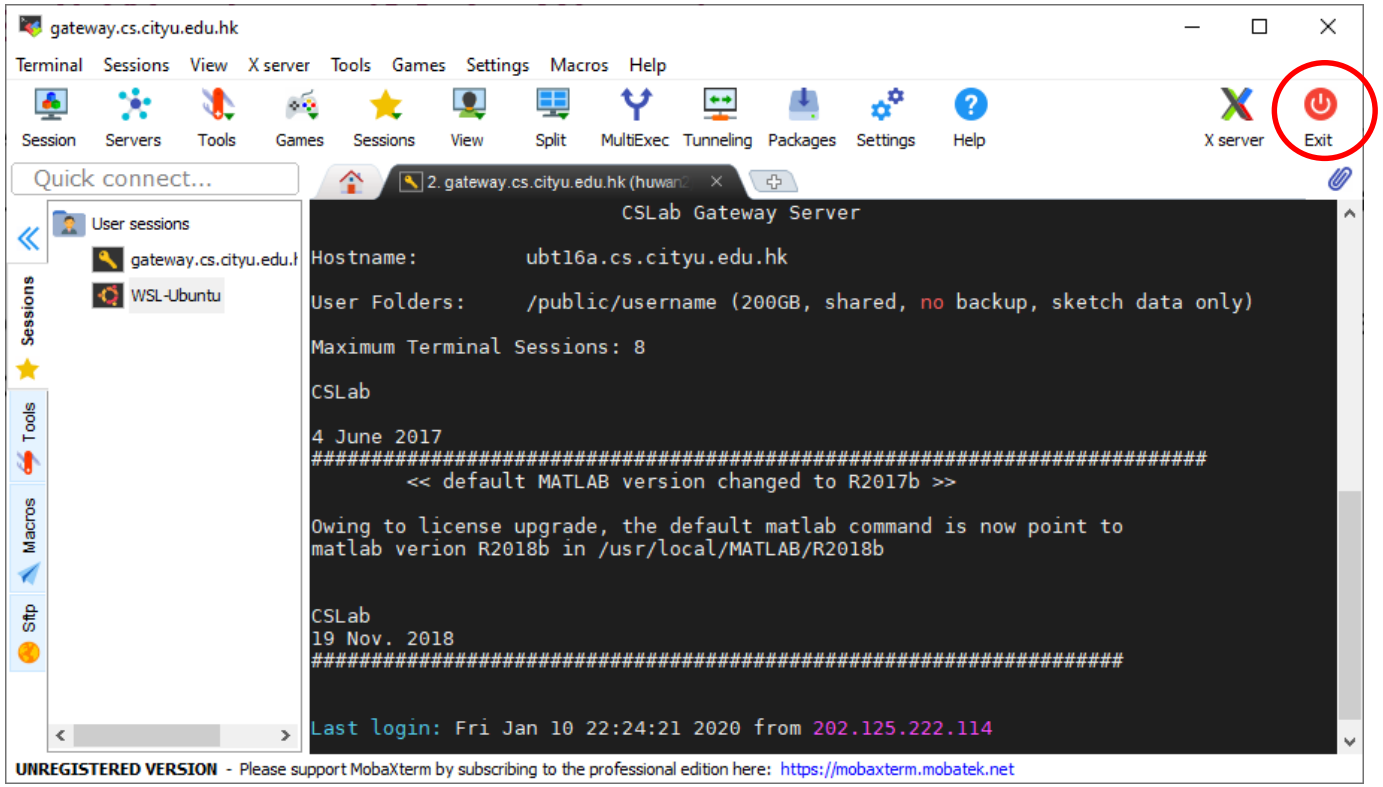
- Your connection will be saved on the left sidebar, so the next time you can start your session by clicking the “gateway.cs.cityu.edu.hk (yourEID)” link.

- You can edit, delete, and move sessions by right clicking on them in the left MobaXterm sidebar.



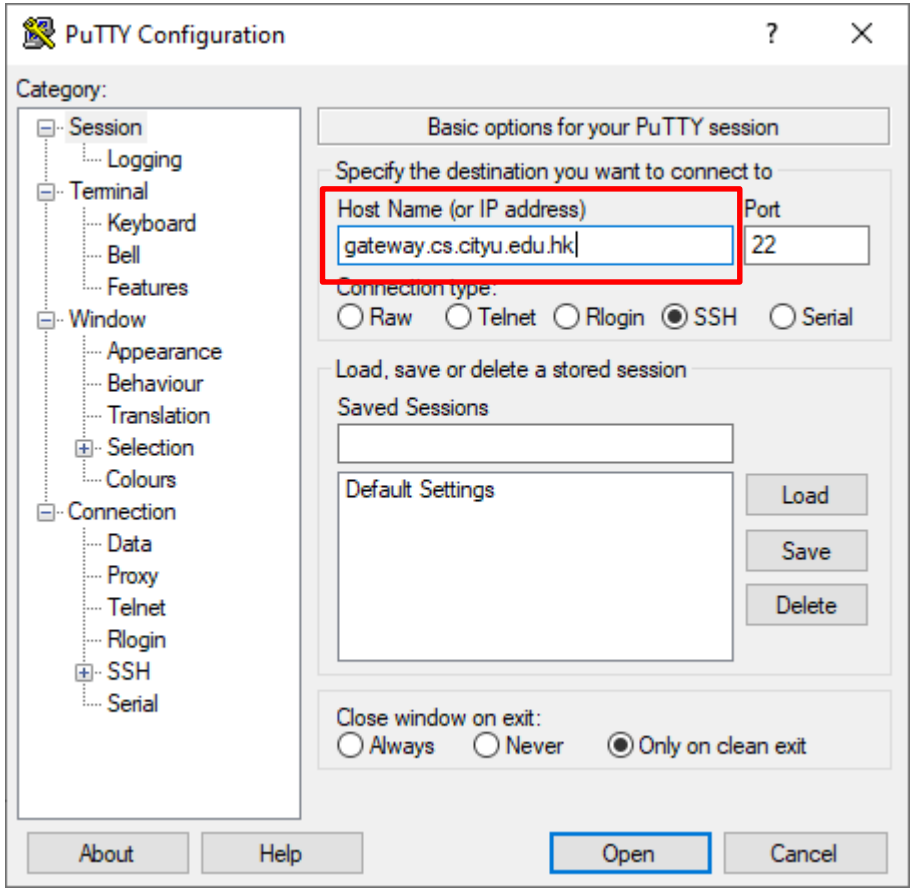
Microsoft Windows (MobaXterm)

- Exit an SSH session (logout)
  - The session terminates when you exit the command-line shell on the server (typically by typing `exit`) to the command line.
  - Alternatively, you can terminate the session by clicking Exit button or closing the terminal window.



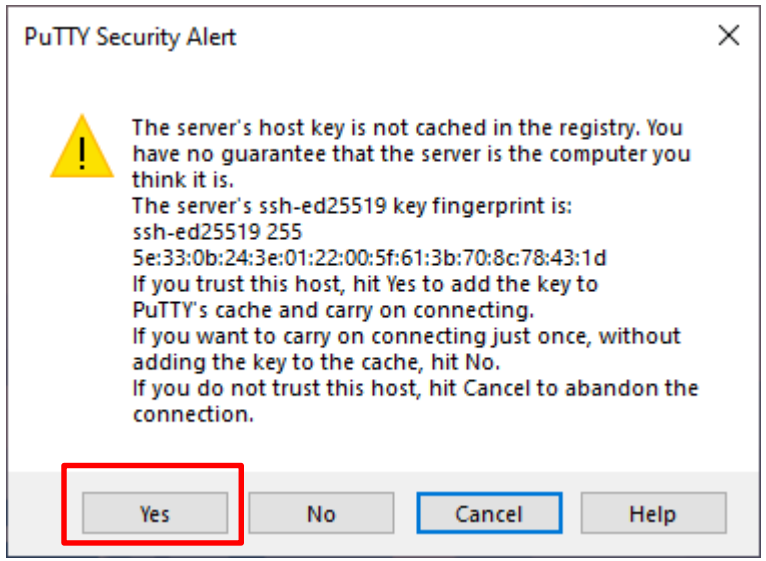
Microsoft Windows (PuTTY)

- Launch **PuTTY** from *Work Desk Menu*. When the software starts, a window titled **PuTTY Configuration** should open. Once you reach the PuTTY Configuration window, enter the host name (this is gateway.cs.cityu.edu.hk) and click Open.



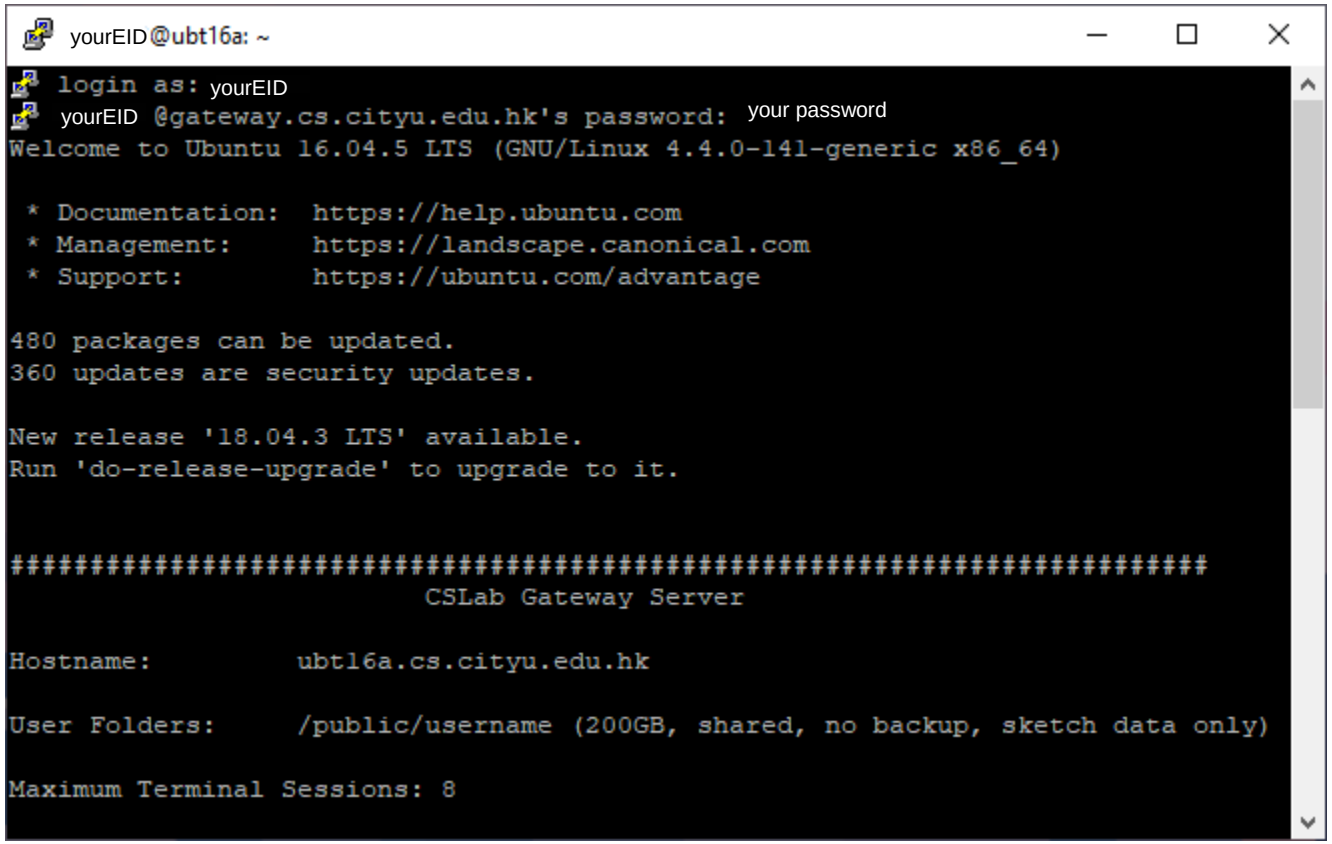
Microsoft Windows (PuTTY)

- When you connect to a server for the first time, you are likely to see a **PuTTY Security Alert** dialog about the server's host key not being cached in the registry. This is normal when you are connecting to a server for the first time, and you should just click **Yes**..



Microsoft Windows (PuTTY)

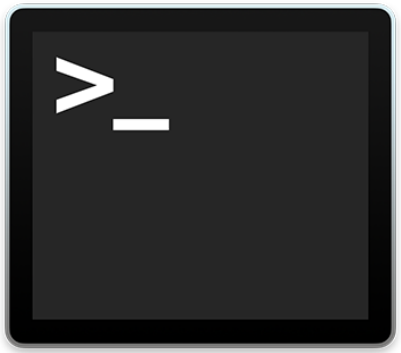
- After the security alert, you should get a terminal window. By default, this is a black, very bland window. It should first ask for your user name and then password (Note that the characters in your password will not be displayed when you type them as a security precaution). After these, you should get a command line on the server.



Microsoft Windows (PuTTY)

- Exit an SSH session (logout)
  - The session terminates when you exit the command-line shell on the server (typically by typing `exit`) to the command line or pressing Control-D.
  - Alternatively, you can forcibly terminate the session by closing the terminal window.

Other Platforms



- Apple macOS
  - Built in Terminal: Applications > Utilities > Terminal
- Linux
  - Built in Terminal: Applications > System > Terminal
- Connect:
  - The tool on Linux for connecting to a remote system using SSH is called, unsurprisingly, **ssh**.
  - The most basic form of the command is: ssh username@remote_host. Username is your EID, and the *remote_host* in this lab is gateway.cs.cityu.edu.hk.
  - Once you have connected to the server, you will probably be asked to verify your identity by providing a password.
  - When you first connect you will receive a notification text, and it is necessary to specify the option of yes to continue the connection.

Other Platforms (cont'd)

Enter your password. Note that your password will not be shown on the screen as you type it, not even as a row of stars (******).

```
$ ssh cctom2@gateway.cs.cityu.edu.hk
cctom2@gateway.cs.cityu.edu.hk's password:
The authenticity of host 'gateway.cs.cityu.edu.hk (144.214.37.68)' can't be established.
RSA key fingerprint is f3:cf:58:ae:71:0b:c8:04:6f:34:a3:b2:e4:1e:0c:8b.
Are you sure you want to continue connecting (yes/no)? yes
cctom2@ubt16a:~$
```

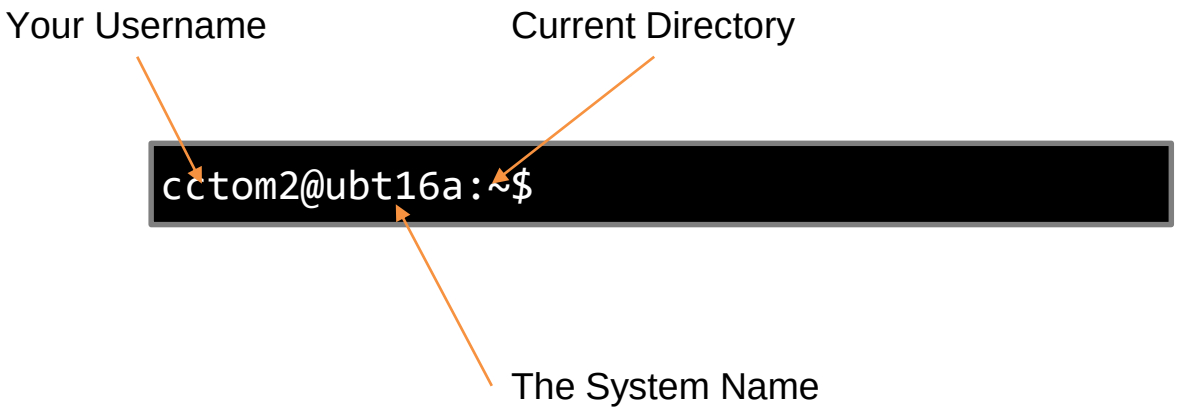
Type "yes".

Linux Interaction

- The Shell
  - Program that interprets commands and sends them to the OS. It provides built-in commands.
  - Linux supports multiple shells. The default on CSLab gateway server is Bash or csh.

"Bash" = "**B**ourne-**a**gain **Sh**ell"
(GNU version of ~1977 shell written by Stephen Bourne)

- The "prompt"
  - After successfully logging in, the shell will always give you a prompt if it is ready to accept commands. A shell prompt normally ends in a $ sign like this. Some shell prompts use % or > instead.



(In Linux " ~ " is a shorthand for your home directory.)

Linux: Command Basics

```
cctom2@ubt16a:~$ command --option argument
```

- Commands have three parts; command, options and arguments/parameters.
- Command: Command/program that does one thing
- Options: Change the way a command does that one thing
  - Short form: Single-dash and one letter e.g. **`ls -a`**
  - Long form: Double-dash and a word e.g. **`ls --all`**
- Argument: Provides the input/output that the command interacts with
- Example: `cal -j 1 2020`. "`cal`" is the command, "-j" is an option (or switch), "1" and "2020" are arguments/parameters.

```
cctom2@ubt16a:~$ cal -j 1 2020
       January 2020
 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  29  30  31
```

Try it out

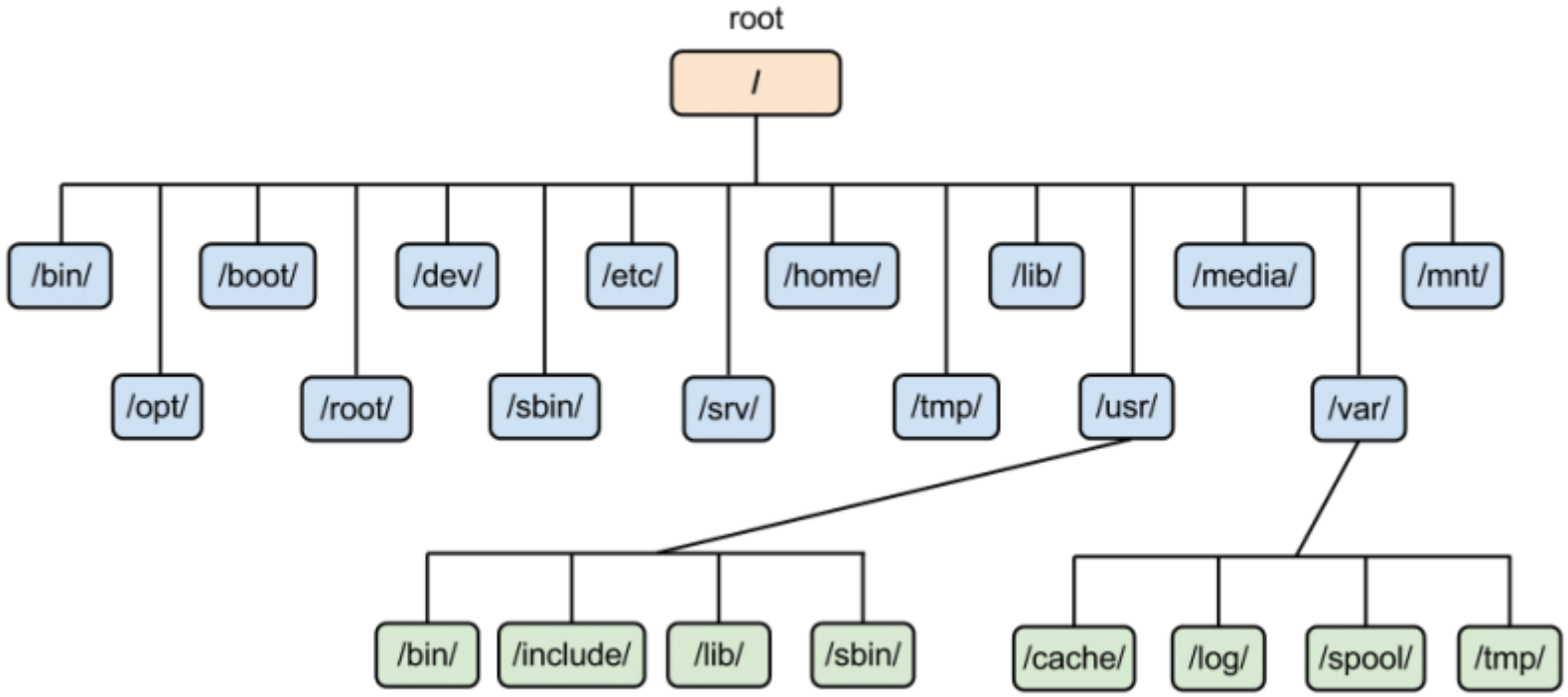
- After you connect, type the following commands:
  - `whoami` # my login
  - `hostname` # name of this computer
  - `echo “Hello, world”` # print characters to screen
  - `date` # print current time/date
  - `cal` # print this month’s calendar
- Help with Commands
  - For more information about any command, use `man` or `info`, `--help` option.
    - `date --help`
    - `man date`
    - `info date`

    Note: press `q` to quit when using `man` or `info`.

Yes, you can always Google it.

Linux The Filesystem

- The structure resembles an upside-down tree
- Directories (a.k.a. folders) are collections of files and other directories.
- Every directory has a parent except for the root directory.
- Many directories have subdirectories.



Source: https://www.rs-online.com/designspark/an-intro-to-linux-file-system-management

Navigating the File System

- Essential navigation commands:
  - `pwd` print current directory
  - `ls` list files
  - `cd` change directory

- We use pathnames to refer to files and directories in the Linux file system. There are two types of pathnames:
  - Absolute – The full path to a directory or file; begins with /
  - Relative – A partial path that is relative to the current working directory; does not begin with /

Navigating the File System (cont'd)

- Special characters interpreted by the shell for filename expansion:
  - `~` your home directory (e.g., /usr1/tutorial/tuta1)
  - `.` current directory
  - `..` parent directory
  - `*` wildcard matching any filename
  - `?` wildcard matching any character
  - `TAB` try to complete (partially typed) filename
- Examples:
  - `cd /usr/local` Change directory to /usr/local/lib
  - `cd ~` Change to home directory (could just type 'cd')
  - `pwd` Print working (current) directory
  - `cd ..` Change directory to the "parent" directory
  - `cd /` Change directory to the "root"
  - `ls -d pro*` Listing of only the directories starting with "pro"

The `ls` command

- Useful options for the "`ls`" command:
  - `ls -a` List all files, including hidden files beginning with a "."
  - `ls -ld *` List details about a directory and not its contents
  - `ls -F` Put an indicator character at the end of each name
  - `ls -l` Simple long listing
  - `ls -lR` Recursive long listing
  - `ls -lh` Give human readable file sizes
  - `ls -lS` Sort files by file size
  - `ls -lt` Sort files by modification time (very useful!)

Some Useful File Commands

`cp [file1] [file2]`	copy file
`mkdir [name]`	make directory
`rmdir [name]`	remove (empty) directory
`mv [file] [destination]`	move/rename file
`rm [file]`	remove (-r for recursive)
`file [file]`	identify file type
`less [file]`	page through file
`head -n N [file]`	display first N lines
`tail -n N [file]`	display last N lines
`ln –s [file] [new]`	create symbolic link
`cat [file] [file2…]`	display file(s)
`touch [file]`	update modification time
`od [file]`	display file contents, esp. binary

Manipulating files and directories

- Examples:
  - `cd` # The same as cd ~
  - `mkdir test`
  - `cd test`
  - `echo ‘Hello everyone’ > myfile.txt`
  - `echo ‘Goodbye all’ >> myfile.txt`
  - `less myfile.txt`
  - `mkdir subdir1/subdir2` # Fails. Why?
  - `mkdir -p subdir1/subdir2` # Succeeds
  - `mv myfile.txt subdir1/subdir2`
  - `cd ..`
  - `rmdir test` # Fails. Why?
  - `rm –rf test` # Succeeds

File Editors

- `nano`
  - Lightweight editor.
- `vim (recommended)`
  - A better version of ‘vi’ (an early full-screen editor). Very fast, efficient.
  - Steep learning curve.
  - Popular among systems programmers.
- `emacs`
  - Swiss-army knife, has modes for all major languages, and can be customized.
  - Formerly steep learning curve has been reduced with introduction of menu and tool bars.

Resources:
[1] Interactive Vim tutorial https://www.openvim.com
[2] Vim Cheat Sheet https://vim.rtorr.com/
[3] How To Learn Vim: A Four Week Plan, https://medium.com/actualize-network/how-to-learn-vim-a-four-week-plan-cd8b376a9b85

CityU

More topics

- I/O Redirection
  - stdin (0) / stdout (1) / stderr (2)
  - I/O redirection with pipes (|)
- Processes & Job Control
  - Use the “`ps`” command to see a listing of processes
  - Use “`top`” command to see active processes
  - Use the “`kill`” command to terminate a job
  - Foreground/background processes

CityU