

# Computer Science 221a/b Software Tools and Systems Programming

## Lab 0 – Introduction to **UNIX** on the **GAUL** sub**Network** of **Computer Science Network**

# Objectives

- **The very basics**
  - Logging in to and out of the GAUL network from your home computer
  - Changing your password???
  - Understanding the UNIX file system
- **Working with directories**
  - Listing the contents of a directory
  - Displaying the current directory
  - Changing directories
  - Creating / deleting directories
- **Working with files**
  - Displaying the contents of files
  - Deleting files
  - Finding files
  - Finding content within files
- **Moving things around**
  - Moving, renaming, and copying files and directories
  - Restoring deleted files from backup
- **More, if there's time...**

# Why UNIX / Linux?

- UNIX and Linux are used everywhere!
  - 90% of the top 1 million Web servers ran Linux in 2018
  - 70% of the top 10 million Web servers ran Linux in 2018
- Good skill to have when entering the workforce
  - Not everyone runs Windows
  - May need it at some point in your career
- 2 for 1 deal
  - Linux was originally designed as a free clone of UNIX
  - Become proficient with one, you essentially know the other

# Part 1 – The Very Basics

## Part 1 – The Very Basics

# Your GAUL Account

- You should use your username / password of your uwo email account
  - No notification
  - Just try to login
- If you can not login within 3 days of enrolment
  - Open a ticket at <http://helpdesk.sci.uwo.ca>

# Logging in to GAUL from Home

- Connect to GAUL using the SSH protocol
  - SSH creates an encrypted connection; stands for *Secure Shell*
- Should know how to SSH from Windows, Mac, and Linux
  - We have all of these systems in our computer labs
- **`compute.gaul.csd.uwo.ca`** is, in general, the only undergraduate server accessible from home.
  - Hence, you should SSH “into this server”
  - That is, you will connect to this server from home

# Connecting from Mac OS

- You already have an SSH client
  - No need to download a program
- Add the **Terminal** to your dock
  - Open **Macintosh HD > Applications > Utilities**
  - Drag **Terminal** to an empty space in your dock
- Click the **Terminal** icon in your dock
- Type: `ssh your_username@compute.gaul.csd.uwo.ca`
  - Replace *your\_username* with your uwo email username

# Connecting from Linux

- You too already have an SSH client
  - No need to download a program
- Run the terminal program from your application menu
  - In **GNOME**, this is usually called **Terminal**
  - In **KDE**, this is usually called **Konsole**
- Type: `ssh your_username@compute.gaul.csd.uwo.ca`
  - Replace *your\_username* with your uwo email username



# Connecting from Windows 10

- You too already have an SSH client
  - No need to download a program
- Run the PowerShell program from your start menu
  - PowerShell opens a **windows PowerShell**
  - One can also use **command prompt window**
- Type: `ssh your_username@compute.gaul.csd.uwo.ca`
  - Replace *your\_username* with your uwo email username

# Connecting from Other Windows

- Use PuTTY
  - Free SSH *client* for Windows
  - Available at <http://bit.ly/3ytNG>
  - Download **putty.exe** under **For Windows on Intel x86**
    - Doesn't need to be installed
    - Just download and run!
    - Perfect for use in the computer labs, where you can't install programs
  - There is also a version available which comes with an installer
    - Currently, **putty-0.74-installer.msi**, **putty-64bit-0.74-installer.msi**
    - This will add shortcuts to your desktop and Start menu

# Logging Out

- Use the **exit** or **logout** command
  - You should log out when you are finished on GAUL to avoid unnecessarily tying up server resources
  - `compute.gaul.csd.uwo.ca` automatically logs you out if you are idle for too long
  - `compute.gaul.csd.uwo.ca` sometimes reboots early in the morning, so you may find yourself automatically logged out when that happens

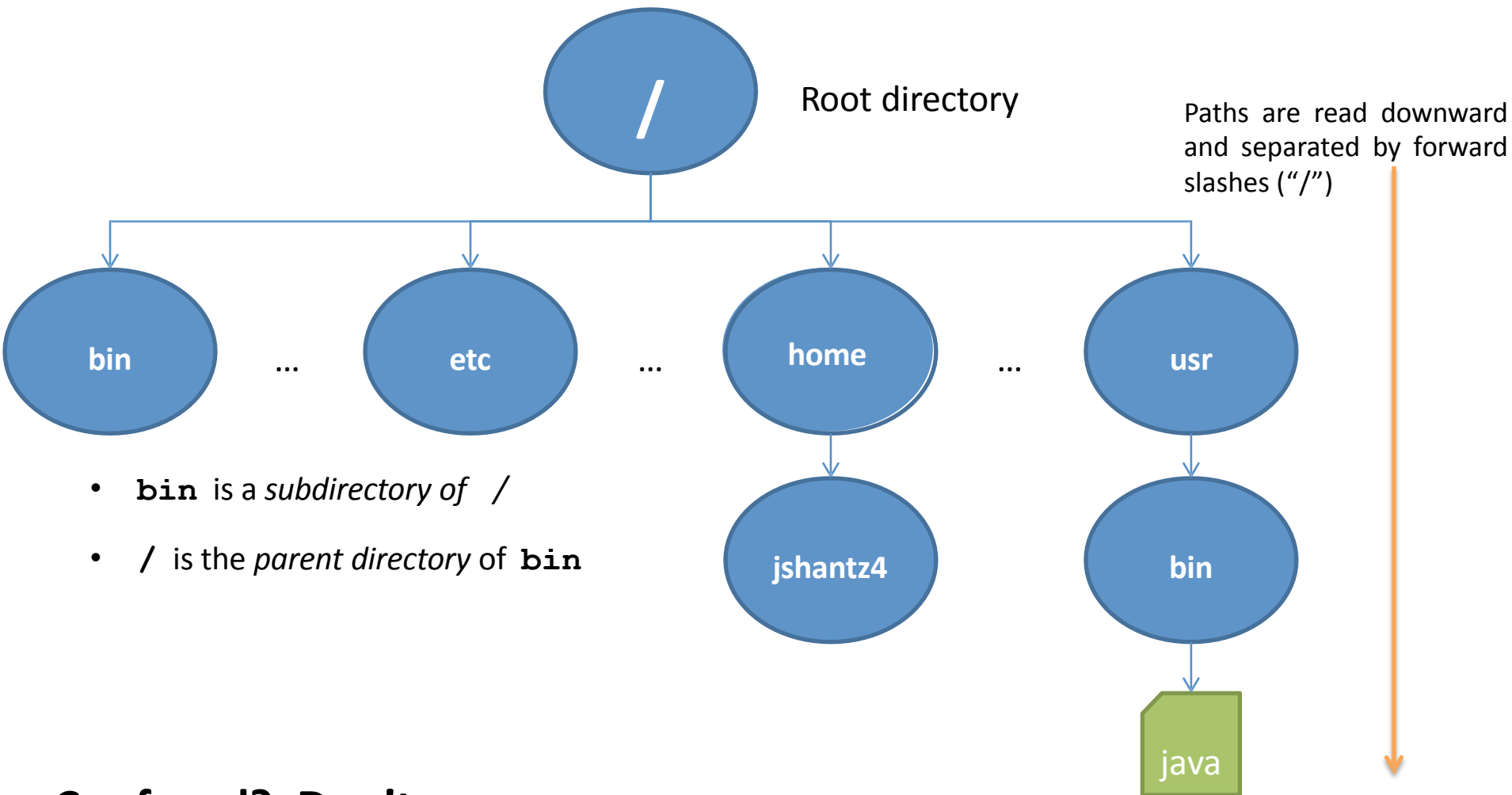
# Changing Your Password

- Can not use the **passwd** command to change. Use uwo identity manager.
- Passwords are case sensitive
  - "A" is not the same as "a"
- Choose a strong password
  - At least 8 characters; preferably more
  - Include a mix of uppercase and lowercase letters
  - Include numbers and symbols (e.g. ! @ # \$ %)
  - Don't use dictionary words – easily cracked
  - Choose a phrase easily remembered, then convert it to *leet-speak*
    - "I hate carrots a lot" ➡ **ih8^s4L0t!**
  - Don't write it down

# UNIX File System

- Files
  - Contain our “data”
  - e.g. MP3, PDF, text, etc.
- Directories
  - Contain files and other directories
  - Analogous to *folders* on Windows and Mac OS
- File system
  - Collection of files and directories
  - Organized into a tree structure

# UNIX File System (cont'd...)



## Confused? Don't worry.

This will become second nature to you as you gain experience with UNIX.

The file **java** is located in **/usr/bin**, so its *path* is **/usr/bin/java**

# UNIX File System (cont'd...)

- **Current working directory**
  - The directory in which you're currently located
- **Parent directory**
  - The directory that contains the directory in question
  - i.e. The directory one level up in the file system tree
- **Special References**
  - Each directory **d** contains the following references:

Reference	Meaning
.	Refers to directory <b>d</b>
..	Refers to the parent of <b>d</b>

**We'll see why these references are needed later**

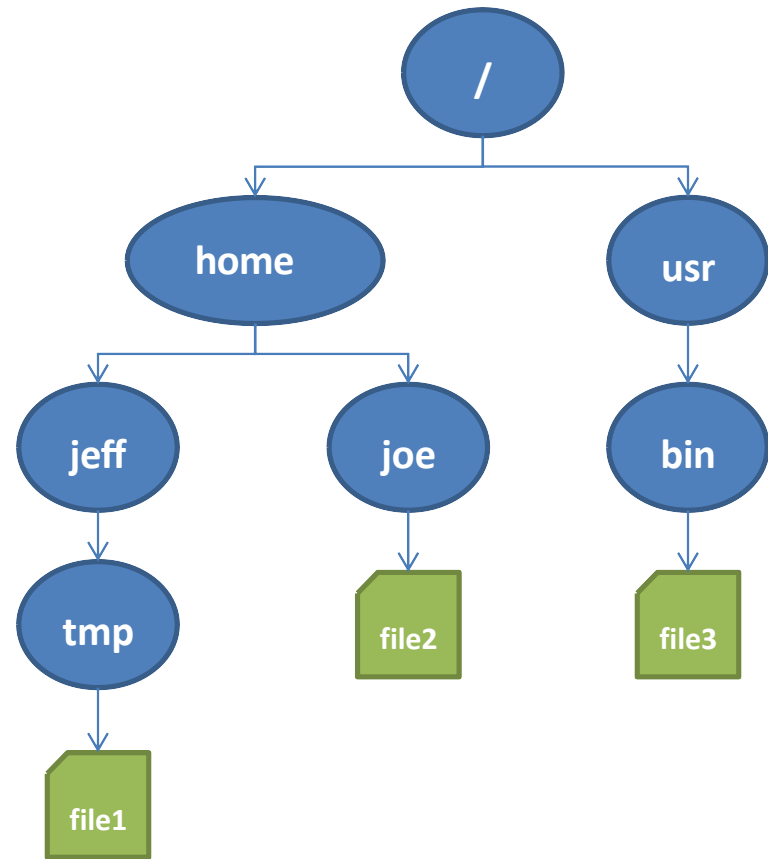
# UNIX File System (cont'd...)

## ■ Home directory

- Special directory that "belongs" to you
- Only you can create files and directories in this directory
- When you login, this is where you are initially placed
- Your home directory is a subdirectory of **/home**
  - e.g. The user **jshantz4** has his home directory in **/home/jshantz4**
- Can be referenced by the tilde character (~)
  - ~ references your home directory
  - **~user** references the home directory of **user**



# Absolute vs. Relative Paths



Suppose we want to access **file3**

- If we're in **/home/jeff**, we could reference **file3** by typing:

**/usr/bin/file3**

- If we're in **/home**, we could again reference **file3** by typing:

**/usr/bin/file3**

- Thus, no matter where we are in the file system, we can access **file3** with the path:

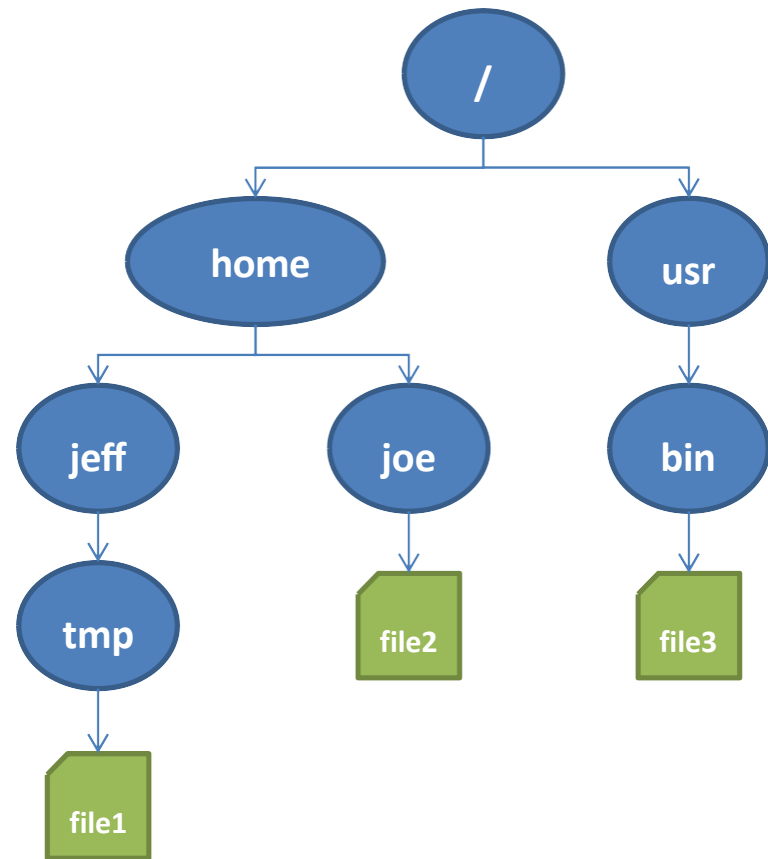
**/usr/bin/file3**

# Absolute vs. Relative Paths (cont'd...)

- **Absolute path**

- The path references the same location, no matter where we currently are in the file system
- Can think of an absolute path as being relative to the root directory

# Absolute vs. Relative Paths (cont'd...)



Suppose again we want to access **file3**

- If we're in **/home/jeff**, we could reference **file3** by typing:

`../../usr/bin/file3`

- If we're in **/home**, we could again reference **file3** by typing:

`../usr/bin/file3`

- This time, we are expressing the path *relative to* the directory we're currently in

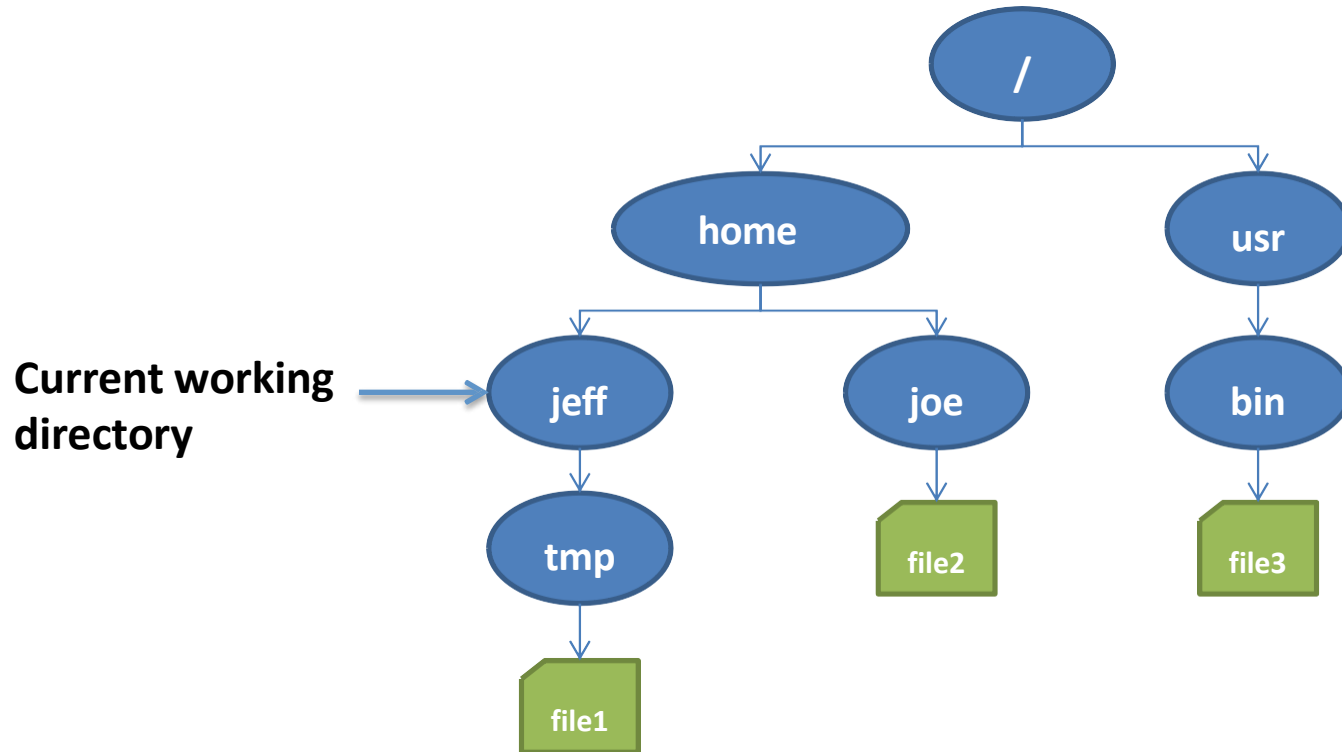
# Absolute vs. Relative Paths (cont'd...)

- **Relative path**

- The path is relative to the current working directory
- Recall the special references that exist in each directory **d**:

Reference	Meaning
.	Refers to directory <b>d</b>
..	Refers to the parent of <b>d</b>

# Absolute vs. Relative Paths (cont'd...)



Want to access	Relative path(s)	Absolute path
file1	<code>tmp/file1</code> <code>./tmp/file1</code>	<code>/home/jeff/tmp/file1</code>
file2	<code>../joe/file2</code>	<code>/home/joe/file2</code>
file3	<code>../../usr/bin/file3</code>	<code>/usr/bin/file3</code>